Subfield-based Parallel Kernel-thinning Algorithms on the BCC Grid

Gábor Karai[a], Péter Kardos[b] and Kálmán Palágyi[c]
Department of Image Processing and Computer Graphics, University of Szeged, Szeged, Hungary

Keywords: Shape Representation, Kernel-thinning, BCC Grid, Topology-preservation.

Abstract: Kernel-thinning is a widely used technique for extracting the topological kernel from a digital object (i.e., producing a minimal structure that is topologically equivalent to the original elongated object). In this paper, two subfield-based parallel kernel-thinning algorithms acting on the non-standard body centered cubic (BCC) grid are presented. Our algorithms combine a sufficient condition for topology preservation with two types of partitionings of the BCC grid, thus both algorithms are topology-preserving. According to our best knowledge, the reported algorithms are the very first parallel thinning algorithms on the BCC grid.

1 INTRODUCTION

A *binary digital picture* (*picture* in short) on a discrete space is composed of *black* or *white* points (Kong and Rosenfeld, 1989). *Skeletonization* provides frequently applied region-based shape descriptors which represent the general shape of binary objects that are present in images (Saha et al., 2016; Saha et al., 2017; Siddiqi and Pizer, 2008). In 3D, there are three types of skeleton-like features: the *centerline*, the *medial surface*, and the *topological kernel*. The centerline is a line-like 1D representation of objects that captures the part-whole structure of the object to be described (Cornea et al., 2007; Sobiecki et al., 2014). In many applications (Saha et al., 2016), it is a concise representation of tubular and tree-like 3D objects. The medial surface provides an approximation to the continuous 3D skeleton, since it can contain 2D surface patches. A topological kernel of a 3D object is a minimal set of points that is topologically equivalent (Kong and Rosenfeld, 1989) to the original object. It is fairly useful in representing or checking the topological structure of the object to be processed. Note that a topological kernel of a 3D object is an isolated point if and only if it does not contain any holes (that donuts have) nor cavities (i.e., bubbles). Otherwise, topological kernels are formed by 1-point thick closed curves and/or 1-point thick closed surfaces.

Several approaches have been proposed for producing skeleton-like features from (segmented) binary objects. Some authors presented comprehensive and concise surveys (Saha et al., 2016; Saha et al., 2017; Siddiqi and Pizer, 2008; Sobiecki et al., 2014; Tagliasacchi et al., 2016). *Thinning* is one of the major skeletonization techniques (Hall, 1996). It is an iterative object-reduction process for producing skeleton-like features in a topology-preserving way: the outmost layer of an object is deleted, and the entire process is repeated until stability is reached. The topology-oriented thinning pays less attention to the metric properties of the object to be represented, since the invariance under arbitrary rotation angles or scaling factors is not fulfilled. In spite of these drawbacks, our attention has been focused on thinning, since thinning is the fastest skeletonization method, it can be implemented easily, it can produce all types of skeleton-like features, the topology-preservation can be guaranteed, and thinning provides practically exquisite descriptors for a number of applications. There are three kinds of 3D thinning algorithms according to the three types of skeleton-like shape features: *curve-thinning* algorithms are used to extract centerlines, *surface-thinning* algorithms produce medial surfaces, while *kernel-thinning* or *reductive shrinking* ones are capable of extracting topological kernels (Hall et al., 1996).

A *parallel reduction* transforms a picture only by changing some set of black points to white ones simultaneously, which is referred to as *deletion* (Hall, 1996). Parallel thinning algorithms are composed of parallel reductions, and they fall into three major categories: fully parallel, subiteration-based (or directional), and subfield-based (Hall, 1996). *Fully par-*

[a] https://orcid.org/0000-0001-9609-8628
[b] https://orcid.org/0000-0001-8857-4102
[c] https://orcid.org/0000-0002-3274-7315

Karai, G., Kardos, P. and Palágyi, K.
Subfield-based Parallel Kernel-thinning Algorithms on the BCC Grid.
DOI: 10.5220/0010831000003122
In *Proceedings of the 11th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2022)*, pages 288-295
ISBN: 978-989-758-549-4; ISSN: 2184-4313

allel algorithms apply the same parallel reduction in each thinning phase; in *subiteration-based* algorithms a cycle of a small set of parallel reductions are assigned to the selected kinds of deletion directions, and only border points of a certain kind can be deleted at a subiteration; *subfield-based* algorithms partition the given digital space into $k \geq 2$ subsets which are alternatively activated, and only some points in the active subfield can be deleted simultaneously. Similarly to the directional approach, an iteration step of a k-subfield algorithm is composed of k subcycles (i.e., parallel reductions).

Existing 3D thinning algorithms act on the conventional 3D cubic grid, in which each point is associated with an element of $\mathbb{Z}^3$ (i.e., a point in the 3D Euclidean space with integer coordinates), and its voxel-representation contains unit cubes. According to our best knowledge, no one proposed kernel-thinning on the body-centered cubic (BCC) grid. The voxel-representations associated with this non-standard grid contain truncated octahedra. The importance of the BCC grid shows an upward tendency due to its advantages of geometric and topologic properties (Čomić and Nagy, 2016; Csébfalvi, 2013; Matej and Lewitt, 1995; Strand, 2004; Strand and Nagy, 2008; Theussl et al., 2001).

In this paper, the very first topology-preserving parallel kernel-thinning algorithms acting on the BCC grid are presented. Both of the proposed algorithms fall into the category of subfield-based.

The rest of this paper is organized as follows: Section 2 gives an outline of the key concepts of digital topology and the relevant results are described. Then in Section 3, two novel kernel-thinning algorithms are proposed. In Section 4 results on some test images produced by our algorithms are given. Finally, we round off this paper with some concluding remarks.

2 BASIC NOTIONS AND RESULTS

Next, we define the key concepts of digital topology as reviewed in (Kong and Rosenfeld, 1989).

A $(14,14)$ *picture* on the BCC grid is a quadruple $(\mathbb{B},14,14,B)$, where an element of $\mathbb{B}$ is assigned to each point; $B \subseteq \mathbb{B}$ denotes the set of *black point*s; each point in $\mathbb{B} \setminus B$ is said to be a *white point*; The same adjacency relation called 14-neighborhood is assigned to the sets of black and white points. Let $N_{14}(p)$ denote the set of points that are 14-adjacent to p, see Fig. 1.

Since the studied adjacency relation is symmetric, its reflexive-transitive closure forms an equivalence relation, and the generated equivalence classes of a

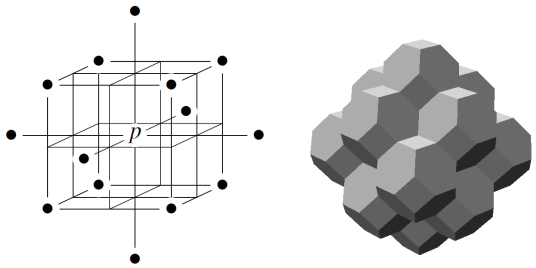


Figure 1: The studied adjacency relation on $\mathbb{B}$ (left). The 14 points marked '•' form the set $N_{14}(p)$. (Note that unmarked elements in $\mathbb{Z}^3$ are not points in $\mathbb{B}$.) The voxel-representation of $N_{14}(p)$, where each voxel is a truncated octahedron (right).

set of points are called *component*s. A *black component* or an *object* is a 14-component of B, while a *white component* is a 14-component of $\mathbb{B} \setminus B$.

A point $p \in B$ is an *interior point* for B, if all points being 14-adjacent to p are in B (i.e., $N_{14}(p) \subset B$), p is called a *border point* if it is not an interior point, and p is said to be an *isolated point* if it forms a singleton object (i.e., $N_{14}(p) \cap B = \emptyset$).

Thinning algorithms, composed of reductions, are required to preserve topology (Kong, 1995). A reduction in 2D does not preserve topology if any object in the input picture is split (into several objects) or is completely deleted, any white component in the input picture is merged with another white component, or a white component is created where there was none in the input picture. There is an additional concept called *hole* in 3D pictures. Holes (which donuts have) are formed from white points, but they are not white components (Kong and Rosenfeld, 1989). Topology preservation in 3D implies that eliminating or creating any hole is not allowed.

A black point is said to be *simple* if its deletion is a topology-preserving reduction. Now we will make use of the following characterization of simple points:

Theorem 1. (Strand and Brunner, 2006) *A point $p \in B$ in picture $(\mathbb{B},14,14,B)$ is* simple *if and only if the following conditions hold:*

1. *$N_{14}(p) \cap B$ contains exactly one component.*
2. *$N_{14}(p) \setminus B$ contains exactly one component.*

It is an easy consequence of Theorem 1 that only non-isolated border points may be simple, and the simpleness is a local property (i.e., it can be decided by examining the points that are 14-adjacent to the given black point). Figure 2 gives four illustrative examples of simple and non-simple points.

Parallel reductions delete a set of points and not just a single black point. Thus we need to consider what is meant by topology preservation when a num-

ber of points are deleted at a time. One of the authors established the following sufficient conditions for topology-preserving reductions:

Theorem 2. (Kardos, 2021) *A reduction is topology-preserving in picture* $(\mathbb{B}, 14, 14, B)$ *if the following conditions hold:*

1. *Only simple points are deleted.*
2. *If two* 14*-adjacent points* p *and* q *are deleted,* q *is simple in picture* $(\mathbb{B}, 14, 14, B \setminus \{p\})$.
3. *If three mutually* 14*-adjacent points are deleted, they can be arranged in a sequence* $\langle p, q, r \rangle$ *such that* p *is simple in picture* $(\mathbb{B}, 14, 14, B)$*,* q *is simple in picture* $(\mathbb{B}, 14, 14, B \setminus \{p\})$*, and* r *is simple in picture* $(\mathbb{B}, 14, 14, B \setminus \{p, q\})$.
4. *No object formed by four mutually* 14*-adjacent black points is deleted completely.*

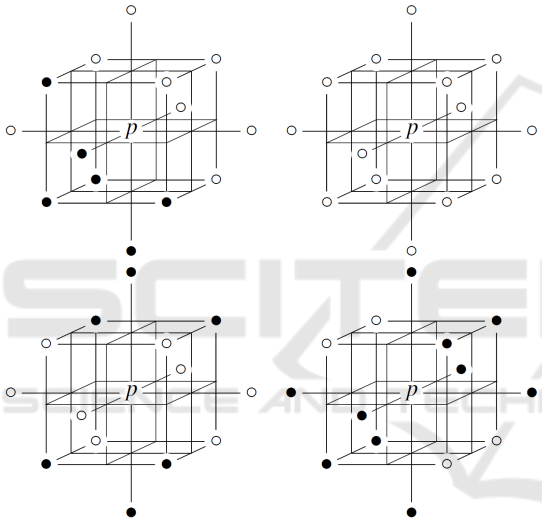


Figure 2: Examples of simple and non-simple points. The positions marked '•' and '○' represent black and white points, respectively. Black point p is simple only in the top left configuration. In the top right example, p is an isolated black point, thus Condition 2 of Theorem 1 is not hold. In the bottom left configuration, we can find two components in $N_{14}(p) \cap B$, hence Condition 1 of Theorem 1 is violated. The bottom right configuration depicts a case where there exist two 14-components in $N_{14}(p) \setminus B$, thus Condition 2 of Theorem 1 does not hold.

In Section 3, two subfield-based parallel kernel-thinning algorithms are reported. We propose the partitions of $\mathbb{B}$ into four and eight subfields as shown in Fig. 3 and Fig. 4, respectively.

Let us state now an important property of these partitionings:

Proposition 1. *If* $p \in S_k(i)$ $(k = 4, 8;\ i = 0, 1, \ldots, k-1)$ *and* $q \in N_{14}(p)$, $q \notin S_k(i)$.

It is obvious by careful examination of Fig. 3 and Fig. 4.

By Proposition 1, Theorem 2 can by simplified for reductions of subfield-based thinning algorithms:

Theorem 3. *A reduction that deletes a subset of* $S_k(i) \cap B$ $(k = 4, 8;\ i = 0, 1, \ldots, k-1)$ *from picture* $(\mathbb{B}, 14, 14, B)$ *is topology-preserving if only simple points are deleted.*

Proof. It is obvious that Condition 1 of Theorem 2 is satisfied.

With the help of Proposition 1, it can be readily seen that there is no mutually 14-adjacent pair (triplet and quadruple) of points in the same subfield according to the proposed partitionings. Thus the last three conditions of Theorem 2 are irrelevant here. □

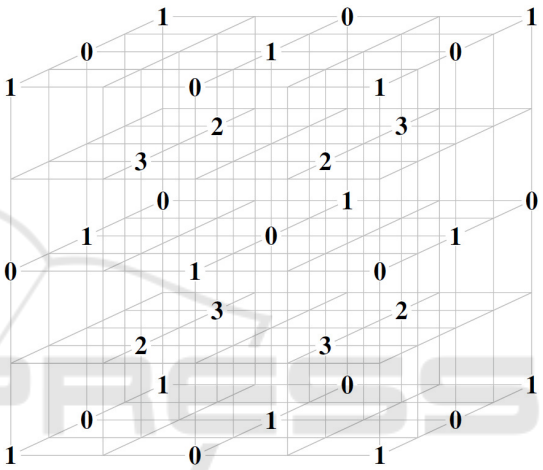


Figure 3: Partition of $\mathbb{B}$ into four subfields. All points marked 'i' are in subfield $S_4(i)$ $(i = 0, 1, 2, 3)$. (Note that unmarked elements in $\mathbb{Z}^3$ are not points in $\mathbb{B}$).

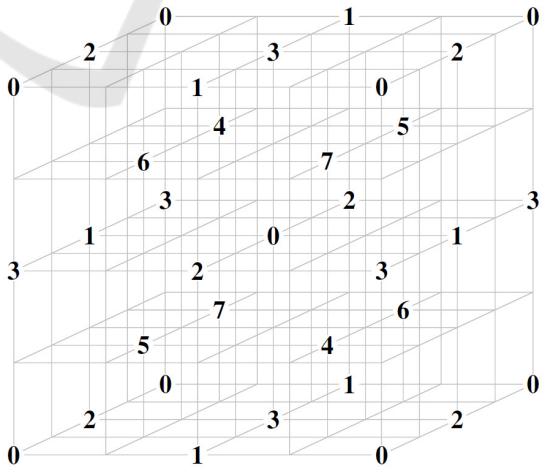


Figure 4: Partition of $\mathbb{B}$ into eight subfields. All points marked 'i' are in subfield $S_8(i)$ $(i = 0, 1, \ldots, 7)$. (Note that unmarked elements in $\mathbb{Z}^3$ are not points in $\mathbb{B}$).

3 NEW SUBFIELD-BASED PARALLEL THINNING ALGORITHMS

In this section, two parallel algorithms are reported:

- **SF-4-TK**: 4-subfield kernel-thinning algorithm,
- **SF-8-TK**: 8-subfield kernel-thinning algorithm.

Algorithm 1 gives the two kernel-thinning algorithms.

Algorithm 1: Algorithm **SF-k-TK** ($k = 4, 8$).

Input: picture $(\mathbb{B}, 14, 14, X)$
Output: picture $(\mathbb{B}, 14, 14, Y)$
// initialize resulted black points
$Y \leftarrow X$
repeat
  // initialize deleted points
  $D \leftarrow \emptyset$
  // k subiterations
  for $i \leftarrow 0$ **to** $k-1$ **do**
    // collect deletable points
    $D(i) \leftarrow \{\ p \mid p \in S_k(i) \cap Y$ and simple $\}$
    // parallel deletion
    $Y \leftarrow Y \setminus D(i)$
    // collect deleted points
    $D \leftarrow D \cup D(i)$
until $D = \emptyset$

In Algorithm 1 the kernel of the **repeat** cycle corresponds to one iteration step that comprises k ($k = 4, 8$) subiterations (i.e., parallel reductions) in which the k subfields (see Fig. 3 and Fig. 4) are alternatively activated. The thinning process is terminated if no points are deleted within an iteration step (i.e., $D = \emptyset$).

Note that a general and computationally efficient implementation scheme for parallel thinning algorithms was proposed in (Palágyi, 2008). This scheme takes advantage of the fact that all thinning algorithms may delete only border points. Thus we do not have to examine the deletability of interior points, and the repeated scans of the entire array (that stores the actual picture) can be avoided by using a linked list that stores the set of border points to be evaluated for possible deletion in the actual thinning phase.

Notice that only simple points can be deleted in each subiteration of all the proposed subfield-based parallel thinning algorithms (see Algorithm 1). Thus we can state the following:

Theorem 4. *Algorithms* **SF-4-TK** *and* **SF-8-TK** *are both topology-preserving.*

Proof. By Theorem 3, it is obvious that each subiteration (i.e., parallel reduction) of our algorithms is topology-preserving.

Since the proposed subfield-based parallel kernel-thinning algorithms are composed of topology-preserving reductions, the entire algorithms are topology-preserving. □

We remark that in our kernel-thinning algorithms (see Algorithm 1) we do not need to use any geometric constraint.

Note that we follow a safe technique for designing topology-preserving parallel thinning algorithms: some sufficient conditions for topology-preserving reductions are combined with parallel thinning strategies and geometric constraints (Palágyi et al., 2012).

Lastly, since the output of Algorithm 1 does not contain any simple point, the presented algorithms are indeed kernel-thinning ones:

Proposition 2. *Algorithms* **SF-4-TK** *and* **SF-8-TK** *extract the topological kernel of the input object.*

4 RESULTS

The proposed two algorithms were tested on objects of different shapes. We present ten illustrative examples below (see Figs. 5-14). The numbers in parentheses under the original images are the counts of object points. The pairs of numbers in parentheses under the produced topological kernels indicate the counts of object points and the numbers of the required iterations, respectively. Notice that the algorithms transform the synthetic object with no hole in Fig. 5 to a single voxel, while the topological kernels of holey objects in Figs. 6-14 are one-voxel thin connected closed curve segments. Notice that in Fig. 14, our algorithms produced exactly the same skeleton-like features, and even the sufficient number of iterations are equal.

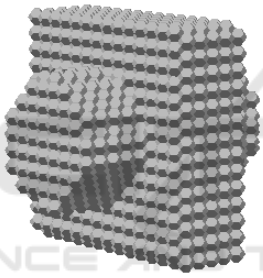
Our implementations were run on a usual desktop (HP ProDesk 400 G4; 3.20 GHz Intel Core i5-6500; Windows 10 x64) and written in C++. The `std::chrono` library was used for runtime measurement. Table 1 contains the computation times of our algorithms for each object shown in Figs. 5-14. Note that reading the input image and writing the output image were not considered here. We can observe that the computational cost does not depend on the number of subfields, due to the fact that any border point belongs to exactly one subfield. Thus, any of those points is examined exactly once in each iteration.

Since the authors do not know any existing algorithm acting on the BCC grid for producing topological kernels, we can not compare the results of **SF-4-TK** and **SF-8-TK** with competing methods. Note that topological kernels could be produced by sequential

deletion of simple points and by various skeletonization approaches, but these strategies are out of scope of this work.

Table 1: Computation times (in millisec.) of Algorithm 1.

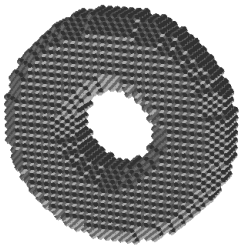
Test object	SF-4-TK	SF-8-TK
Syntetic object	0.993	0.992
Torus	1.994	1.995
Gear	36.869	36.889
Hand	31.947	32.897
Letter A	9.942	9.954
'Fertility'	29.908	29.920
Helicopter	38.890	38.895
Cube with 1 hole	37.866	37.890
Cube with 2 holes	33.914	33.909
Cube with 3 holes	31.891	31.902



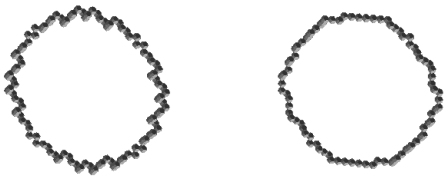
(2 806)

(1, 8) (1, 10)

Figure 5: A $32 \times 32 \times 32$ image of a synthetic object (top) and its topological kernels produced by algorithms **SF-4-TK** (bottom left) and **SF-8-TK** (bottom right).

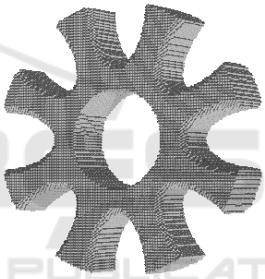


(10 028)



(112, 7) (84, 8)

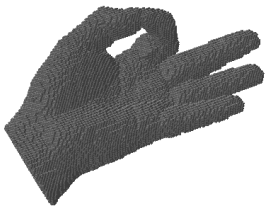
Figure 6: A $64 \times 64 \times 19$ image of a synthetic object (top) and its topological kernels produced by algorithms **SF-4-TK** (bottom left) and **SF-8-TK** (bottom right).



(149 142)

(218, 25) (234, 27)

Figure 7: A $45 \times 191 \times 191$ image of a gear (top) and its topological kernels produced by algorithms **SF-4-TK** (bottom left) and **SF-8-TK** (bottom right).



(113 751)

(151, 51)

(165, 43)

Figure 8: A $191 \times 96 \times 114$ image of a hand (top) and its topological kernels produced by algorithms **SF-4-TK** (bottom left) and **SF-8-TK** (bottom right).

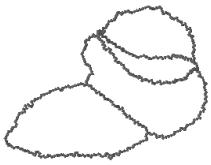
(41 964)

(109, 17)

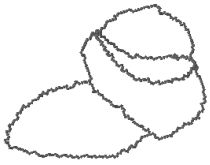
(109, 20)

Figure 9: A $100 \times 100 \times 40$ image of a letter (top) and its topological kernels produced by algorithms **SF-4-TK** (bottom left) and **SF-8-TK** (bottom right).

(100 188)

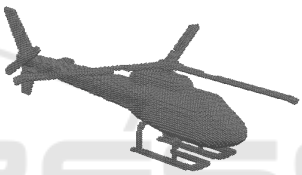


(471, 18)



(459, 17)

Figure 10: A $138 \times 70 \times 189$ image of 'fertility' (top) and its topological kernels produced by algorithms **SF-4-TK** (bottom left) and **SF-8-TK** (bottom right).

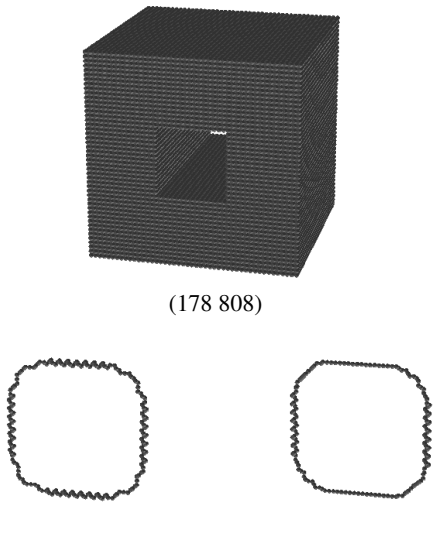


(68 432)

(304, 85)

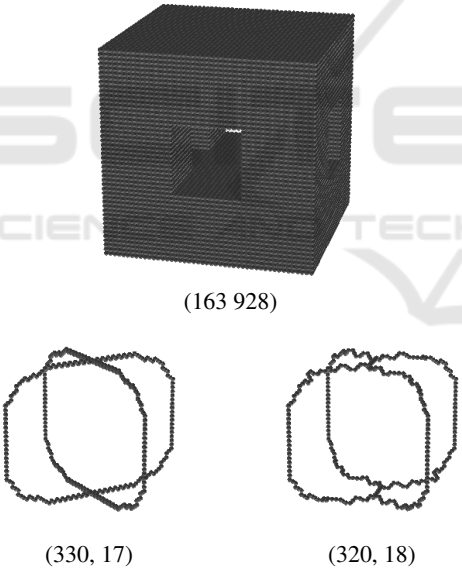
(272, 72)

Figure 11: A $381 \times 103 \times 255$ image of a helicopter (top) and its topological kernels produced by algorithms **SF-4-TK** (bottom left) and **SF-8-TK** (bottom right).



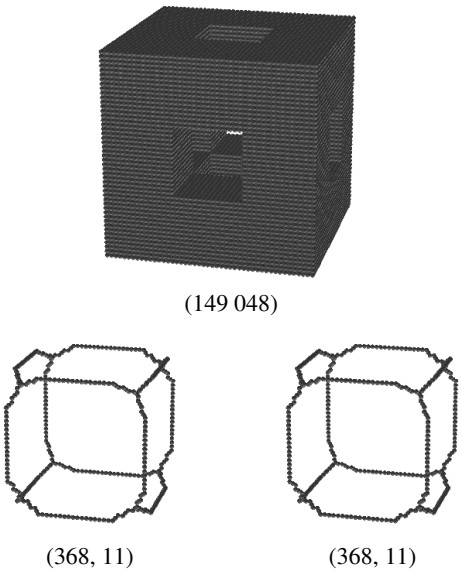
(178 808) (184, 20) (156, 21)

Figure 12: A $93 \times 93 \times 93$ image of a holey cube (top) and its topological kernels produced by algorithms **SF-4-TK** (bottom left) and **SF-8-TK** (bottom right).



(163 928) (330, 17) (320, 18)

Figure 13: A $93 \times 93 \times 93$ image of a holey cube with more holes (top) and its topological kernels produced by algorithms **SF-4-TK** (bottom left) and **SF-8-TK** (bottom right).



(149 048) (368, 11) (368, 11)

Figure 14: A $93 \times 93 \times 93$ image of a cube with even more holes (top) and its topological kernels produced by algorithms **SF-4-TK** (bottom left) and **SF-8-TK** (bottom right).

5 CONCLUSIONS

In this paper, the very first 3D kernel-thinning algorithms on the BCC grid are proposed. These parallel algorithms fall into the subfield-based category, and they can produce topological kernels. The topological correctness of our algorithm is guaranteed by the safe designing technique: a sufficient condition for topology-preserving reductions is combined with a parallel thinning strategy.

In a future work, we plan to construct new symmetric and asymmetric point-based sufficient conditions for topology preservation. Combining these conditions with all types of parallel thinning methods and various geometric constraints yields a variety of parallel 3D curve-, surface-, and kernel-thinning algorithms acting on the BCC grid.

REFERENCES

Čomić, L. and Nagy, B. (2016). A combinatorial coordinate system for the body-centered cubic grid. *Graphical Models*, 87:11–22.

Cornea, N., Silver, D., and Min, P. (2007). Curve-skeleton properties, applications, and algorithms. *IEEE Transactions on Visualization and Computer Graphics*, 13:530–548.

Csébfalvi, B. (2013). Cosine-weighted b-spline interpolation: A fast and high-quality reconstruction scheme for the body-centered cubic lattice. *IEEE Transactions*

on Visualization and Computer Graphics, 19:1455–1466.

Hall, R. W. (1996). Parallel connectivity-preserving thinning algorithms. In Kong, T. Y. and Rosenfeld, A., editors, *Topological algorithms for digital image processing*, pages 145–179. Elsevier Science.

Hall, R. W., Kong, T. Y., and Rosenfeld, A. (1996). Shrinking binary images. In Kong, T. Y. and Rosenfeld, A., editors, *Topological algorithms for digital image processing*, pages 31–98. Elsevier Science.

Kardos, P. (2021). Topology preservation on the bcc grid. *Journal of Combinatorial Optimization*.

Kong, T. Y. (1995). On topology preservation in 2-d and 3-d thinning. *International Journal of Pattern Recognition and Artificial Intelligence*, 9:813–844.

Kong, T. Y. and Rosenfeld, A. (1989). Digital topology: Introduction and survey. *Computer Vision, Graphics, and Image Processing*, 48:357–393.

Matej, S. and Lewitt, R. M. (1995). Efficient 3d grids for image reconstruction using spherically-symmetric volume elements. *IEEE Transactions on Nuclear Science*, 42:1361–1370.

Palágyi, K. (2008). A 3d fully parallel surface-thinning algorithm. *Theoretical Computer Science*, 406:119–135.

Palágyi, K., Németh, G., and Kardos, P. (2012). Topology preserving parallel 3d thinning algorithms. In Brimkov, V. E. and Barneva, R. P., editors, *Digital Geometry Algorithms. Theoretical Foundations and Applications to Computational Imaging*, pages 165–188. Springer.

Saha, P. K., Borgefors, G., and Sanniti di Baja, G. (2016). A survey on skeletonization algorithms and their applications. *Pattern Recognition Letters*, 76:3–12.

Saha, P. K., Borgefors, G., and Sanniti di Baja, G., editors (2017). *Skeletonization: Theory, methods and applications*. Academic Press.

Siddiqi, K. and Pizer, S., editors (2008). *Medial Representations – Mathematics, Algorithms and Applications, Computational Imaging and Vision, vol. 37*. Springer, New York.

Sobiecki, A., Jalba, A., and Telea, A. (2014). Comparison of curve and surface skeletonization methods for voxel shapes. *Pattern Recognition Letters*, 47:147–156.

Strand, R. (2004). Surface skeletons in grids with non-cubic voxels. In *Proc. of the 17th International Conference on Pattern Recognition, ICPR 2004*, pages 548–551. Cambridge.

Strand, R. and Brunner, D. (2006). Simple points on the body-centered cubic grid. Technical Report 42, Centre for Image Analysis, Uppsala University, Uppsala, Sweden.

Strand, R. and Nagy, B. (2008). Weighted neighbourhood sequences in non-standard three-dimensional grids - metricity and algorithms. In *Proc. of the 14th IAPR International Conference on Discrete Geometry for Computer Imagery, DGCI 2008*, pages 201–212. Springer, LNCS, vol 4992.

Tagliasacchi, A., Delame, T., Spagnuolo, M., Amenta, N., and Telea, A. (2016). 3d skeletons: A state-of-the-art report. In *Proc. of the 37th Annual Conference of the European Association for Computer Graphics, EG'16*, pages 573–597.

Theussl, T., Möller, T., and Gröller, M. E. (2001). Optimal regular volume sampling. In *Proc. Visualization, VIS'01*, pages 91–98.