Distinguishing the 3 primitive 3D-topological configurations: simplex, hole, tunnel *

Abstract *With most of the work focussing on 2D representations, topology preserving hierarchies have received a lot of attention. Concepts for extending such representations to handle any dimension have also been the subject of active research in the recent years, but very little work has been done to collapse a huge amount of volumetric data into it's minimal topologically equivalent data structure. This paper presents 3D combinatorial maps and the primitive operations needed to simplify such a representation. Minimal configurations of the three primitive topological configurations, simplex, hole, and tunnel, are studied. First experimental results and possible applications show the potential of the approach.*

1 Introduction

Handling "structured geometric objects" is important for many applications related to geometric modeling, computational geometry, image analysis, etc.; one has often to distinguish between different parts of an object, according to properties which are relevant for the application (e.g. mechanical, photometric, geometric properties).

For instance for geological modeling, the sub-ground is made of different layers, maybe split by faults, so layers are sets of (maybe not connected) geological blocks.

For e.g. in image analysis, a region is a (structured) set of pixels or voxels, or more generally an abstract cellular complex consisting of dimensions $0, 1, 2, 3$ ... (i.e. 0-cells are vertices, 1-cells are edges, 2-cells are faces, 3-cells are volumes, ...) and a bounding relation [17].

The structure, or the topology, of the object is related to the decomposition of the object into sub-objects, and to the relations between these sub-objects.

Basically, topological information is related to the cells and their adjacency and bounding relations. Other information (embedding information) are associated to these sub-objects, and describe for instance their shapes (e.g. a point, resp. a curve, a part of a surface, is associated with each vertex, resp. each edge, each face), their textures or colors, or other information depending on the application.

Many topological models have been conceived for representing the topology of subdivided objects, since different types of subdivisions have to be handled: general complexes [8, 9] or particular manifolds [1, 2], subdivided into any cells [14, 12] or into regular ones (e.g. simplices, cubes, etc.) [13, 20]. Few models are defined for any dimensions [3, 21, 5, 19]. Some of them are (extensions of) incidence graphs or adjacency graphs. Their principle is often simple, but

- they cannot deal with any subdivision without loss of information, since it is not possible to describe the relations between two cells precisely if they are incident in several locations;
- operations for handling such graphs are often complex, since they have to handle simultaneously different cells of different dimensions.

Other structures are "ordered" [5, 19], and they do not have the drawbacks of incidence or adjacency graphs. A subdivided object can be described at different levels, so several works deal with hierarchical topological models and topological pyramids [11, 3, 18]. For geometric modeling, levels are often not numerous. For image analysis, more levels are needed since the goal is to rise up information which is not known a priori.

[7, 15] show that 2D combinatorial maps are suitable topological structures to be used in 2D segmentation. Many domains need to work in 3D imagery (e.g. medicine, geology), so the theoretical framework of 2D combinatorial maps has been extended to 3D [10, 4]. In order to use 3D topological structures for 3D image segmentation one has to define basic operations. In this paper only two basic operations are introduced: the contraction and the removal operation.

Attempts have been made to reduce such representations [10] to a certain extent, without guaranteeing the minimal representation. We extend this work in order to find minimal representations of the topological configurations of the initial data. And distinguish between them using the remaining pseudo-elements.

A short introduction of the 2D and 3D combinatorial map is given in Section 2. In Section 3 the two operations, namely contraction and removal, are properly applied to three objects. We show examples of the three basic structures in 3D: a simply connected volume, a volume with a hole (volume enclosing other volume), and a volume with

* This paper was supported by the Austrian Science Fund under grant FSP-S9103-N04.

a tunnel (donut) and their minimal configurations that preserve the topology.

2 Combinatorial Maps

Combinatorial maps and *generalized maps* define a general framework which allows us to encode any subdivision of nD topological spaces orientable or non-orientable with or without boundaries. They encode all the incidence relations and consist of abstract elements, called darts $\mathcal{D}$ and a set of permutations β_i. i-cells are implicitly encoded by subsets of $\mathcal{D}$ which can be obtained using the β_i permutations. (When encoding the same configuration, differences between the two mentioned map types are limited to the number of darts, number of permutations, and their meaning).

In the case of combinatorial maps, for each dimension, there is more then one way of attributing the permutations, but the number of permutations used for a certain dimension and how many of them are involutions is fixed i.e. for an nD combinatorial map there is 1 permutation and $n-1$ involutions (an involution is a permutation whose orbits are of size 1 or 2).

2D and 3D combinatorial maps are given in more detail in the following sections.

2.1 2D Combinatorial Maps

$2D$ Combinatorial maps may be understood as a particular encoding of a planar graph, where each edge is split into two half-edges, the so called darts. A $2D$ combinatorial map is formally defined by the triplet $G = (\mathcal{D}, \sigma, \alpha)$ [6] where $\mathcal{D}$ represents the set of darts, σ is a permutation on $\mathcal{D}$ encountered when turning clockwise around each vertex (the cycles of σ encode the vertices), and α is an involution on $\mathcal{D}$ which maps each of the two darts of one edge to the other one (the cycles of α encode the edges). The cycles of the permutation φ, defined as $\varphi = \sigma \circ \alpha$, encode the faces of the combinatorial map. (see Fig. 1d) [10] uses β_1 to refer to φ and β_2 to refer to α and represents the $2D$ combinatorial map as $G = (\mathcal{D}, \beta_1, \beta_2)$ (see Fig. 1b).

2.2 3D Combinatorial Maps

A $3D$ combinatorial map is formally defined by $G = (\mathcal{D}, permutation, involution_1, involution_2)$, with the following two notations (and meanings) for the permutations studied until now: $G = (\mathcal{D}, \beta_1, \beta_2, \beta_3)$ [4] and $G = (\mathcal{D}, \gamma, \sigma, \alpha)$ [4]. Further on, we will present the first one.

Like in the similar $2D$ combinatorial map notation, the permutation β_1 connects darts belonging to the same face and the same volume, preserving their ordering on the boundary of the face, and the involution β_2 connects 2 darts, part of the same edge and the same volume. The additional involution, β_3, links 2 darts that belong to the same face and same edge (and the 2 volumes separated by the respective face). β_3 can be regarded like a glue, which brings together neighboring volumes defined be the $2D$ manifolds encoded by β_1 and β_2 (see Fig. 2).

For a certain dart d, the set of darts implicitly representing the i-cell containing the dart d is obtained by applying 2 of the 3 permutations β_i any number of times and in any combination to the dart d. (i is defined by the 2 permutations applied) [4].

2.2.1 Operations on the 3D Combinatorial Maps

We apply two operations to an i-cell: removal (removes the i-cell and merges the 2 (i+1)-cells that it was separating) and contraction (contracts the i-cell to a (i-1)-cell by merging it's 2 neighboring (i-1)-cells). For our experiments we used the following 4 operations: *edge contraction*, *face contraction*, *volume contraction* and *face removal*. (See Table 1)

Our maps encode volumes from the input data as vertices and thus *edge contraction* is the equivalent to merging two such neighboring volumes. The other 3 operations are applied to simplify/collapse the resulting representation, while preserving the correct topological configuration. The last one (*face removal*) is needed to deal with the special case of "face self loop", which is a face that encloses a volume alone, and which is bounded by one edge. (Such a self loop can be the result of the contraction operations described).

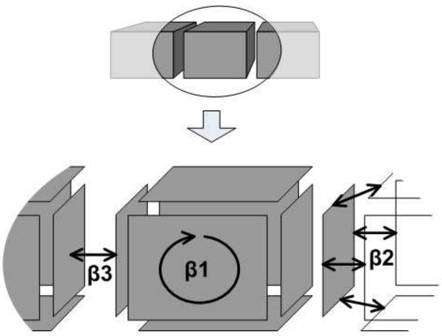


Figure 2: $3D$ Combinatorial map permutations

2.2.2 Pseudo elements

To keep the topological encoding consistent, the simplification process keeps i-cells which help encoding inside-like relations. In $2D$ this means keeping self-loops and parallel edges which surround at least one vertex, in $3D$ this concept is translated as parallel faces and "face self loops" (faces bounded by a single edge) which enclose a vertex. As shown in the following sections, these pseudo elements let us discriminate between different topological configurations.

2.2.3 Multiple minimal encodings

Every i-cell needs to be bounded by at least one (i-1)-cell i.e. a volume is bounded by at least one face, a face by at least one edge, and an edge by at least 1 vertex. This leads to multiple encodings for the same topological configuration which cannot be reduced/collapsed anymore. For example a single volume, can be represented as a volume bounded by 2 faces bounded by the same edge (self loop) and 1 vertex (a globe obtained from gluing together 2 halves around the self loop which is the equator) (Fig. 3a), or as a volume bounded by 1 face, bounded by 1 edge connecting 2 vertices (a soap bubble hanging in the middle of the straw)(Fig. 3b). So, depending on the operations applied and their order, starting from the same initial configuration, we can obtain different output configurations that are topologically equivalent.

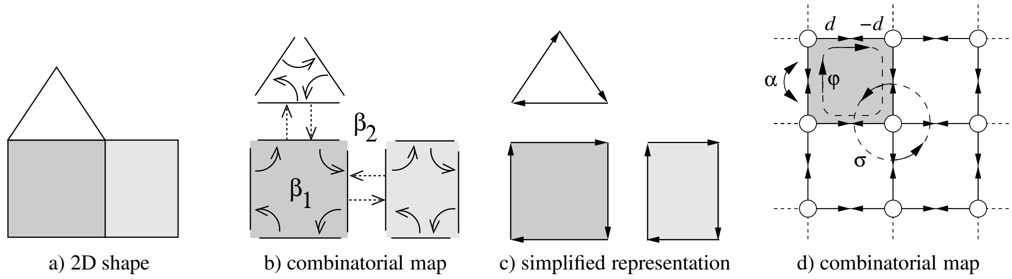


Figure 1: 2D Combinatorial maps using different notations.

Operation	Preconditions	Result
edge contraction	edge connects 2 different vertices, no volumes or faces will be removed	the 2 vertices are merged, contracted edge is removed
face contraction	face is bounded by 2 different edges no volumes or vertices will be removed	the 2 bounding edges are merged contracted face is removed
volume contraction	volume is bounded by 2 different faces no vertices or edges will be removed	the 2 bounding faces are merged the volume is removed
face removal	face is incident to 2 different volumes no vertices or edges are removed	the 2 incident volumes are merged the face is removed

Table 1: Operations applied to the 3D Combinatorial map

3 Connected component analysis

As mentioned in Section 2.2.1, in our setup voxels from the input data are represented as vertices and adjacency relations between 2 voxels are represented by connecting their 2 associated vertices by an edge. An additional vertex is used to represent the background volume. For the sake of clarity, this vertex is not drawn in the initial configuration images of our experiments.

The algorithm for identifying the connected components is as follows (each operations is applied only if the preconditions mentioned in Table 1 are satisfied):

1. contract all edges connecting two vertices that belong to the same connected component
2. contract all faces bounded by exactly two edges
3. contract all volumes bounded by exactly two faces
4. remove all "face self loops"

The four steps are repeated until Step 1 does not find any more contractable edges. In each such iteration, Steps 2-4 are repeated until neither of them finds any more candidates for contraction/removal.

3.1 2 x 2 x 2 Cube - 1 connected component

The first experiment is the reduction of a $2 \times 2 \times 2$ **cube** where each voxel has the same label. Fig. 4a shows the initial combinatorial map for this object. (The labels of vertices and edges correspond to the labels used by our library.) The final configuration is shown in Fig. 4b. The map is reduced to 4 darts defining 2 vertices, 2 edges, 1 face and 1 volume.

One vertex represents the background and the other one represents the voxels of the initial cube that has been merged into 1 element. These 2 vertices are connected by 1 face that is bounded by 2 edges.

3.2 3 x 3 x 3 Cube with enclosed object inside - 2 connected components

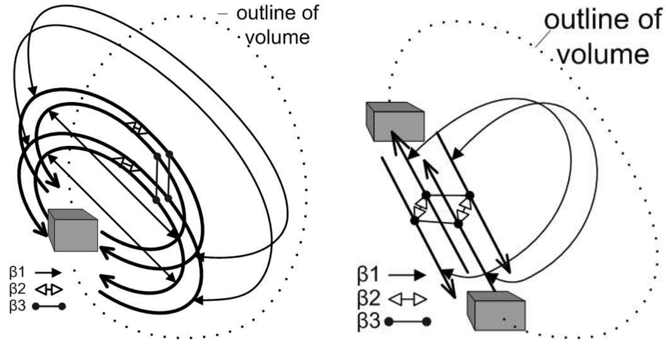
To demonstrate that the topology is preserved during the simplification of the combinatorial map, the second experiment reduces a **cube that completely encloses another object**. Fig. 4c shows the initial combinatorial map for this configuration. The two objects are reduced to a combinatorial map consisting of 16 darts defining 4 vertices, 5 edges, 3 faces and two volumes (see Fig. 4d).

The outer cube enclosing the inner object is merged into 2 vertices connected by a single edge. These 2 vertices (vertex 17 and 26) connect to the background (vertex 28) and the inner object (vertex 14). In addition the edge between these 2 vertices defines a face that completely encloses the inner object (vertex 14) representing the inclusion relation of this object being inside the outer cube.

3.3 3 x 3 x 2 Cuboid with object tunnel inside - 1 connected component

In $3D$ there are basically 2 types of *inside* configurations. The 3^{rd} experiment shows the **reduction of a torus** which is surrounding (but not completely enclosing) another object. Fig. 4e shows the initial combinatorial map for this experiment. The 2 objects are reduced to a combinatorial map consisting of 24 darts defining 3 vertices, 5 edges, 4 faces and 2 volumes (see Fig. 4f).

The torus is merged into 1 vertex (vertex 17) connected to the background (vertex 19) and the inner tunnel (vertex 14). The tunnel (vertex 14) is connected on both sides with the background (edges -755 and -155). The fact that the torus surrounds the tunnel is represented by the self loop (edge -679) and the cone like face/surface (bounded by the self-loop edge -679 and the edge -826; visualized by the dotted lines).



a) 2 faces, 1 vertex b) 1 face, 2 vertices

Figure 3: Multiple minimal encodings for one volume

Configuration	Darts	Vertices	Edges	Faces	Volumes
2x2x2 cube (initial)	120	9	20	18	7
2x2x2 cube (final)	4	2	2	1	1
2x2x2 cube (final - pseudo elements)			**0**	**0**	**0**
3x3x3 with object inside (initial)	576	28	80	84	32
3x3x3 with object inside (final)	16	4	5	3	2
3x3x3 with object inside (final - pseudo elements)			**0**	**1**	**1**
3x3x2 with tunnel inside (initial)	352	19	51	52	20
3x3x2 with tunnel inside (final)	24	3	5	4	2
3x3x2 with tunnel inside (final - pseudo elements)			**1**	**1**	**0**

Table 2: Number of cells in each experimented configuration

3.4 Discriminating between the 3 configurations

As can be seen from the experiment results, discriminating between the first configuration and the other two is very easy, and can be done just by looking at the labels of the obtained vertices. The second and third configurations are more complex, because of the containment relation and cannot be discriminated based only on the vertices. Here, edges and faces have to be taken into considerations. An object having an edge self loop (or edge cycle) surrounds another object (Fig. 4f), and an object having a face self-loop (or face cycle) encloses another one (Fig. 4d). Note that in experiment 3, the adjacency of the tunnel and the background is also shown by the 2 edges connecting it to the background.

4 Outlook

Connected component analysis is certainly one of the first experiments to do when testing out a new representation that should preserve topology, but the possibilities do not stop here. Next steps will include the extension to $3D$ of the Minimum Spanning Tree pyramid concept [15] used for segmentation of $2D$ images, and using it to segment volumetric data and videos ($2D$+time). Further on, having this implementation, we can pursue research in describing videos using actions, events, and relations, following the concept presented in [16].

5 Conclusions

The paper presents the basic operations that can collapse a high resolution voxel complex into its topologically equivalent smallest counterpart. We demonstrated the correctness of the underlying software library by the three basic configurations in 3D: a simply connected volume, a volume with a hole and a volume with a tunnel. The resulting structures contain pseudo elements characterizing the respective topology.

References

[1] S. Ansaldi, L. de Floriani, and B. Falcidieno. Geometric Modeling of Solid Objects by Using a Face Adjacency Graph Representation. *Computer Graphics*, 19(3):131–139, 1985.

[2] B. Baumgart. A Polyhedron Representation for Computer Vision. In *AFIPS National Computer Conferenc Proc.*, volume 44, pages 589–596, Anaheim, May 1975.

[3] Y. Bertrand, G. Damiand, and C. Fiorio. Topological Encoding of 3D Segmented Images. In G. Borgefors, I. Nyström, and G. S. di Baja, editors, *International Conference on Discrete Geometry for Computer Imagery*, volume 1953 of *Lecture Notes in Computer Science*, pages 311–324. Springer-Verlag, Germany, 2000.

[4] A. Braquelaire, G. Damiand, J-P. Domenger, and F. Vidil. Comparison and convergence of two topological models for 3d image segmentation. In *Workshop on Graph-Based Representations in Pattern Recognition*, number 2726 in Lecture Notes in Computer Science, pages 59–70, York, England, June 2003.

[5] E. Brisson. Representing Geometric Structures in D Dimensions: Topology and Order. *Discrete and Computational Geometry*, 9:387–426, 1993.

[6] Luc Brun and Walter G. Kropatsch. Dual Contraction

of Combinatorial Maps. Technical Report PRIP-TR-54, Institute f. Computer Aided Automation 183/2, Pattern Recognition and Image Processing Group, TU Wien, Austria, 1999. Also available through http://www.prip.tuwien.ac.at/ftp/pub/publications/trs/tr54.ps.gz.

[7] Luc Brun and Walter G. Kropatsch. Irregular Pyramids with Combinatorial Maps. In Francesc J. Ferri, José M. Iñesta, Adnan Amin, and Pavel Pudil, editors, *Advances in Pattern Recognition, Joint IAPR International Workshops on SSPR'2000 and SPR'2000*, volume 1876 of *Lecture Notes in Computer Science*, pages 256–265, Alicante, Spain, August 2000. Springer, Berlin Heidelberg, New York.

[8] P. Cavalcanti, P. Carvalho, and L. Martha. Non-manifold Modeling: An Approach Based on Spatial Subdivision. *Computer-Aided Design*, 29(3):209–220, 1997.

[9] G.A. Crocker and W.F. Reinke. An Editable Nonmanifold Boundary Representation. *Computer Graphics and Applications*, 11(2):39–51, 1991.

[10] G. Damiand. *Définition et étude d'un modèle topologique minimal de représentation d'images 2d et 3d*. Thèse de doctorat, Université Montpellier II, Décembre 2001.

[11] L. De Floriani, E. Puppo, and P. Magillo. A Formal Approach to Multiresolution Hypersurface Modeling. In R. Straber, W. andKein and R. Rau, editors, *Geometric Modeling: Theory and Practice*, pages 302–323. Springer-Verlag, 1997.

[12] D. Dobkin and M. Laszlo. Primitives for the manipulation of three-dimensional subdivisions. *Algorithmica*, 4(1):3–32, 1989.

[13] V. Ferruci and A. Paoluzzi. Extrusion and Boundary Evaluation for Multidimensional Polyhedra. *Computer-Aided Design*, 23(1):40–50, 1991.

[14] L. Guibas and J. Stolfi. Primitives for the Manipulation of General Subdivisions and the Computation of Voronoi Diagrams. *ACM Trans. on Graphics*, 4(2):74–123, 1985.

[15] Yll Haxhimusa, Adrian Ion, Walter G. Kropatsch, and Luc Brun. Hierarchical Image Partitioning using Combinatorial Maps. In D. Chetverikov, L. Czuni, and M. Vincze, editors, *Joint Hungarian-Austian Conference on Proceedings on Image Processing and Pattern Recognition, HACIPPR 2005 - OAGM 2005/KPAF 2005*, pages 179–186, Veszprm, Hungary, 11-13, May 2005. OCG.

[16] Adrian Ion, Yll Haxhimusa, and Walter G. Kropatsch. A Graph-Based Concept for Spatiotemporal Information in Cognitive Vision. In L. Brun and M. Vento, editors, *5th IAPR-TC15 Workshop on Graph-based Representation in Pattern Recognition*, volume 3434 of *Lecture Notes in Computer Science*, pages 223–232, Poitiers, France, April 2005. Springer, Berlin Heidelberg, New York.

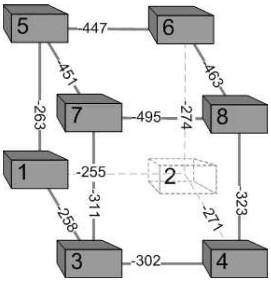
[17] Vladimir A. Kovalevsky. Finite topology as applied to image analysis. *Computer Vision, Graphics, and Image Processing*, 46:141–161, 1989.

[18] W.G. Kropatsch. Building Irregular Pyramids by Dual Graph Contraction. *IEE-Proc. Vision, Image and Signal Processing*, 142(6):366–374, 1995.

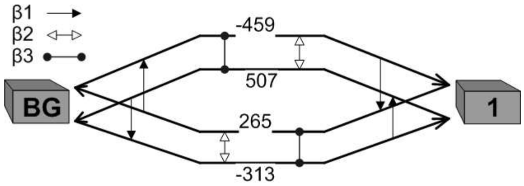
[19] P. Lienhardt. N-dimensional Generalized Combinatorial Maps and Cellular Quasi-manifolds. *Int. J. of Comp. Geom. and Appl.*, 4(3):275–324, 1994.

[20] A. Paoluzzi, F. Bernardini, C. Cattani, and V. Ferrucci. Dimension Independent Modeling with Simplicial Complexes. *ACM Trans. on Graphics*, 12(1):56–102, 1993.

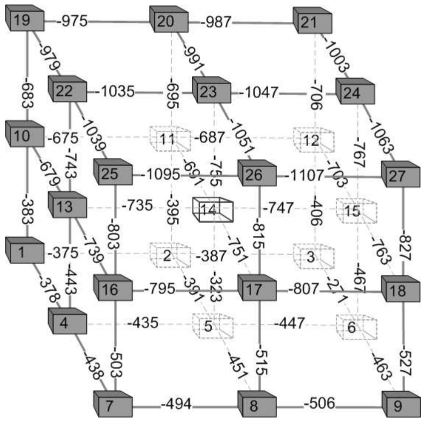
[21] J. Rossignac and M. O'Connor. SGC: a Dimension-independent Model for Pointsets with Internal Structures and Incomplete Boundaries. In M. J. Wozny, J. Turner, and K. Preiss, editors, *In Geometric Modeling for Product Engineering*, pages 145–180. Elsevier Science, 1989.



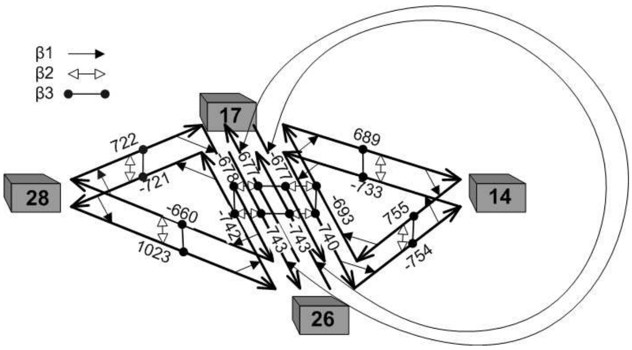
a) 2 × 2 × 2 initial map



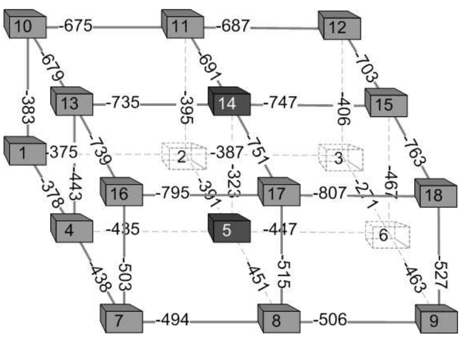
b) 2 × 2 × 2 final map



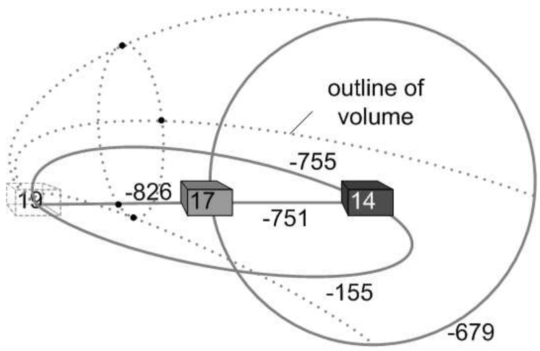
c) 3 × 3 × 3 initial map



d) 3 × 3 × 3 final map



e) 3 × 3 × 2 initial map



f) 3 × 3 × 2 final map

Figure 4: The 3 primitive 3D topological configurations: simplex(a,b), hole(c,d), tunnel(e,f)