# Calculating the Number of Tunnels

Fajie Li<sup>\*</sup> and Reinhard Klette<sup>+</sup>

\* Groningen University, The Netherlands

<sup>+</sup> The University of Auckland, New Zealand

**Abstract.** This report<sup>1</sup> considers 2-regions of grid cubes and proposes an algorithm for calculating the number of tunnels of such a region. The graph-theoretical algorithm proceeds layer by layer; a proof of its correctness is provided, and its time complexity is also given.

# 1 Introduction

Calculations of numbers of tunnels (i.e., of the first Betti number  $\beta_1$ ) have been a subject of interest in 3D digital topology [8], in particular in the context of image analysis (see, e.g., [14, 16, 23]), in graph theory (see, e.g., [28]), or in computational modeling of 3D forms (see, e.g., [9]),

The Euler characteristic  $\chi(K) = \alpha_0 - \alpha_1 + \alpha_2$  of a bounded 3D set K is defined by numbers of cells in a (not uniquely defined) Euclidean surface complex;  $\alpha_0$ is the number of 0-cells (vertices),  $\alpha_1$  the number of 1-cells (edges), and  $\alpha_2$  the number of 2-cells (faces). The Poincaré formula [22]

$$\chi(K) = \beta_0 - \beta_1 + \beta_2$$

allows a calculation based on Betti numbers  $\beta_0$  (number of connected components),  $\beta_1$  (see above), and  $\beta_2$  (number of closed surfaces). (For details, see, for example, Chapter 6 in [12].) The Euler characteristic is used for various purposes, such as for characterizing the topology of cold dense matter at subnuclear density [27] (using integral geometry), in material sciences [26] (using estimates per volume unit), or for studying 3D biomedical structures such as cancellous bone architecture [24]. The number of tunnels is of interest in digital topology; see, e.g., [11, 17, 19, 25]. For a review on different algorithms for calculating the Euler number, and variations in selecting the adjacency model, see [20]. See also Section 5.3.3 in [12].

Obviously, a known Euler number does not yet specify numbers  $\beta_1$  or  $\beta_2$ , even if  $\beta_0 = 1$  can be assumed.

[8] proposed an incremental, theoretically efficient algorithm for calculating Betti numbers. However, now for more than 12 years, no implementation of this algorithm has been documented somewhere on the net. The authors believe that the algorithm, as proposed in this report for calculating  $\beta_1$ , is conceptually easier to follow, and report about their implementation.

<sup>&</sup>lt;sup>1</sup> A revised version of [15].



**Fig. 1.** Left: office tower (note: both open windows contribute to  $\beta_1$ , but could also be closed, and still we could expect to have  $\beta_1 > 0$ ) has  $\beta_0 = 1$ ,  $\beta_1 > 0$ , and  $\beta_2 > 0$ . Middle: the Menger sponge has  $\beta_0 = 1$  and  $\beta_2 = 0$  for any iteration,  $\beta_1 = 9$  tunnels at the start (top), and  $\beta_1 = 209$  tunnels at the first iteration. Right: Sierpinski tetrahedron (start and first iteration), and its sculptural representation.

Figure 1 illustrates the complexity of calculating tunnels. A set K of the complexity of an office tower (left) would be considered 'floor by floor' by our algorithm (assuming an approximating cuboidal model). There can be 'hierarchies of cavities' in such a set K; 'columns' or 'openings between adjacent floors' contribute to the number of tunnels. In contrast, the Menger sponge (already in appropriate cuboidal structure) does only have one closed surface (i.e.,  $\beta_2 = 1$ ). Here, local counts of surface cells (e.g., as in Section 5.3.3 of [12]) are sufficient for identifying the Euler number, and, thus, also the number of tunnels. At the start of the Menger sponge iterations, we (can) have  $\alpha_0 = 40$ ,  $\alpha_1 = 78$ , and  $\alpha_2 = 30$  (note that a Euclidean surface complex is not uniquely defined), and at the first iteration  $\alpha_0 = 680$ ,  $\alpha_1 = 1398$ , and  $\alpha_2 = 510$ . Consequently, we have  $\beta_1 = 9$  (start) or  $\beta_1 = 209$  (first iteration) tunnels.<sup>2</sup> The Sierpinski tetrahedron also has  $\beta_2 = 0$ , for any iteration, and its cuboidal representation (e.g., by Gauss digitization) would allow similar calculations.

This report presents a hierarchical method for computing the number of tunnels of a 3D binary picture (to be precise, of 2-regions of grid cubes). Images are first processed column by column (runs), then layer by layer, and finally those '2D graphs', deduced from individual layers, are merged into a single '3D

 $<sup>^2</sup>$  In the notation of [12], [1] defined a gap by a separator. A 2-region of grid cubes has a tunnel iff it has a 2-gap. Besides this, numbers of tunnels or of 2-gaps (see [5]) are not related to one-another.

graph' in order to compute the number of tunnels. The method is based on the idea of utilizing linear skeletons.

The report starts with a theoretical part which describes how to infer a '3-D' graph from the '2-D' graphs of the layers. This is followed by implementation details, an analysis of time complexity of the presented algorithm, and experiments, illustrating its use. Conclusions are given at the end of the report.

# 2 Definitions

We apply the grid cell model [12], using 0-, 1-, 2-, or 3-cells in 3D space, which are vertices, grid edges, grid squares, or grid cubes, respectively. For example, two 3-cells (*cubes*, for short)  $c_1$  and  $c_2$  are 2-adjacent iff  $c_1 \neq c_2$  and the intersection  $c_1 \cap c_2$  contains a 2-cell.

In general, let S be a countable set and A an adjacency relation on S, defining an *adjacency structure*. If p and q are adjacent (in formal notation: pAq) then p, q are an *adjacency pair*. For any  $p \in S$ , the set  $A(p) = \{q : pAq\}$  is the *adjacency* set of p. Maximal connected subsets of S are components of S. An adjacency structure [S, A] is called an *adjacency graph* iff it has the following properties: A(p) is finite for any  $p \in S$ ; S is connected with respect to A; and any finite subset  $M \subseteq S$  has at most one infinite complementary component. Any finite component of an adjacency graph is called a *region*. The set A(M) of all nodes adjacent to  $M \subseteq S$  is called the *adjacency set* of M. If [S, A] is an adjacency graph, two disjoint subsets  $M_1$  and  $M_2$  of S are called *adjacent* iff  $A(M_1) \cap M_2 \neq \phi$ , denoted by  $M_1 \not A M_2$  or  $(M_1, M_2) \in \mathcal{A}$ . Let  $\mathcal{R}$  be a partition of S into regions without including the infinite background component. The undirected graph  $[\mathcal{R}, \mathcal{A}]$  is the *region adjacency graph* of [S, A]. For example, 2-adjacency of cubes defines 2-regions and related adjacency graphs of regions of cubes.

Following Section 4.2.2 of [12], the merging of two adjacent nodes p and qin a graph G = [S, A] is defined by replacing p and q with a new node r that is adjacent to every node in  $S \setminus \{p, q\}$  to which p or q was originally adjacent. A finite sequence of merging operations is called a *contraction*; see Figure 2. If graph  $G_1$  is contracted into  $G_2$ , then  $G_1$  and  $G_2$  are homeomorphic.

The *linear skeleton* (see Section 6.3.3 of [12]) of a set  $M \subseteq \mathbb{R}^3$  is defined by continuos contractions. We transform a linear skeleton into a graph by having a



**Fig. 2.**  $G_1$  is contracted into  $G_4$ ;  $G_4$  can be further contracted into a loop, but not into a single node (without any edge).



Fig. 3. Two 2-regions of cubes with homotopic graphs.  $S_1$  does not contain cube  $c_1$ , and  $S_2$  does not contain cubes  $c_2$  and  $c_3$ .

node at each of its singular points (see Section 7.1.3), and arcs between singular points define the edges.

**Definition 1.** Let S be a 2-region of cubes and G a graph. G is a homotopic graph of S iff G is homeomorphic to the linear skeleton of S.

In Figure 3,  $G_1$  and  $G_2$  are both homotopic graphs of  $S_1$ , and  $G_3$  and  $G_4$  are both homotopic graphs of  $S_2$ .

Two relational structures  $[S_1, R_1]$  and  $[S_2, R_2]$  are called *isomorphic* iff there exists a one-to-one mapping f from  $S_1$  onto  $S_2$  such that  $pR_1q$  iff  $f(p)R_2f(q)$ , for all  $p, q \in S_1$ . f is called an *isomorphism*.

**Definition 2.** Let S be a 2-region and G a homotopic graph of S. S is called a minimal cubical set iff G is not isomorphic to any homotopic graph of a 2-region S', where S' is obtained by removing a cube from S.

A *subregion* is a (non-empty) connected subset of a region. Assume a region adjacency graph and a partition of involved regions into subregions; these subregions define a *subregion adjacency graph*.

**Definition 3.** Let L be a set of cubes and H a homotopic graph of L. Let  $\mathcal{R}$  be a partition of L into subregions without including the infinite background component. Let  $[\mathcal{R}, \mathcal{A}]$  be the subregion adjacency graph of  $\mathcal{R}$ . If  $[\mathcal{R}, \mathcal{A}]$  is isomorphic to H, then  $[\mathcal{R}, \mathcal{A}]$  is called a homotopic subregion adjacency graph of L with respect to H, denoted by  $\overline{H}$ .

Let f be the isomorphism; for the whole set V(H) (The set of all vertices in H.), f(V(H)) is denoted by  $\overline{V}(H)$ . Figure 4 shows a layer L of cubes. H is a homotopic graph of L. Let  $\overline{H}$  be a subregion adjacency graph of L, and  $\overline{V}$  the vertices of  $\overline{H}$ . Each vertex  $\overline{v} \in \overline{V}$  is a set of cubes.

In the rest of this report, we denote a graph by G = [V, E], where V = V(G) is the set of nodes and E is the set of edges. If the relation  $\mathcal{A}$  (or set E) is obviously defined for a homotopic subregion adjacency graph  $\overline{H} = [\mathcal{R}, \mathcal{A}] = [V, E]$  then we denote  $\overline{H}$  just by  $V(\overline{H})$  or  $\overline{V}$ .

Let G = [V, E] be a graph and  $V_1 \subseteq V$ . Subgraph  $[V_1, E_1]$  is called the *restriction* of G on  $V_1$ , where  $E_1 = \{v_1v_2 : v_1 \in V_1 \land v_2 \in V_1\}$ . – The following definition is a special case of Definition 2.64 in [13].

**Definition 4.** Let S be a family of cubical sets in kD space, where k = 0, 1, 2, 3. Let X be the set of *i*-cells contained in S, with  $i \leq k$ . Let  $p, q \in X$  such that p is a j-cell, q is a (j-1)-cell and p is the unique j-cell in X incident with q,  $j \leq k$ . Let  $X' = X \setminus \{p, q\}$  and  $S' = \bigcup_{c \in X'\}} c$ . Then S' is called obtained from S via an elementary collapse of p by q.

# 3 Basics

Let S be a family of cubical sets in kD space, where k = 0, 1, 2, 3. Let p be a j-cell, q a (j-1)-cell, for  $j \leq k$ . Let S' be obtained from S via an elementary collapse of p by q. Let  $H_*(S)$  or  $H_*(S')$  be the homology group (see Section 6.4.6 in [12]) of S or S', respectively. Then, by Theorem 2.68 in [13], it follows that

$$H_*(S') \sim H_*(S) \tag{1}$$

where  $\sim$  is the homology (equivalence) relation.

**Lemma 1.** Let G = [V, E] be a connected finite graph. There exists a minimal cubical set S such that G is a homotopic graph of S.

*Proof.* 1. Let |V| = n and  $V = \bigcup_{i=1}^{n} \{v_i\}$ . Let  $d_i$  be the degree of  $v_i \in V$ . Without loss of generality, assume that  $d_i \leq d_{i+1}$ , for i = 1, 2, ..., n - 1.

2. For each  $v_i$ , construct a simple arc of length  $2d_1 - 1$ , denoted by  $g_{1_i} = \bigcup_{j=1}^{2d_1-1} \{(2i-1,j,1)\}$ , where (2i-1,j,1) are the coordinates of the center of a grid cube, for i = 1, 2, ..., n.



Fig. 4. A layer of cubes and a homotopic subregion adjacency graph with respect to a homotopic graph of it.

3. To connect  $v_1$  to its neighbors, let  $A(v_1) = \{v_{1_1}, v_{1_2}, \ldots, v_{1_{d_1}}\}$  be the adjacency set of node  $v_1$ , with  $1_1 < 1_2 < \ldots < 1_{d_1}$ . Assume that  $j = 1_r$ , where  $r \in \{1, 2, \dots, d_1\}.$ 

If j = 2, then add a cube (2, 1, 1) between  $g_{1_1}$  and  $g_{1_2}$ . Otherwise, let  $g(j, z) = \bigcup_{x=1}^{2j-1} \{(x, j, z)\} \cup \{(1, j, z - 1), (2j - 1, j, z - 1)\}$ , where z is an integer; then add g(j,2) between  $g_{1_1}$  and  $g_{1_j}$ .

4. If  $v_2v_1$  s an edge of G, then let  $d'_2 = d_2 - 1$ . Otherwise, let  $d'_2 = d_2$ . Add

a simple arc of length  $2d'_2$  to  $g_{1_i}$ , denoted by  $g_{2_i} = g_{1_i} \cup (\bigcup_{j=2d_1}^{2d_1+2d_2-1} \{(2i-1,j,1)\})$ , for  $i = 2, 3, \ldots, n$ . Analogous to Step 3, we can connect  $v_2$  to its neighbors except its possible neighbor  $v_1$  which has already been connected to  $v_2$  in Step 3 if  $v_1v_2$  is an edge of G.

5. Analogous to Step 4, we can connect  $v_i$  to its neighbors, where i = $3, 4, \ldots, n.$ 

6. Suppose there are l(i, j) edges between  $v_i$  and  $v_j$  (i < j). Construct l(i, j)simple arcs of length 2(j-i) + 3 as follows: Let  $g(i,j) = \bigcup_{z=3}^{\tilde{l}(i,j)+2} g(j,z)$ .

7. Let S be the union of all cubes constructed so far. If there is a cube c in S such that there is only one cube c' in S such that  $c \cap c'$  is a 2-cell then remove c from S. Repeat this operation until S does not contain such a cube c, and S is as required. 

In Figure 5,  $S_2$  is obtained from  $S_1$  after removing 6 cubes.  $S_2$  is a minimal cubical set and  $G_1$  is a homotopic graph of  $S_2$ .  $S_3$  is a minimal cubical set and  $G_2$  is a homotopic graph of  $S_3$ . – From homology (1) and Lemma 1, we obtain the following:

**Theorem 1.** If a cubical 2-region S is minimal such that a given graph G is a homotopic graph of S, then we have  $\beta_1(S) = \beta_1(G)$ .

We also recall the following theorem by J. W. Alexander (see Theorem 6.8 in [12]).



Fig. 5. Illustration for Lemma 1.



Fig. 6. Illustration to the proof of Lemma 2.

**Theorem 2.** Let  $S_1$  and  $S_2$  be two Euclidean complexes defined by partitions of polyhedra  $\cup K_1$  and  $\cup K_2$ . If  $\cup K_1$  and  $\cup K_2$  are homeomorphic then  $K_1$  and  $K_2$  have the same Betti numbers.

**Corollary 1.** If S is a 2-connected cubical set and G a homotopic graph of S. Then the first Betti number of S is equal to the Betti number of G.

*Proof.* Let S' is a minimal cubical 2-region such that G is a homotopic graph of S. Then S' is homeomorphic to S. Theorems 1 and 2 prove the corollary.  $\Box$ 

We recall that a sequence  $(p_0, p_1, \dots, p_n)$  of 2-cells is a 1-path of 2-cells iff  $p_{i+1}$  is 1-adjacent to  $p_i$ , for  $0 \le i \le n-1$ .  $p_0, p_n$  are the endnodes of this path.

Let S be a 2-region of cubes with nonempty intersections with grid layers L. Let  $S_i$  be a component of the intersection of S with layer L,  $H_i$  a homotopic graph of  $S_i$ ,  $\bar{H}_i$  be a homotopic subregion adjacency graph of  $S_i$  with respect to  $H_i$ , and let  $\bar{u}_i, \bar{v}_i, \bar{w}_i \in \bar{H}_i$ , for  $i = 1, \ldots, i_{max}$ .

**Lemma 2.** If there exists a 1-path of 2-cells  $(p_0, p_1, \dots, p_n)$  such that  $p_0$  is contained in  $\bar{u}_1 \cap \bar{u}_2$  and  $p_n$  is contained in  $\bar{w}_1 \cap \bar{w}_2$ , and  $p_i$  belongs to S, where  $i = 0, 1, \dots, n$ , then the set of cubes  $(\bar{u}_1 \cup \bar{u}_2) \cup (\bar{w}_1 \cup \bar{w}_2)$  is simply connected.

*Proof.* Let g be a closed curve contained in  $\bar{u}_1 \cup \bar{u}_2 \cup \bar{w}_1 \cup \bar{w}_2$ . Let  $P_2$  be an arc of g such that  $P_2$  is contained in  $\bar{u}_2 \cup \bar{w}_2$  and  $P_2 \cap (p_0, p_1, \dots, p_n) = \{A, B\}$ . Since  $\bar{u}_2 \cup \bar{w}_2$  is connected then (see Figure 6) arc  $AP_2B$  can be continuously contracted into line segment  $AP_1B \subset p_0, p_1, \dots, p_n$ . Therefore, within  $\bar{u}_1 \cup \bar{u}_2 \cup \bar{w}_1 \cup \bar{w}_2, g$  can be continuously contracted into a line segment and then into a single point in  $(p_0, p_1, \dots, p_n)$ .

Now consider a pair  $(v_1, v_2)$ , where  $v_i \in V(H_i)$  and  $\bar{v}_i \in V(\bar{H}_i)$ , for i = 1, 2. Let c be the number of components of  $\bar{v}_1 \cap \bar{v}_2$  (It is a set of 2-cells.).

Operation 1. If  $c \ge 1$  then add c edges between  $v_1$  and  $v_2$  to graph  $H_1 \cup H_2$ . Let E be the set of all edges added. Let  $V_i$  be the set of such  $v_i \in V(H_i)$ . In other words,  $V_1 = \{v_1 : v_1 \in V(H_1) \land v_1 v_2 \in E\}$  and  $V_2 = \{v_2 : v_2 \in V(H_2) \land v_1 v_2 \in E\}$ . Then  $V_i \subseteq V(H_i)$ , for i = 1, 2.

Operation 2. Let  $\overline{U}_i$  be the restriction of  $\overline{H}_i$  on  $\overline{V}_i$ , for i = 1, 2. For each pair  $(\overline{u}_i, \overline{w}_i)$ , where  $\overline{u}_i, \overline{w}_i \in \overline{U}_i$ , let g be the number of 1-paths of 2-cells  $(p_0, p_1, \dots, p_n)$  such that  $p_0$  is contained in  $\overline{U}_1 \cap \overline{U}_2$  and  $p_n$  is contained in  $\overline{w}_1 \cap \overline{w}_2$ . See Figure 7 for an example.



Fig. 7. Example with g = 3.

(Note that  $\bar{u}_1 \cap \bar{u}_2$  and  $\bar{w}_1 \cap \bar{w}_2$  are also sets of 2-cells.) Delete g edges between  $u_1$  and  $w_1$  in  $H_1$ .

Let  $(H_1 \cup H_2)$  be the resulting graph after applying Operations 1 and 2 on  $H_1 \cup H_2$ . Then we have

# **Lemma 3.** $(H_1 \cup H_2)$ is a homotopic graph of the set $S_1 \cup S_2$ of cubes.

*Proof.* Since  $H_i$  is a homotopic graph of  $S_i$ , it follows that  $S_i$  can be continuously contracted into  $H_i$ , where i = 1, 2. The edges, added between  $H_1$  and  $H_2$ , are not redundant because when  $S_i$  is contracted continuously, the subset of cubes  $\bar{v}_1 \cup \bar{v}_2$  must also be contracted or extracted continuously. Therefore, the only possible redundant edges (because of duplicating) may be edges such as  $u_i w_i$  in  $H_i$ , for i = 1, 2. Together with Lemma 2, this proves the lemma.

Let  $H = \bigcup_{i=1}^{l} H_i$ . Repeat Operations 1 and 2 on  $H_i \cup H_{i+1}$  until Stop (this is guaranteed to happen, after a finite number of steps), where  $i = 1, 2, \ldots, l$ -1. Let the resulting graph be (H). Then, by Lemma 3, we have the following:

**Theorem 3.** (H) is a homotopic graph of S.

The following algorithm is justified by this theorem.

### 4 Subprocess and Algorithm

At first we present an algorithm for producing a homotopic subregion adjacency graph and a homotopic graph, for any given layer of cubes within the given 2-region. This is then used to describe our main algorithm.

#### 4.1 Preparation: Runs

A layer of cubes can be decomposed into a number of columns (of cubes); cubes in the same column have the same x-coordinate. A column of cubes can then be decomposed into a number of runs (i.e., runs) of cubes. Cubes in the same run have consecutive y-coordinates (i.e., each run is a simple cube-arc). Each run can be represented by its two endcubes.

A run  $I_1$  is *left-(right-)adjacent* to a run  $I_2$  if the x-coordinate of  $I_1$  is less (greater) than that of  $I_2$ , and there exists a cube  $c_i \in I_i$  such that  $c_1 \cap c_2$  is a 1-cell.

Each left- or right-adjacent run of run I is called a *neighbor* of I. A *run-path* is a sequence of runs  $(i_0, i_1, \dots, i_n)$  such that  $i_{j+1}$  is the unique right neighbor of  $i_j$   $(0 \le j \le n-1 \text{ and } n \ge 1)$ ;  $i_0$  and  $i_n$  are called the *endruns* of the path. The endrun with smaller (larger) x-coordinate is called the first (second) endrun. A *maximal* run-path is a run-path such that the number of left neighbors of its first endrun is not one, and the number of right neighbors of its second endrun is also not one. If the number the neighbors of the second endrun of a run-path  $p_I$  is greater than one, then it is called the *head-run* of  $p_I$ , denoted by  $h_{p_I}$ . Otherwise the first endrun of  $p_I$  is called the head-run of  $p_I$ , is called the *tail-run* of  $p_I$ . We remove the tail-run from the run-path if the number of its the neighbors is greater than one. The resulting run-path is called the *reduced* maximal run-path. Thus, if the number of neighbors of the tail-run is one, then the reduced maximal run-path of a run-path coincides with itself.

In Figure 4, the run in column 18 (the cubes in this run have x-coordinate 18 and y-coordinates between 1 and 20) has 3 left neighbors and 4 right neighbors. There are two maximal run-paths from column 3 to column 18. The run in column 3 is the first end- and tail-run of both, and the run in column 18 is the second end- and the head-run of both. With the tail-run removed, they induce two reduced maximal run-paths from column 4 to column 18. There is also a maximal run-path from column 5 to 11 (i.e., a disconnected "island" in the largest of the three holes). The run in column 5 is the first end- and head-run of it, and the run in column 11 is the second end- and tail-run of it. It coincides with its reduced maximal run-path (The tail-run in column 11 is not removed because it has only one neighbor).

#### 4.2 Single-Layer Subprocess

For a single layer of cubes, do the following:

- 1. Get the set of all runs, denoted by  $S_I$ .
- 2. For each run in  $S_I$ , get and set its left and right neighbors.
- 3. Get the set of all maximal run-paths, denoted by  $S_P$ .
- 4. For each run-path in  $S_P$ , get and set its head- and tail-runs.
- 5. Get the set of all reduced maximal run-paths from  $S_P$ , denoted by  $S_{RP}$ .

6. Get the set of all runs, denoted by  $S_H$  such that for each run h in  $S_H$ , the number of neighbors of h is greater than two, or the number of left neighbors of h equals two and the number of right neighbors of h equals zero, or the number of left neighbors of h equals zero and the number of right neighbors of h equals two.

7. Get the set of head-runs of all run-paths in  $S_{RP}$ , denoted by  $S_{PH}$ .

8. For each h in  $S_H \cup S_{PH}$ , reset its neighbors as follows:

8.1. For each pair  $h_1$ ,  $h_2$  in  $S_H \cup S_{PH}$ , if they have a common neighbor I, remove I in the set of neighbors of  $h_i$  where the index i satisfies  $x_i < x_{i+1} \pmod{2}$ , where  $x_i$  is the x-coordinate of  $h_i$ , for i = 1, 2.

8.2. For each h in  $S_H$ , remove its neighbor if this neighbor is contained in a run-path in  $S_{RP}$  (Note that 8.2 and 8.1 are independent of each other because, by definition, a run-path has at least two runs).

9. For each head-run h in  $S_H \cup S_{PH}$ , let  $C_h = C \cup C_N \cup C_p$ , where C is the set of all cubes contained in h,  $C_N$  is the set of all cubes contained in neighbors of h, and  $C_p$  is the set of all cubes contained in reduced maximal run-paths belonging to h (i.e.,  $C_P = \{c : c \in p \in S_{RP} \land p \text{ belongs to } h\}$ ).

10. Get a homotopic subregion adjacency graph  $\overline{H}$  by  $S_H \cup S_{PH}$ , where each subregion is a set  $C_h$  as obtained in Step 9;

11. Get a homotopic graph H from H.

Section 5.1 illustrates this algorithm by means of an example. We use now this single-layer subprocess for describing the main procedure, which computes the number of tunnels of a 2-region of cubes.

#### 4.3 Algorithm

Assume a 2-region S of cubes. We proceed as follows:

1. Decompose S into layers, defined by all those z-coordinates with cells in S. (Each *layer* is thus the subset of S defined by an identical z-coordinate.) Let  $L_i$  be the *i*-th layer of S, where i = 1, 2, ..., m, and m is the number of layers of S.

2. By the single-layer subprocess, construct a homotopic graph  $H_i$  for each layer  $L_i$ , and a homotopic subregion adjacency graph,  $\bar{H}_i$ , with respect to  $H_i$ .

3. Without loss generality, let  $L_1$  and  $L_2$  be any two consecutive layers. Let  $H_1$  or  $H_2$  be homotopic graphs of  $L_1$  or  $L_2$ , respectively. Let  $\bar{H}_1$  or  $\bar{H}_2$  be homotopic subregion adjacency graphs with respect to  $H_1$  or  $H_2$ , respectively. For each pair  $(v_1, v_2)$ , where  $v_1 \in V(H_1)$ ,  $v_2 \in V(H_2)$ ,  $\bar{v}_1 \in V(\bar{H}_1)$ , and  $\bar{v}_2 \in V(\bar{H}_2)$ , do the following:

3.1. Apply the FILL algorithm (see, e.g., [12]) to find the number of components of  $\bar{v}_1 \cap \bar{v}_2$  (which is a set of 2-cells), denoted by c.

3.2. If  $c \geq 1$ , then add c edges between  $v_1$  and  $v_2$  in graph  $H_1 \cup H_2$ . Let E be the set of all edges added. Let  $V_i$  be the set of these  $v_i \in V(H_i)$ . In other words,  $V_1 = \{v_1 : v_1 \in V(H_1) \land v_1 v_2 \in E\}$  and  $V_2 = \{v_2 : v_2 \in V(H_2) \land v_1 v_2 \in E\}$ . Then  $V_i \subseteq V(H_i)$ , for i = 1, 2.

3.3. Let  $\overline{U}_i$  be the restriction of  $\overline{H}_i$  on  $\overline{V}_i$ , where i = 1, 2. For each pair  $(\overline{u}_i, \overline{w}_i)$ , where  $\overline{u}_i, \overline{w}_i \in \overline{U}_i$ , apply the FILL algorithm to find the number of 1-paths of 2-cells  $(p_0, p_1, \dots, p_n)$  such that  $p_0$  is contained in  $\overline{u}_1 \cap \overline{u}_2$ , and  $p_n$  is contained in  $\overline{w}_1 \cap \overline{w}_2$  (where both  $\overline{u}_1 \cap \overline{u}_2$  and  $\overline{w}_1 \cap \overline{w}_2$  are again sets of 2-cells), denoted by p. Delete p edges between  $u_1$  and  $w_1$  in  $H_1$  (see Lemma 2).

4. Apply Step 3 to every pair  $L_i$  and  $L_{i+1}$  of two consecutive layers, for  $i = 1, 2, \ldots, i_{max} - 1$ . Let G be the resulting graph. Then the number of tunnels of S equals E(G) - V(G) + 1, see Equation (2).

For this final step we make use of a basic result in combinatorial graph theory (see, for example, [4, 28]). Let G be a graph with n nodes, m edges and k connected components. Then we have

$$\beta_1(G) = m - n + k \tag{2}$$

A proof of this equation is straightforwardly by mathematical induction on the number of edges: a new edge either increments the number of 1-cycles or decrements the number of connected components.

### 5 Examples

#### 5.1 An Example for the Single-Layer Subprocess

For an implementation in C++ or Java,  $S_I$ ,  $S_P$ ,  $S_{RP}$ ,  $S_H$ ,  $S_{PH}$ , and neighbors of a run can be represented by *Vector Objects*. Each run is represented by a *Class*. We discuss the layer of cubes as shown in Figure 4.

Step 1. Column 18 has only one run (with 20 cubes). Column 0 has three runs. There are 62 runs altogether.

Step 2. A Vector Object in C++ or Java solves this tasks straightforwardly.

Step 3. Simply go through each run I in  $S_I$  to get the maximal run-path which contains I, then take the union of these maximal run-paths to get  $S_P$ . There are seven elements in  $S_P$ . Apart from the three already described in Section 4, there are two more maximal run-paths from column 0 to 3 and another two from column 14 to 17.

Step 4. For each run-path, its head- and tail-runs can be found and set by checking the number of neighbors of its endruns. The two maximal run-paths from column 0 to 3 have their head-run in column 3 and tail-run in column 0 while the two from column 14 to 17 have their head-run in column 17 and tail-run in column 14.

Step 5.  $S_{RP}$  (a Vector Object) can remove an element easily if necessary. Corresponding to the four elements in  $S_P$  from Step 4, we remove two reduced maximal run-paths, from column 1 to 3, with tail-run in column 0. Another two reduced maximal run-paths from column 15 to 17 have their head-run in column 17, with tail-run in column 14 removed (because it has 3 neighbors; so it must be removed).

Step 6. Each run in  $S_H$  will later correspond to, or induce a subregion. A run is put into  $S_H$  iff it has more than 3 neighbors, or it is "the end of a cycle" of runs.

Step 7. Each run in  $S_{PH}$  will also later correspond to, or induce a subregion.

Step 8. This step ensures that each run must be a neighbor of only one other run, or contained in only one run-path of only one element in  $S_H \cup S_{PH}$ . The



Fig. 8. Three input examples, used for illustrating the main algorithm.

run in column 2 with y-coordinate 3 is now a left neighbor of the run in column 3. It is no longer a right neighbor of the run in column 1. Moreover, the run in column 4 with y-coordinate 3 is now contained in a reduced maximal run-path with the head-run in column 18. It is no longer a right neighbor of the run in column 3.

Step 9. In this step we get all those cubes "represented" (or "dominated") by an element in  $S_H \cup S_{PH}$ . If we think of these cubes as being "supporters" of an element in  $S_H \cup S_{PH}$ , then the head-run in column 0 (with *y*-coordinates from 7 to 19) has 13 supporters, the head-run in column 1 (with *y*-coordinates from 0 to 5) has 9 supporters, and the head-run in column 3 (with *y*-coordinates from 2 to 19) has 31 supporters. Each cube must be a supporter, and can only support one head-run.

Step 10. Each of the polygons shown in Figure 4 contains cubes of one  $C_h$  obtained in Step 9. For any pair  $C_{h_1}$  and  $C_{h_2}$ , obtained in Step 9, let k be the number of 2-cells in  $C_{h_1} \cap C_{h_2}$ . Then there will be k edges between  $C_{h_1}$  and  $C_{h_2}$  in the homotopic subregion adjacency graph  $\overline{H}$  shown in Figure 4, where  $\overline{v}_i$  represents  $C_{h_i}$ , for i = 1, 2, ..., 7.

Step 11. See H in Figure 4.

#### 5.2 Examples for the Main Algorithm

Each cube is represented (in the grid-point model) by the coordinates (x, y, z) of its center, where x, y, and z are integers. Consequently, each 2-cell is represented by the coordinates (x, y, z) of its center, where two of them are integers, and the other one is a half-integer *i*.5. Figure 8 shows the following five sets of cubes:

Let  $g_1 = \{ (1, 0, 0), (2, 0, 0), (3, 0, 0), (4, 0, 0), (5, 0, 0), (6, 0, 0), (6, 1, 0), (6, 2, 0), (6, 3, 0), (6, 4, 0), (1, 1, 0), (1, 2, 0), (1, 3, 0), (1, 4, 0), (1, 5, 0), (2, 5, 0) \}$ 

0), (3, 5, 0), (4, 5, 0), (5, 5, 0), (6, 5, 0) }.  $g_1$  is a simple 2-curve of 20 cubes in layer 2.

Let  $g_2 = \{ (9, 0, 0), (10, 0, 0), (11, 0, 0), (12, 0, 0), (13, 0, 0), (14, 0, 0), (14, 1, 0), (14, 2, 0), (14, 3, 0), (14, 4, 0), (9, 1, 0), (9, 2, 0), (9, 3, 0), (9, 4, 0), (9, 5, 0), (10, 5, 0), (11, 5, 0), (12, 5, 0), (13, 5, 0), (14, 5, 0) \}$ .  $g_2$  is also a simple 2-curve of 20 cubes in layer 2.

Let  $g_3 = \{ (3, 2, 1), (4, 2, 1), (5, 2, 1), (6, 2, 1), (7, 2, 1), (8, 2, 1), (9, 2, 1), (10, 2, 1), (11, 2, 1), (12, 2, 1), (12, 3, 1), (12, 4, 1), (12, 5, 1), (12, 6, 1), (12, 7, 1), (12, 8, 1), (3, 3, 1), (3, 4, 1), (3, 5, 1), (3, 6, 1), (3, 7, 1), (3, 8, 1), (4, 8, 1), (5, 8, 1), (6, 8, 1), (7, 8, 1), (8, 8, 1), (9, 8, 1), (10, 8, 1), (11, 8, 1) \}.$   $g_3$  is a simple 2-curve of 30 cubes in layer 1.

Let  $g_4 = \{ (7, 2, 0), (8, 2, 0) \} = \{ c_1, c_2 \}$ .  $g_4$  is a simple 2-arc of two cubes in layer 2. and

Let  $g_5 = \{ (7, 5, 0), (8, 5, 0) \} = \{ c_3, c_4 \}$ .  $g_5$  is also a simple 2-arc of two cubes in layer 2.

**Example 1.** We consider  $S_1 = L_1 \cup L_2$ , where  $L_1 = g_3$  and  $L_2 = g_1 \cup g_2$ .  $\bar{H}_1 = [\bar{V}_1, \bar{E}_1]$ , where  $\bar{V}_1 = \{\bar{v}_3, \bar{v}_4\}$ ;  $\bar{H}_2 = [\bar{V}_2, \bar{E}_2]$ , where  $\bar{V}_2 = \{\bar{v}_1, \bar{v}_2, \bar{v}_5, \bar{v}_6\}$ , with  $\bar{v}_1 = \{ (10, 0, 0), (11, 0, 0), (12, 0, 0), (13, 0, 0), (14, 0, 0), (14, 1, 0), (14, 2, 0), (14, 3, 0), (14, 4, 0), (14, 5, 0) \}$ 

 $\bar{v}_2 = \{ (9, 0,0), (9, 1, 0), (9, 2, 0), (9, 3, 0), (9, 4, 0), (9, 5, 0), (10, 5, 0), (11, 5, 0), (12, 5, 0), (13, 5, 0) \}$ 

 $\bar{v}_5 = \{ (6, 0, 0), (6, 1, 0), (6, 2, 0), (6, 3, 0), (6, 4, 0), (6, 5, 0), (2, 5, 0), (3, 5, 0), (4, 5, 0), (5, 5, 0) \}$ 

 $\bar{v}_6 = \{ (1, 0, 0), (2, 0, 0), (3, 0, 0), (4, 0, 0), (5, 0, 0), (1, 1, 0), (1, 2, 0), (1, 3, 0), (1, 4, 0), (1, 5, 0) \}$ 

Sets  $\bar{v}_1$ ,  $\bar{v}_2$ ,  $\bar{v}_5$  and  $\bar{v}_6$  are simple 2-arcs, each of ten cubes in layer 2. Furthermore  $\bar{v}_3 = \{ (4, 8, 1), (5, 8, 1), (6, 8, 1), (7, 8, 1), (8, 8, 1), (9, 8, 1), (10, 8, 1), (11, 8, 1), (12, 8, 1), (12, 2, 1), (12, 3, 1), (12, 4, 1), (12, 5, 1), (12, 6, 1), (12, 7, 1) \}$  $\bar{v}_4 = \{ (3, 2, 1), (3, 3, 1), (3, 4, 1), (3, 5, 1), (3, 6, 1), (3, 7, 1), (3, 8, 1), (4, 2, 1), (1$ 

1), (5, 2, 1), (6, 2, 1), (7, 2, 1), (8, 2, 1), (9, 2, 1), (10, 2, 1), (11, 2, 1) } Sets  $\bar{v}_3$  and  $\bar{v}_4$  are simple 2-arcs, each containing 15 cubes in layer 1.

For Step 3.2 of the algorithm, see Figure 8 or the simple calculation  $\bar{v}_4 \cap \bar{v}_2 = \{ (9, 2, 0.5) \}$ . This implies that the number of components of  $\bar{v}_4 \cap \bar{v}_2$  equals one. So we add one edge between  $v_4$  and  $v_2$  in  $H_1 \cup H_2$ .

Analogously,  $\bar{v}_4 \cap \bar{v}_5 = \{ (3, 5, 0.5), (6, 2, 0.5) \}$ , which implies that the number of components of  $\bar{v}_4 \cap \bar{v}_5$  equals 2. So we add two edges between  $v_4$  and  $v_5$  in  $H_1 \cup H_2$ .

Let  $E = \{ e_1, e_2, e_3, e_4 \}$  be the set of added edges, and  $V_1 = \{ v_3, v_4 \}$  and  $V_2 = \{ v_2, v_5 \}$  the sets of vertices.

Step 3.3. The restriction of  $\overline{H}_1$  on  $\overline{V}_1$  is  $\overline{U}_1 = \overline{H}_1$ , and that of  $\overline{H}_2$  on  $\overline{V}_2$  is  $\overline{U}_2 = \{ \overline{v}_2, \overline{v}_5 \}$ .

By Figure 8 or a simple calculation, we have  $\bar{v}_3 \cap \bar{v}_1 = \{ (12, 5, 0.5) \}$ . It follows that the number of components of  $\bar{v}_3 \cap \bar{v}_1$  equals one, and we add one edge between  $v_3$  and  $v_1$  in  $H_1 \cup H_2$ .

Analogously,  $(\bar{v}_2 \cup \bar{v}_5) \cap (\bar{v}_3 \cup \bar{v}_4)$  does not contain a 1-path of 2-cells  $(p_0, p_1, \dots, p_n)$  such that  $p_0$  is contained in  $\bar{v}_2 \cap \bar{v}_3$  and  $p_n$  is contained in  $\bar{v}_5 \cap \bar{v}_4$ . Thus, we do not have to delete edge  $v_3v_4$  in  $H_1 \cup H_2$ .

Step 3.4.  $H_1 \cup H_2$  is shown on the top of Figure 8 [case (1)].

Step 4. By Figure 8, the number of edges of the homotopic graph  $H_1 \cup H_2$  equals ten and that of the vertices equals six; consequently, the number of tunnels of  $S_1$  equals 10 - 6 + 1 = 5.

**Example 2.** Let  $S_2 = L_1 \cup L_2 \cup g_4 = S_1 \cup g_4$ . Step 3.2 is exactly the same as in Example 1.

Step 3.3, by examining Figure 8 or simple calculation, we have  $(\bar{v}_2 \cup \bar{v}_5) \cap (\bar{v}_3 \cup \bar{v}_4)$  contains a 1-path of 2-cells ((9, 2, 0.5), (8, 2, 0.5), (7, 2, 0.5), (6, 2, 0.5)) such that (9, 2, 0.5) is contained in  $\bar{v}_3 \cap \bar{v}_2$  ((9, 2, 0.5) = (9, 2, 1)  $\cap$  (9, 2, 0)) and (6, 2, 0.5) is contained in  $\bar{v}_4 \cap \bar{v}_5$  ((6, 2, 0.5) = (6, 2, 1)  $\cap$  (6, 2, 0)). Thus, we have to delete edge  $v_3v_4$  in  $H_1 \cup H_2$ .

Step 3.4.  $H_1 \cup H_2$  is shown in the middle of Figure 8 [case (2)]. Step 4. The number of tunnels of  $S_2$  equals 10 - 6 + 1 = 5.

**Example 3.** Now consider  $S_3 = L_1 \cup L_2 \cup g_5 = S_1 \cup g_5$ . Step 3.2 is exactly the same as in Example 1.

Step 3.3. By examining Figure 8, we can see that  $(\bar{v}_2 \cup \bar{v}_5) \cap (\bar{v}_3 \cup \bar{v}_4)$  does not contain any 1-path of 2-cells  $(p_0, p_1, \dots, p_n)$  such that  $p_0$  is contained in  $\bar{v}_3$  $\cap \bar{v}_2$  and  $p_n$  is contained in  $\bar{v}_4 \cap \bar{v}_5$ . Thus, we do not have to delete edge  $v_3v_4$ in  $H_1 \cup H_2$ .

Step 3.4.  $H_1 \cup H_2$  is shown at the bottom of Figure 8 [case (3)]. Step 4. The number of tunnels of  $S_3$  equals 11 - 6 + 1 = 6.

### 6 Computational Complexity

#### 6.1 Single- Layer Subprocess

Let l be the number of cubes in the considered layer L, and r or c the number of rows or columns in L, respectively.

Step 1 can be computed in time  $\mathcal{O}(l)$ , because  $|S_I| \leq l$ . Each run can have at most  $\lfloor r/2 \rfloor$  left (right) neighbors, so Step 2 can be computed in  $\mathcal{O}(lr)$ . Since each run-path has at most c runs and  $|S_P| \leq |S_I| \leq l$ , it follows that Step 3 can be done in  $\mathcal{O}(lc)$ . Analogous to Step 1, Steps 4, 5, 6 and 7 can be completed in  $\mathcal{O}(l)$ . In Step 8.1, the combination number of pairs  $(h_1, h_2)$  equals because  $|S_H \cup S_{PH}| \leq |S_I| \leq l$ . Furthermore, since each  $h_i$  has at most r neighbors, it follows that Step 8.1 can be computed in  $\mathcal{O}(l^2r)$ . In Step 8.2, there are at most lr ways to take a neighbor of a run in  $S_H$ , and there are also at most lr ways to take a run-path in  $S_{RP}$ . Thus, this step can be finished in  $\mathcal{O}((lr)^2)$ . For Step 9, note that  $|S_H \cup S_{PH}| \leq |S_I| \leq l$ ,  $|C| \leq r$ ,  $|C_N| \leq r$ , and  $|C_P| \leq rc$ . That means that Step 9 has time complexity  $\mathcal{O}(lrc)$ . Finally, it is clear that Step 10 has complexity  $\mathcal{O}(l^2)$ , and this immediately also follows for Step 11. In summary, a run of the single-layer subprocess has time complexity  $\mathcal{O}((lr)^2)$ .

# 6.2 Main Algorithm

Let S be our input, a 2-region of cubes, and  $i_{max}$  the number of layers in S. Let  $l_i$  be the number of cubes in the *i*th layer of S, denoted by  $L_i$ . Let  $r_i$  or  $c_i$  be the numbers of rows or columns in layer  $L_i$ , respectively.

Obviously, Step 1 can be done in  $\mathcal{O}(i_{max})$ . Step 2 is discussed in Section 6.1. For Step 3.1, the labeling algorithm (as, for example, in [12]) has complexity  $\mathcal{O}(r_ic_i)$ . Note that  $|V(\bar{H}_i)| \leq l_i$ , for i = 1, 2. It follows that the combination number of pairs  $(\bar{v}_1, \bar{v}_2)$  equals  $l_1l_2$ ; thus Step 3.1 has time complexity  $\mathcal{O}(l_i^2r_ic_i)$ . For Step 3.2, note that  $|V_i| \leq l_i$ , consequently  $|E| \leq l_1l_2$ . Therefore, Step 3.2 can be computed in  $\mathcal{O}(l_i^2)$ . In Step 3.3, the combination number of pairs  $(\bar{u}_i, \bar{w}_i)$ equals

$$\binom{l_i}{2}$$

and the number of 1-paths  $(p_0, p_1, \dots, p_n)$  of 2-cells, such that  $p_0$  is contained in  $\bar{u}_1 \cap \bar{u}_2$  and  $p_n$  is contained in  $\bar{w}_1 \cap \bar{w}_2$ , is at most  $\lfloor r_i/2 \rfloor$ , where i = 1, 2. [The labeling algorithm has complexity  $\mathcal{O}(r_i c_i)$ .] Thus, this step can be completed in  $\mathcal{O}(l_i^2 r_i^2 c_i)$ . Step 4, finally, recalls the subprocess for all  $i_{max}$  layers, and delivers the final result. Let

$$l = \max_{1 \le i \le i_{max}} l_i, \ r = \max_{1 \le i \le i_{max}} r_i, \ \text{and} \ c = \max_{1 \le i \le i_{max}} c_i$$

The main algorithm has time complexity  $\mathcal{O}(i_{max}l^2r^2c)$ .

# 7 Concluding Remarks

[23] computes the number of tunnels in a  $3 \times 3 \times 3$  neighborhood of any cube of a given 2-region. It does not deal with the problem of computing the number of tunnels for the whole 2-region.

[2, 3] discuss the computational complexity of homology (as being 'large'). Both [6] and [10] have developed software to compute homology. However, they are (so far) unable to inform about the computational complexity of their algorithms, and so are [13, 21]. The latter two compute homology for any finite k-dimensional cubical set (no restrictions on dimensions or shapes). [6] suggests a search for improved algorithms and for new approaches to the task of homology computation while [13] leaves the problem of improving the efficiency of their algorithms to the future. [10] states that time complexity of the proposed algorithm may grow 'horrifically'.

We have presented our algorithm and analyzed its computational complexity, which is relatively low compared to other options. We also believe that the graphtheoretical nature of our algorithm is of general interest. (Obviously, alternative techniques could also be considered such as run graphs [18] or Morse theory and Reeb graphs [7].)

Finally, note that connectedness is also defined for cubical sets of any dimension, and digitization schemes for arbitrary sets in nD space (see [12]). It is our conjecture that the presented graph-theoretical approach may be generalized to compute homology for digitized sets in arbitrary dimensions.

### References

- E. Andres, R. Acharya, and C. Sibata. Discrete analytical hyperplanes. *Graphical Models Image Processing*, 59:302–309, 1997.
- S. Basu. Computing the first few Betti numbers of semi-algebraic sets in single exponential time. Preprint submitted to Elsevier Science (30 October 2005), see www.math.gatech.edu/~saugata.
- S. Basu, R. Pollack, and M. Roy. Computing the first Betti number and describing the connected components of semi-algebraic sets. In Proc. STOC, pages 304–312, 2005 (see also new and improved version on www.math.gatech.edu/~saugata).
- 4. C. Bonnington and C. Little. *The Foundations of Topological Graph Theory*. Springer, New York, 1995.
- V. E. Brimkov, A. Maimone, G. Nordo, R. P. Barneva, and R. Klette. The number of paps in binary pictures. In Proc. *ISVC*, pages 35–42, 2005.
- CHomP (Atlanta) and CAPD (Kraków). Homology algorithms and software. www.math.gatech.edu/ chomp/homology/.
- K. Cole-McLaughlin, H. Edelsbrunner, J. Harer, V. Natarajan, and V. Pascucci. Loops in Reeb graphs of 2-manifolds. *Discrete Comput. Geom.*, 32:231–244, 2004.
- C. J. A. Delfinado and H. Edelsbrunner. An incremental algorithm for Betti numbers of simplicial complexes on the 3-sphere. *Comput. Aided Geom. Design*, 12:771–784, 1995.
- M. Desbrun, E. Kanso, and Y. Tong. Discrete differential forms for computational modeling. In ACM SIGGRAPH 2005 Course Notes on Discrete Differential Geometry, Chapter 7, 2005.
- 10. V. De Silva. Plex A Matlab library for studying simplicial homology. math.stanford.edu/comptop/programs/plex/plexintro.pdf.
- S. Fourey and R. Malgouyres. A concise characterization of 3D simple points. Discrete Applied Mathematics, 125:59–80, 2002.
- 12. R. Klette and A. Rosenfeld. *Digital Geometry: Geometric Methods for Digital Picture Analysis.* Morgan Kaufmann, San Francisco, 2004.
- T. Kaczynski, K. Mischaikow and M. Mrozek. Computational Homology. Applied Mathematical Science, Volume 157, Springer-Verlag, 2004.
- T. Y. Kong and A. Rosenfeld. Digital topology: introduction and survey. Computer Vision Graphics Image Processing, 48:357–393, 1989.
- 15. F. Li and R. Klette. Calculation of the number of tunnels. Institute for Mathematics and its Applications, Minneapolis, Technical Report, April 2006.

- 16. G. Lohmann. Volumetric Image Analysis. Wiley & Teubner, Chichester, 1998.
- C.-M. Ma and S.-Y. Wan. Parallel thinning algorithms on 3D (18, 6) binary images. Computer Vision Image Understanding, 80: 364–378, 2000.
- G. Monagan and M. Röösli. Appropriate base representation using a run graph. In Proc. Int. Conf. Document Analysis Recognition, pages 623626, 1993.
- A. Nakamura, K. Morita, and K. Imai. B-problem. CITR-TR-180, The University of Auckland, Computer Science Department, 2006.
- 20. J. Ohser, W. Nagel, and K. Schladitz. The Euler number of discretized sets on the choice of adjacency in homogeneous lattices. In *Morphology of Condensed Matter*. *Physics and Geometry of Spatially Complex Systems*, LNP 600, pages 275–298, 2002.
- S. Peltier, S. Alayrangues, L. Fuchs, and J. Lachaud. Computation of homology groups and generators. In Proc. DGCI, LNCS 3429, pages 195-205, 2005.
- 22. H. Poincaré. Analysis situs. J. Ecole Polytechnique (2), 1:1-121, 1895.
- P. K. Saha and B. B. Chaudhuri. 3D digital topology under binary transformation with applications. *Computer Vision Image Understanding*, 63:418–429, 1996.
- P. K. Saha, B. R. Gomberg, and F. W. Wehrli. Three-dimensional digital topological characterization of cancellous bone architecture. *Int. J. Imaging Systems Technology*, 11:81–90, 2000.
- S. N. Srihari. Representation of three-dimensional digital images. ACM Comput. Surveys, 13:399–424, 1981.
- A. Tscheschel and D. Stoyan. On the estimation variance for the specific Euler-Poincaré characteristic of random networks. J. Microsc., 211:80-88, 2003.
- G. Watanabe, K. Sato, K. Yasuoka, and T. Ebisuzaki. Microscopic study of slablike and rodlike nuclei: Quantum molecular dynamics approach. *Physical Review*, C 66, 012801, 5 pages, 2002.
- A. T. White. On the genus of the composition of two graphs. Pacific J. Math., 41:275-279, 1972.