

Efficiently Storing Well-Composed Polyhedral Complexes Computed Over 3D Binary Images

Rocio Gonzalez-Diaz¹ · Maria-Jose Jimenez¹  · Belen Medrano¹

Received: 11 July 2016 / Accepted: 1 March 2017 / Published online: 13 March 2017
© Springer Science+Business Media New York 2017

Abstract A 3D binary image I can be naturally represented by a combinatorial-algebraic structure called cubical complex and denoted by $Q(I)$, whose basic building blocks are vertices, edges, square faces and cubes. In Gonzalez-Diaz et al. (Discret Appl Math 183:59–77, 2015), we presented a method to “locally repair” $Q(I)$ to obtain a polyhedral complex $P(I)$ (whose basic building blocks are vertices, edges, specific polygons and polyhedra), homotopy equivalent to $Q(I)$, satisfying that its boundary surface is a 2D manifold. $P(I)$ is called a *well-composed polyhedral complex over the picture I* . Besides, we developed a new codification system for $P(I)$, encoding geometric information of the cells of $P(I)$ under the form of a 3D grayscale image, and the boundary face relations of the cells of $P(I)$ under the form of a set of structuring elements. In this paper, we build upon (Gonzalez-Diaz et al. 2015) and prove that, to retrieve topological and geometric information of $P(I)$, it is enough to store just one 3D point per polyhedron and hence neither grayscale image nor set of structuring elements are needed. From this “minimal” codification of $P(I)$, we finally present a method to compute the 2-cells in the boundary surface of $P(I)$.

Keywords 3D binary image · Well composedness · Cubical complex · Well-composed polyhedral complex

1 Introduction

Consider \mathbb{Z}^3 as the set of points with integer coordinates in 3D space \mathbb{R}^3 . A *3D binary image* (also called *picture*) is a set $I = (\mathbb{Z}^3, 26, 6, F_I)$ (or $I = (\mathbb{Z}^3, F_I)$, for short), where $F_I \subset \mathbb{Z}^3$ is the *foreground*, $F_I^c = \mathbb{Z}^3 \setminus F_I$ the *background*, and $(26, 6)$ is the adjacency relation for the foreground and background, respectively. A 3D binary image I is *well composed* [14] if the boundary surface of its continuous analog is a 2D manifold. A continuous analog $C(I)$ of I (see, for example, [13, 14]) is constructed by interpreting each point in F_I as a unit cube in \mathbb{R}^3 . If two points in I are adjacent, then the corresponding cubes in $C(I)$ are connected. The boundary surface of $C(I)$ is the set of faces of the cubes that separate $C(I)$ from its complement.

Well-composed images enjoy important topological and geometric properties in such a way that several algorithms used in computer vision, computer graphics and image processing are simpler. For example, if the image is well composed, thinning algorithms do not suffer from the irreducible thickness problem [16] and the digitization process comes to be topology preserving [17]. The property of well composedness is also used as an important tool for proving theoretical results in digital topology [20]. Unfortunately, natural and synthetic images are not a priori well composed. There are several “repairing” methods for turning them into well-composed images, such as, for example, [15, 23], but in those papers, the topology of the object is not preserved.

A different but related viewpoint has been adopted in several papers on which an isosurface is constructed to represent the boundary of a digitized object, like in [13, 24]. In

Partially supported by MTM2015-67072-P (MINECO/FEDER, UE).

✉ Maria-Jose Jimenez
majiro@us.es
<http://personal.us.es/majiro>

Rocio Gonzalez-Diaz
rogodi@us.es
<http://personal.us.es/rogodi>

Belen Medrano
belenmg@us.es
<http://personal.us.es/belenmg>

¹ Department of Applied Mathematics I, School of Computer Engineering, University of Seville, Seville, Spain

this line, marching cubes [18] is a widely used method to create triangle boundary surfaces from 3D binary images. That algorithm computes the boundary surface adding triangles by looking at the 8-cube configurations. However, some holes/cracks can appear when creating the boundary surface due to ambiguities in some configurations. To obviate these ambiguities, a novel method called connectivity-consistent marching cubes was developed in [8]. The resulting surface correctly reflexes the topology of the original image only if the adjacency relation is properly chosen. On the other hand, the extraction of isosurfaces from well-composed images using the marching cubes algorithm or some octree-based algorithms can be simplified [22]. In [11], the authors gave a polyhedral boundary surface whose vertices are border points of the given 3D binary image, in such a way that adjacent vertices are m -adjacent points, for $m = 6, 18, 26$. Their method does not guarantee that the obtained boundary surface is a 2D manifold.

Some other works in the literature extend the notion of digital well composedness to nD gray-valued images (see, for example, [2, 19]). In [3], a method is proposed that produces, without any interpolation, well-composed gray-valued images in nD.

The notion of well composedness was also extended in [4, 5, 25] to 3D complete polyhedral complexes K embedded in \mathbb{R}^3 . This way, K is *well composed* if the boundary surface of its geometric realization is a 2D manifold.

In [5], we presented a method to “locally repair” the cubical complex $Q(I)$ associated with a voxel-based representation of a given 3D binary image I , to obtain a well-composed polyhedral complex $P(I)$ homotopy equivalent to $Q(I)$. A codification system called *ExtendedCubeMap (ECM) representation* of $P(I)$, was developed in that paper. In the ECM representation, each cell of $P(I)$ is “encoded” by a point in \mathbb{Z}^3 which has been assigned a value (the dimension of the cell) and with a specific neighboring configuration of points determined by the geometry and the topology of the cell. In other words, the (geometric) information of the cells of $P(I)$ is codified under the form of a 3D grayscale image g_P and boundary face relations between the cells of $P(I)$ are codified under the form of a set B_P of structuring elements (or templates) that can be stored as indexes in a look-up table.

In [7], we proved that geometric information of the cells of $P(I)$ can be encoded just by using a 3D binary image (no grayscale values are needed). This way, the type of 3D coordinates codifying each cell, together with information encoded at its neighbor points, provide both geometry and the boundary face relations of each cell without the need of using a set of structuring elements.

In this paper, we go further and prove that it is enough to store just one 3D point per polyhedron to recover geometric and topological information of $P(I)$. This way, the resulting 3D binary image $M = (\mathbb{Z}^3, F_M)$ encoding $P(I)$ could be

compressed in order to be more efficiently stored. We also present an algorithm to compute the 2-cells in the boundary surface of $P(I)$.

A naive demo implemented in MATLAB for computing the minimal encoding M of $P(I)$ and the ECM representation and boundary surface of $P(I)$ directly from M can be downloaded from [6]. Observe that storing $P(I)$ as an image facilitates its computational treatment. For example, the naive way to store any polyhedral complex is by giving a list of its cells and boundary matrices in which 1 at position (i, j) means that cell j lies in the boundary of cell i . This way, to compute the boundary of a given cell, we have to deal with huge matrices what is computationally expensive. Using images, the boundary of a given cell can be obtained in constant time as it is shown in Procedure 2.

An outline of the paper is as follows: In Sect. 2, we fit to cubical complexes context, where the presented ideas are rather trivial but they are worth it to mention in order to help the understanding of next section. The bulk of the paper lies on Sect. 3. In Sect. 3.1, Procedure 1 is recalled, for computing the ECM representation of $P(I)$. In Sect. 3.2, we present the main results of the paper including Procedure 3, designed to compute the minimal encoding of $P(I)$ in which just one point per polyhedron of $P(I)$ is stored, showing that neither grayscale image, nor set of structuring elements are needed. To recover geometric and topological information of $P(I)$, Procedure 4 is also presented. In Sect. 4, Procedure 5 is developed to detect the 2-cells belonging to the boundary surface of $P(I)$. The paper ends up with a section of conclusions and future work, followed by an “Appendix” with the proofs of the results, which have been moved there for a better understanding of the paper. Figure 1 is a general scheme of the paper showing the relations between the procedures presented. Finally, Table 1 presents a list of notations used in the paper.

2 3D Binary Images and Cubical Complexes

In this section, we explain how to associate a cubical complex $Q(I)$ to a 3D binary image I . Later, we explain how to encode $Q(I)$ by a 3D binary image J . In fact, this section can be considered as an introduction to Sect. 3 which is the heart of the paper.

2.1 3D Binary Images Represented by Cubical Complexes

A **unit** closed cube with square faces parallel to the coordinate planes, centered at a point $p \in \mathbb{Z}^3$ is called a *voxel*. The *0-faces* of a given voxel c are its 8 corners (vertices), its 1-faces are its 12 edges, its 2-faces are its 6 square faces and, finally, its 3-face is the voxel itself. The *continuous analog* of

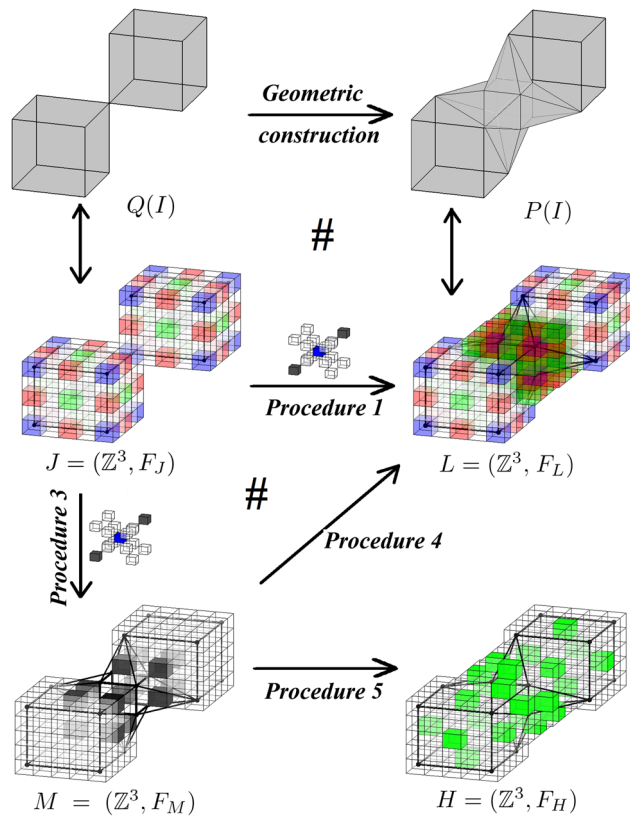


Fig. 1 A global scheme showing the relations between different structures presented in this paper. The input example is a 3D cubical complex $Q(I)$ and consists of two voxels sharing a common vertex. Move right to see the well-composed polyhedral complex $P(I)$ homotopy equivalent to $Q(I)$. Move down to see $Q(I)$ and $P(I)$ encoded as 3D binary images J and L , respectively. Procedure 1 in p. 10 directly transforms J to L . Procedure 3 in p. 13 computes a minimal encoding M of $P(I)$ directly from J . Procedure 4 in p. 13 computes L from M . Finally, Procedure 5 in p. 14 computes the set F_H of 2-cells in the boundary surface of $P(I)$ directly from M

a given 3D binary image $I = (\mathbb{Z}^3, F_I)$, denoted by $C(I)$, is the set of voxels centered at the points of F_I . The *boundary surface* of $C(I)$ is the set of points in \mathbb{R}^3 that are shared by the voxels centered at points of F_I and those centered at points of F_I^c (see [1, 10, 21]). In this paper, **voxels are re-scaled with factor 4** (like in [5, 7]), so that $C(I)$ is composed of size-4 cubes (that will be called, again, voxels) centered at points $4F_I \subset 4\mathbb{Z}^3$. Now, the set of voxels of $C(I)$, together with all their faces, can be associated with a cubical complex denoted by $Q(I)$, whose elements are called *cells* and whose geometric realization is exactly $C(I)$.

Let r_σ denote the barycenter of a cell $\sigma \in Q(I)$. Then, we have that:

- If σ is a voxel (size-4 cube) then $r_\sigma \in \mathcal{E}_3 = \{(4i, 4j, 4k)\}_{i,j,k \in \mathbb{Z}}$;
- If σ is a size-4 square face then $r_\sigma \in \mathcal{E}_2 = \{(4i + 2, 4j, 4k), (4i, 4j + 2, 4k), (4i, 4j, 4k + 2)\}_{i,j,k \in \mathbb{Z}}$;

- If σ is a size-4 edge then $r_\sigma \in \mathcal{E}_1 = \{(4i + 2, 4j + 2, 4k), (4i, 4j + 2, 4k + 2), (4i + 2, 4j, 4k + 2)\}_{i,j,k \in \mathbb{Z}}$;
- If σ is a vertex then $r_\sigma \in \mathcal{E}_0 = \{(4i + 2, 4j + 2, 4k + 2)\}_{i,j,k \in \mathbb{Z}}$.

Finally, since the point r_σ is unequivocally determined by the cell σ , we will say that r_σ *encodes* σ .

As we will see in Sect. 3, the reason of re-scaling with factor 4 is not only to maintain integer coordinates in the points that encode the cells of $Q(I)$, but also to maintain integer coordinates for the points that encode the cells of the well-composed polyhedral complex $P(I)$ over the picture I .

2.2 Cubical Complexes Encoded by 3D Binary Images

In this subsection, we explain how the cubical complex $Q(I)$ described above can be encoded using a 3D binary image that will be denoted by $J = (\mathbb{Z}^3, F_J)$.

First, let us define the following sets of neighbors of a point $p \in \mathbb{Z}^3$:

- $N_6^t(p) = \{p + q, q \in \{(\pm t, 0, 0), (0, \pm t, 0), (0, 0, \pm t)\}\}$,
- $N_{12}^t(p) = \{p + q, q \in \{(0, \pm t, \pm t), (\pm t, 0, \pm t), (\pm t, \pm t, 0)\}\}$,
- $N_8^t(p) = \{p + q, q \in \{(\pm t, \pm t, \pm t)\}\}$,
- $N^{\leq t}(p) = \{p + q, q \in \{(x_1, x_2, x_3) \in \mathbb{Z}^3 \text{ s.t. } |x_i| \leq t, i = 1, 2, 3\}\}$.

For example, $N_6^1(p)$ are the 6-neighbors of p and $N^{\leq 1}(p)$ is the $3 \times 3 \times 3$ block centered at point p , composed of 27 points in \mathbb{Z}^3 .

Definition 1 A 3D binary image $J = (\mathbb{Z}^3, F_J)$ *encodes* a given cubical complex $Q(I)$ if:

- For any cell $\sigma \in Q(I)$, the barycenter of σ is in F_J .
- For any point $p \in F_J$, p is the barycenter of a cell σ in $Q(I)$.

From now on **we identify the cells in $Q(I)$ with the points in F_J that encode them**. This way, when we say “a cell $p \in F_J$,” we refer to the cell $\sigma \in Q(I)$ encoded by such point p .

In the following remark, we show that we can deduce the dimension of a given cell $p \in F_J$ by looking at its type of coordinates, and obtain all its faces by looking at its neighboring cells.

Remark 1 [7] A point $p \in 2\mathbb{Z}^3$ encodes:


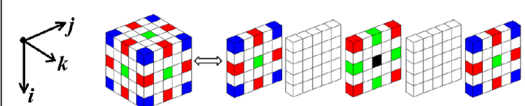
- a vertex in $Q(I)$ iff $p \in \mathcal{E}_0 \cap F_J$;
- an edge in $Q(I)$ iff $p \in \mathcal{E}_1 \cap F_J$;
- a square face in $Q(I)$ iff $p \in \mathcal{E}_2 \cap F_J$;
- a voxel in $Q(I)$ iff $p \in \mathcal{E}_3 \cap F_J$.

Table 1 Different notations used throughout the paper

Notation	Explanation	Section
$I = (\mathbb{Z}^3, F_I)$	A 3D binary digital image (also called picture)	1
$C(I)$	Continuous analog of I	1
$Q(I)$	Cubical complex associated with I	2.1
$\mathcal{E}_i : i = 0, 1, 2, 3$	Subset of $2\mathbb{Z}^3$ used to encode the i -cells of $Q(I)$	2.1
$N_6^i(p), N_{12}^i(p), N_8^i(p), N^{\leq i}(p)$	Different sets of neighbors of p	2.2
$J = (\mathbb{Z}^3, F_J)$	3D binary image encoding all the cells of $Q(I)$	2.2
$C = (\mathbb{Z}^3, F_C)$	A 3D binary image encoding a critical structuring element	2.2
$P(I)$	The well-composed polyhedral complex over the picture I	3
$c, c(v), k(v, w), p_\ell(\dots), h_\ell(\dots)$	Different polyhedra used to construct $P(I)$	3
$E_p = (g_p, B_p, h_p)$	ExtendedCubeMap representation of $P(I)$	3.1
S-block $S(p)$	Specific sets of neighbors of p used in the computation of E_p	3.1
$\mathcal{O}_i : i = 0, 1, 2, 3$	Subsets of $\mathbb{Z}^3 \setminus 2\mathbb{Z}^3$ used to encode the i -cells of $P(I) \setminus Q(I)$	3.2
$L = (\mathbb{Z}^3, F_L)$	3D binary image encoding all the cells of $P(I)$	3.2
$M = (\mathbb{Z}^3, F_M)$	3D binary image encoding the 3-cells of $P(I)$	3.2
$E(p), F(p), C(p)$	Set of corner points of p , being $p \in \mathcal{E}_1$ a size-4 edge, $p \in \mathcal{E}_2$ a size-4 square face and $p \in \mathcal{E}_3$ a size-4 cube, respectively	3.2
$\partial P(I)$	Boundary surface of $P(I)$	4
$H = (\mathbb{Z}^3, F_H)$	3D binary image encoding the 2-cells of $\partial P(I)$	4

Table 2 (Color online) (a.1) A voxel c centered at a point p . (a.2) Set of points encoding c together with all its faces. Color illustrates the type of coordinate of the point, which provides the dimension of the encoded cell: blue for 0-cells (i.e., points in \mathcal{E}_0), red for 1-cells (i.e., points in \mathcal{E}_1), green for 2-cells (i.e., points in \mathcal{E}_2) and black for 3-cells (i.e., points

in \mathcal{E}_3). (a.3) Relative position of the faces of a voxel. $\mathbf{1}$ is at position $(0, 0, 0)$. Given a voxel p , its faces are those $p + q$, with q a non-null position in the array. i -Axis in (a.2) corresponds to rows in array (a.3), j -axis to columns and k -axis to sequence of matrices

Voxel c	ECM representation of c and its faces	3D binary image encoding c and its faces.
a.1) 	a.2) 	a.3) $\begin{pmatrix} 10101 \\ 00000 \\ 10101 \\ 00000 \\ 10101 \end{pmatrix} \begin{pmatrix} 00000 \\ 00000 \\ 00000 \\ 00000 \\ 00000 \end{pmatrix} \begin{pmatrix} 10101 \\ 00000 \\ \underline{10101} \\ 00000 \\ 10101 \end{pmatrix} \begin{pmatrix} 00000 \\ 00000 \\ 00000 \\ 00000 \\ 00000 \end{pmatrix} \begin{pmatrix} 10101 \\ 00000 \\ 10101 \\ 00000 \\ 10101 \end{pmatrix}$

Let $p \in F_J$ be an ℓ -cell of $Q(I)$ then:

- the set of $(\ell - 1)$ -faces of p are $N_6^2(p) \cap \mathcal{E}_{\ell-1} \cap F_J$ if $\ell \geq 1$;
- the set of $(\ell - 2)$ -faces of p are $N_{12}^2(p) \cap \mathcal{E}_{\ell-2} \cap F_J$ if $\ell \geq 2$;
- and the set of $(\ell - 3)$ -faces of p are $N_8^2(p) \cap \mathcal{E}_{\ell-3} \cap F_J$ if $\ell = 3$.

Observe that there exists a bijective correspondence between voxels of $Q(I)$ and points $p \in \mathcal{E}_3 \cap F_J$.

See Table 2 in which a voxel c (on the left), together with all its faces, is encoded by a set of points (on the middle). Colors are used to distinguish dimension of cells. The array in Table 2a.3 provides the relative position of the faces of a voxel. That is, given a voxel p , its faces are those $p + q$, being q a non-null position in the array.

A vertex in $Q(I)$ is *critical* if its neighborhood in the boundary surface of $C(I)$ is not a 2D manifold. More generally, a cell in $Q(I)$ is *critical* if it is incident to a critical vertex. We can find such critical vertices directly in the binary image $J = (\mathbb{Z}^3, F_J)$ by searching for the points $p \in \mathcal{E}_0 \cap F_J$ around which a *critical structuring element* from those in Fig. 2 (module reflections and 90° rotation) fits (see [5]). More specifically, a critical structuring element is a binary image $C = (D_C, F_C)$, defined over certain domain $D_C \subset \mathbb{Z}^3$, composed of background points, B_C , and foreground points, F_C (that is, $D_C = B_C \cup F_C$), such that F_C contains the coordinate origin $o = (0, 0, 0)$ (blue point in Fig. 2) and D_C is a subset of points of $N_8^2(o)$. The structuring element $C = (D_C, F_C)$ is said to fit around a point $p \in \mathcal{E}_0 \cap F_J$ if for each $q \in F_C$, $p + q \in F_J$ and for each $q \in D_C \setminus F_C$, $p + q \in \mathbb{Z}^3 \setminus F_J$.

Finally, regarding time and space complexity, suppose I has n points in its foreground. Then, in the worst case, there

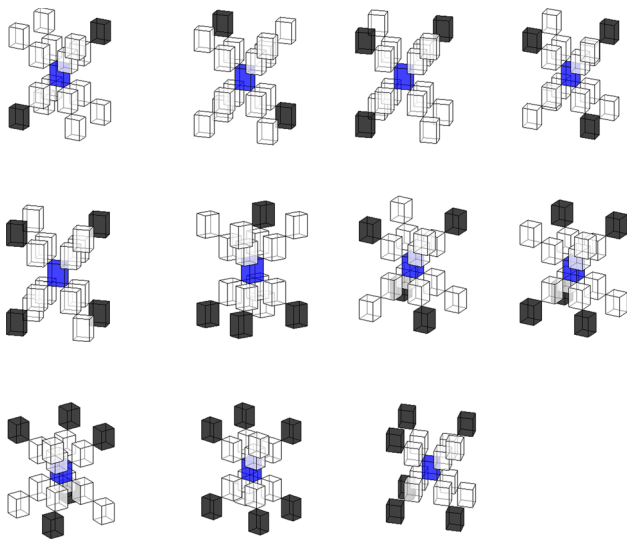


Fig. 2 (Color online) Critical structuring elements (modulo reflections and 90° rotation). These eleven critical structuring elements also appeared in [5, Fig. 4] and they encode the eleven 8-cube critical configurations around a vertex (in blue)

are $8n$ points in F_J encoding vertices of $Q(I)$. For each vertex, we have to see if one of the critical structuring elements fits. So, the set $R \subset F_J$ of critical vertices of $Q(I)$ can be obtained in linear time, and the number of points in R is at most $8n$.

3 Well-Composed Polyhedral Complexes Over Pictures

Roughly speaking, a regular cell complex X is a collection of ℓ -dimensional cells glued together by their boundaries (*faces*), in such a way that a non-empty intersection of any two cells of X is a cell in X . A regular cell complex is a polyhedral complex if its ℓ -cells are ℓ -polytopes. In particular, a cubical complex is a polyhedral complex whose ℓ -cells are ℓ -cubes. If an ℓ -cell $\sigma \in X$ lies in the boundary of an ℓ' -cell $\sigma' \in X$, being $\ell \leq \ell'$, then we say that σ is a *face* of σ' and, conversely, σ' is *incident to* σ . A cell μ is *maximal* if it is not a face of any other cell $\sigma \in X$. In this paper, we deal with complexes embedded in \mathbb{R}^3 . This way, a 3D polyhedral complex K [12] is a combinatorial structure by which a space is decomposed into vertices (0-cells), edges (1-cells), polygons (2-cells) and polyhedra (3-cells).

Two 3D polyhedral complexes K and K' are *homotopy equivalent* if the geometric realization, $|K|$, of K can be continuously deformed into the geometric realization, $|K'|$, of K' . That is, there exist continuous functions $f : |K| \rightarrow |K'|$, $g : |K'| \rightarrow |K|$, $H : |K| \times [0, 1] \rightarrow |K'|$ and $H' : |K'| \times [0, 1] \rightarrow |K|$ such that if $x \in |K|$ then $H(x, 0) = gf(x)$ and $H(x, 1) = x$; if $x' \in |K'|$ then $H(x', 0) = fg(x')$ and $H(x', 1) = x'$ (see [9]).

The subcomplex ∂K of K is constructed as follows: add an ℓ -cell $\sigma \in K$ to ∂K , together with all its faces, if σ is face of exactly one $(\ell + 1)$ -cell of K . A 3D polyhedral complex K is *complete* if all its maximal cells have dimension 3. Finally, K is *well composed* if it is complete and the geometric realization of ∂K is a 2D manifold. In this case, the geometric realization of ∂K is the boundary surface of the geometric realization of K . From now on, if there is no confusion, we **identify the complexes K and ∂K with their geometric realizations**.

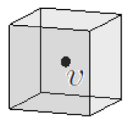
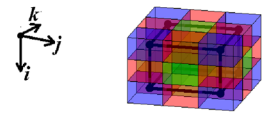
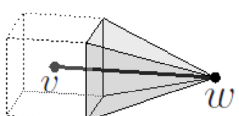
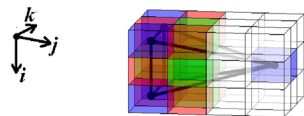
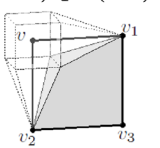
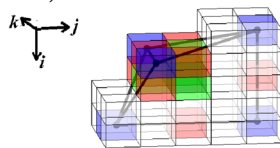
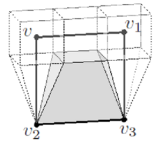
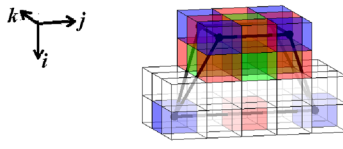
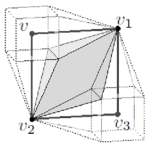
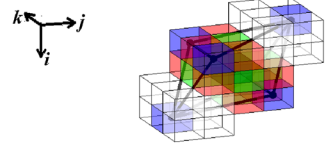
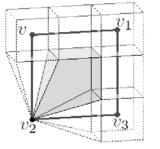
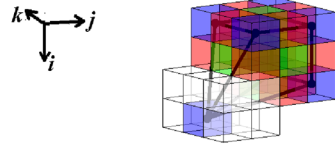
In [5], we developed a procedure to get as output a 3D well-composed polyhedral complex $P(I)$ homotopy equivalent to the cubical complex $Q(I)$ representing an input 3D binary image I . $P(I)$ is said to be the *well-composed polyhedral complex over the picture I* . The idea under the geometric construction of $P(I)$ is to replace each critical vertex v by a size-2 cube and all the cells (of any dimension) incident to v by specific polyhedra accordingly, so that all the pieces fit appropriately. The set S of 27 types of polyhedra used in [5] to construct $P(I)$ is described below.

Definition 2 The set S of polyhedra used to construct $P(I)$ consists of:

- (a) The voxel (size-4 cube) c centered at a point in \mathcal{E}_3 . See Table 2a.1.
- (b) The size-2 cube $c(v)$ centered at a point $v \in 2\mathbb{Z}^3$ with faces parallel to the coordinate planes. See Table 3b.1. Let $s(v)$ denote a square face of $c(v)$.
- (c) The pyramid $k(v, w)$, determined by $v, w \in \mathcal{E}_0$, such that the edge with endpoints v and w (denoted by $e(v, w)$) is a size-4 edge and the pyramid has apex w and base the square face $s(v)$ whose barycenter lies on $e(v, w)$. See Table 3c.1. Let $t(v, w)$ denote a triangle face of $k(v, w)$.
- (d) The polyhedra $\{p_\ell(v, v_1, v_2, v_3)\}_{\ell=1,2,3,4}$, where $v, v_1, v_2, v_3 \in \mathcal{E}_0$, are distinct points forming a size-4 square face s , being v adjacent to v_1 and v_2 . Besides,
 - $p_1(v, v_1, v_2, v_3)$ is determined by the edges $e(v_1, v_3)$ and $e(v_2, v_3)$, and the triangle faces $t(v, v_1)$ and $t(v, v_2)$, whose barycenters lie inside s . See Table 3d.1.
 - $p_2(v, v_1, v_2, v_3)$ is determined by the edge $e(v_2, v_3)$, the triangle faces $t(v, v_2), t(v_1, v_3)$, and the square face $s(\frac{v+v_1}{2})$, whose barycenters lie inside the square face s . See Table 3e.1.
 - $p_3(v, v_1, v_2, v_3)$ is determined by the four triangle faces $t(v, v_1), t(v, v_2), t(v_3, v_1)$ and $t(v_3, v_2)$ whose barycenters lie inside s . See Table 3f.1.
 - $p_4(v, v_1, v_2, v_3)$ is determined by the square faces $s(\frac{v+v_1}{2})$ and $s(\frac{v_1+v_3}{2})$, and the triangle faces $t(v, v_2)$ and $t(v_3, v_2)$, whose barycenters lie inside s . See Table 3g.1.

Table 3 (Color online) First column: 7 out of 27 polyhedra used to construct well-composed polyhedral complexes over pictures. Second column: Grayscale image corresponding to the ECM representation of each polyhedron p of the first column (together with all its faces). Third column: The 3D binary image encoding p together with all its faces.

i -Axis corresponds to rows in the arrays of third column, j -axis to columns and k -axis to sequence of matrices. Finally, empty entries in the matrices correspond to ones, which encode other cells. We did not add them to focus on the points encoding the polyhedron and its faces, in each case. Finally, observe that $\mathbf{1}$ encodes the polyhedron p

Polyhedron p	ECM rep. of p and its faces	3D binary image encoding p and its faces.
b.1) $c(v)$ 	b.2) 	b.3) $\begin{pmatrix} 111 \\ 111 \\ 111 \end{pmatrix} \begin{pmatrix} 111 \\ \mathbf{1}11 \\ 111 \end{pmatrix} \begin{pmatrix} 111 \\ 111 \\ 111 \end{pmatrix}$
c.1) $k(v, w)$ 	c.2) 	c.3) $\begin{pmatrix} 1100 \\ 1100 \\ 1100 \end{pmatrix} \begin{pmatrix} 1100 \\ \mathbf{1}101 \\ 1100 \end{pmatrix} \begin{pmatrix} 1100 \\ 1100 \\ 1100 \end{pmatrix}$
d.1) $p_1(\dots)$ 	d.2) 	d.3) $\begin{pmatrix} & 00 \\ 1100 \\ 1100 \\ 00000 \\ 00000 \end{pmatrix} \begin{pmatrix} & 01 \\ 1100 \\ \mathbf{1}101 \\ 00000 \\ 10101 \end{pmatrix} \begin{pmatrix} & 00 \\ 1100 \\ 1100 \\ 00000 \\ 00000 \end{pmatrix}$
e.1) $p_2(\dots)$ 	e.2) 	e.3) $\begin{pmatrix} 111 \\ 111 \\ 00000 \\ 00000 \end{pmatrix} \begin{pmatrix} 111 \\ \mathbf{1}11 \\ 00000 \\ 10101 \end{pmatrix} \begin{pmatrix} 111 \\ 111 \\ 00000 \\ 00000 \end{pmatrix}$
f.1) $p_3(\dots)$ 	f.2) 	f.3) $\begin{pmatrix} & 00 \\ 1100 \\ 111 \\ 0011 \\ 00 \end{pmatrix} \begin{pmatrix} & 01 \\ 1100 \\ \mathbf{1}11 \\ 0011 \\ 10 \end{pmatrix} \begin{pmatrix} & 00 \\ 1100 \\ 111 \\ 0011 \\ 00 \end{pmatrix}$
g.1) $p_4(\dots)$ 	g.2) 	g.3) $\begin{pmatrix} 111 \\ 111 \\ 0011 \\ 00 \end{pmatrix} \begin{pmatrix} 111 \\ \mathbf{1}11 \\ 0011 \\ 10 \end{pmatrix} \begin{pmatrix} 111 \\ 111 \\ 0011 \\ 00 \end{pmatrix}$

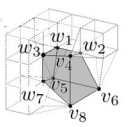
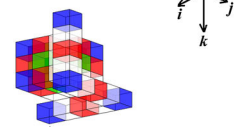
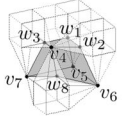
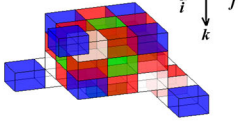
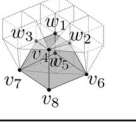
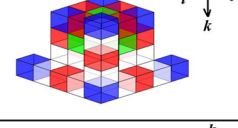
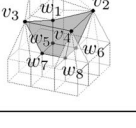
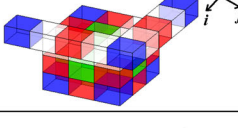
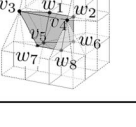
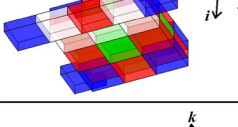
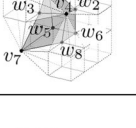
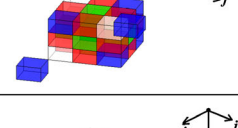
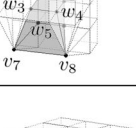
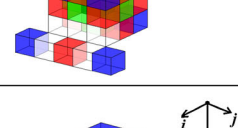
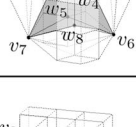
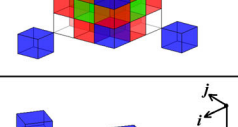
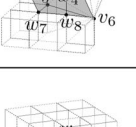
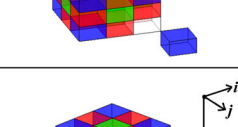
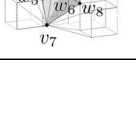
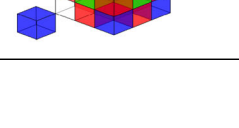
(e) The 22 hexahedra $\{h_\ell(v_1, \dots, v_8)\}_{\ell=1, \dots, 22}$ (where $\{v_1, \dots, v_8\} \subset \mathcal{E}_0$ are the vertices of a voxel c) and are determined by a set of vertices $\{x_1, \dots, x_8\}$. Each vertex x_i is either v_i or the vertex w_i of $c(v_i)$ that lies inside c . Observe that $h_1(v_1, \dots, v_8)$ (when $x_i = v_i$ for $i = 1, \dots, 8$) is the voxel c itself. Similarly, $h_{22}(v_1, \dots, v_8)$ (when $x_i = w_i$ for $i = 1, \dots, 8$) is the size-2 cube $c(\frac{v_1+\dots+v_8}{8})$ being $\frac{v_1+\dots+v_8}{8} \in \mathcal{E}_3$. All the other possible vertex combinations lead to the 20 hexahedra shown in first column of Tables 4 and 5.

Observe that some of the 2-faces of $p_\ell(v, v_1, v_2, v_3)$, $\ell = 1, 3, 4$, are not planar. To convert $p_\ell(v, v_1, v_2, v_3)$, $\ell = 1, 3, 4$ into a polyhedron, we replace each non-planar 2-face by an edge and two triangle faces as shown in Fig. 3. From now on, **we identify** $p_\ell(v, v_1, v_2, v_3)$, $\ell = 1, 3, 4$, **with the polyhedron obtained.**

In Fig. 4, a simple example is shown of how the critical cells of $Q(I)$ are replaced by some specific polyhedra from the set S to obtain $P(I)$.

Table 5 (Color online) First column: List of 10 out of 22 types of hexahedra $h_\ell(v_1, \dots, v_8)$ described in Definition 2.e. Second column: Grayscale image corresponding to the ECM representation of each hex-

ahedron h of the first column, together with all its faces. Third column: 3D binary image encoding the hexahedron $h_\ell(v_1, \dots, v_8)$ and its faces. $\underline{1}$ is at position $\frac{v_1+\dots+v_8}{8}$

Hexahedron h	ECM rep. for h	3D binary image encoding h . $\underline{1}$ is at position $(0, 0, 0)$.
		$\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 \\ 1 & \underline{1} & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{pmatrix}$
		$\begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 \\ 1 & \underline{1} & 1 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 1 & 0 \end{pmatrix}$
		$\begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 \\ 1 & \underline{1} & 1 \\ 1 & 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{pmatrix}$
		$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 \\ 1 & \underline{1} & 1 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{pmatrix}$
		$\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 \\ 1 & \underline{1} & 1 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{pmatrix}$
		$\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 \\ 1 & \underline{1} & 1 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{pmatrix}$
		$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 \\ 1 & \underline{1} & 1 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{pmatrix}$
		$\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 \\ 1 & \underline{1} & 1 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 0 & 0 \\ 1 & 0 \end{pmatrix}$
		$\begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 \\ 1 & \underline{1} & 1 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$
		$\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 \\ 1 & \underline{1} & 1 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$

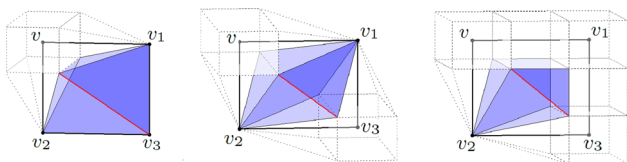


Fig. 3 Replacing each non-planar face in $p_\ell(v, v_1, v_2, v_3)$, $\ell = 1, 3, 4$ by an edge and two triangle faces

3.1 Computing $P(I)$ Via Its ExtendedCubeMap (ECM) Representation

In this subsection, we will see how the well-composed polyhedral complex $P(I)$ over a picture I is computed via its ExtendedCubeMap representation, which consists of a grayscale image $g_P : D_P \rightarrow \mathbb{Z}$, a bijection $h_P : D_P \rightarrow P(I)$ and a set B_P of structuring elements providing both geometric information and boundary face relations between the cells of $P(I)$. We recall now the formal definition of ECM representation of $P(I)$.

First, recall that given a 3D grayscale image $g : \mathbb{Z}^3 \rightarrow \mathbb{Z}$, a *structuring element* is a grayscale image $b : D_b \subset \mathbb{Z}^3 \rightarrow \mathbb{Z}$ defined on a certain domain D_b , with $o = (0, 0, 0) \in D_b$; the structuring element b is said to *fit around a point* $p \in \mathbb{Z}^3$ (with respect to g) if, for any $q \in D_b$, $g(p + q) = b(q)$.

Definition 3 [5] An *ExtendedCubeMap (ECM) representation* of the well-composed polyhedral complex $P(I)$ over a picture I is a triple $E_P = (g_P, B_P, h_P)$ where:

- $h_P : D_P \rightarrow P(I)$ is a bijective function, for a certain domain $D_P \subset \mathbb{Z}^3$ with as many points as cells in $P(I)$. For each cell σ , the point $p_\sigma = h_P^{-1}(\sigma)$ encodes σ .
- $g_P : \mathbb{Z}^3 \rightarrow \{-1, 0, 1, 2, 3\}$ is a 3D grayscale image, such that:
 - $g_P(p)$ is the dimension of $h_P(p)$ for $p \in D_P$;
 - $g_P(p) = -1$ for $p \in \mathbb{Z}^3 \setminus D_P$.
- B_P is the set of structuring elements $\{b : D_b \subset \mathbb{Z}^3 \rightarrow \mathbb{Z}\}$ shown in Fig. 5 (modulo reflections and 90-degree rotations) satisfying that for any ℓ -cell $\sigma \in P(I)$, there

exists a single structuring element $b_\sigma : D_\sigma \rightarrow \mathbb{Z}$ in B_P , with $o = (0, 0, 0) \in D_\sigma$ and $b_\sigma(o) = \ell$, such that:

- b_σ fits around p_σ .
- A point $p \in \mathbb{Z}^3$ encodes an $(\ell - 1)$ -face of σ if and only if $p - p_\sigma \in D_\sigma$ and $b_\sigma(p - p_\sigma) = \ell - 1$.

Examples of the grayscale image $g_P : D_P \rightarrow \mathbb{Z}^3$ are illustrated in the second column of all the tables of the paper. The values of g_P , which coincide with the values of the dimension of the cells, are represented by colors: blue for 0-cells, red for 1-cells, green for 2-cells and black for 3-cells. Besides, Fig. 5 shows the set B_P of structuring elements needed to compute the points in D_P encoding the boundary cells of any cell in $P(I)$.

Before explaining how to construct the ECM representation of the well-composed polyhedral complex $P(I)$ over a picture I , we need to introduce the following technical definitions. The S_ℓ -block $S_\ell(p)$, centered at point $p \in 2\mathbb{Z}^3$, is defined as follows (see Fig. 6):

- $S_0((4i + 2, 4j + 2, 4k + 2)) = \{(4i + 2 \pm t_1, 4j + 2 \pm t_2, 4k + 2 \pm t_3)\}_{t_1, t_2, t_3 \in \{0, 1\}}$.
- $S_1((4i + 2, 4j + 2, 4k)) = \{(4i + 2 \pm t_1, 4j + 2 \pm t_2, 4k)\}_{t_1, t_2 \in \{0, 1\}}$;
- In analogous way, $S_1((4i + 2, 4j, 4k + 2))$ and $S_1((4i, 4j + 2, 4k + 2))$ can be defined;
- $S_2((4i + 2, 4j, 4k)) = \{(4i + 2 \pm t, 4j, 4k)\}_{t \in \{0, 1\}}$;
- In analogous way, $S_2((4i, 4j + 2, 4k))$ and $S_2((4i, 4j, 4k + 2))$ can be defined;
- $S_3((4i, 4j, 4k)) = \{(4i, 4j, 4k)\}$.

In Fig. 6b, the S_ℓ -blocks centered at the points encoding ℓ -cells incident to p are shown. In the sequel, sometimes we will omit the subindex and write S -block instead of S_ℓ -block. Notice that the S -blocks are disjoint sets and that $\mathbb{Z}^3 = \bigsqcup_{p \in 2\mathbb{Z}^3} S(p)$. Also notice that given a point $p \in \mathcal{E}_0$, the S -blocks $S(q)$ centered at points $q \in N^{\leq 2}(p)$ are subsets of $N^{\leq 2}(p)$.

Now, we recall the method explained in [5] that is performed on the binary image $J = (\mathbb{Z}^3, F_J)$ encoding $Q(I)$,

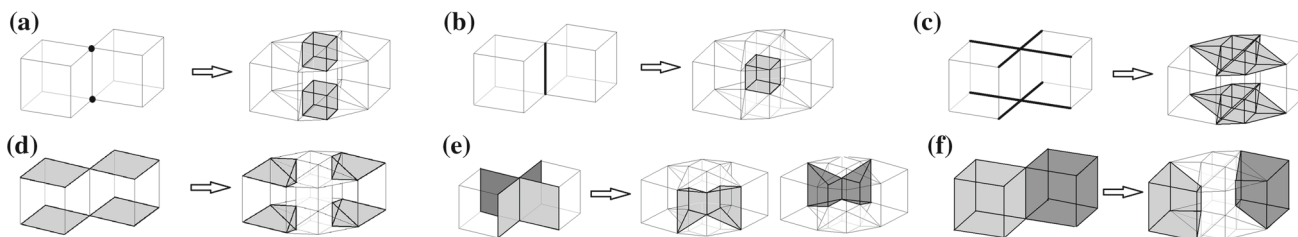


Fig. 4 Critical configuration of two cubes sharing an edge: **a** each critical vertex is replaced by a size-2 cube; **b** the shared critical edge is replaced by a size-2 cube; **c** the other critical edges are replaced by pyra-

mids; **d** four critical faces are replaced by polyhedra of type $p_1(\dots)$; **e** the other critical edges are replaced by polyhedra of type $p_2(\dots)$; **f** each voxel is replaced by a hexahedron of type $h_2(\dots)$

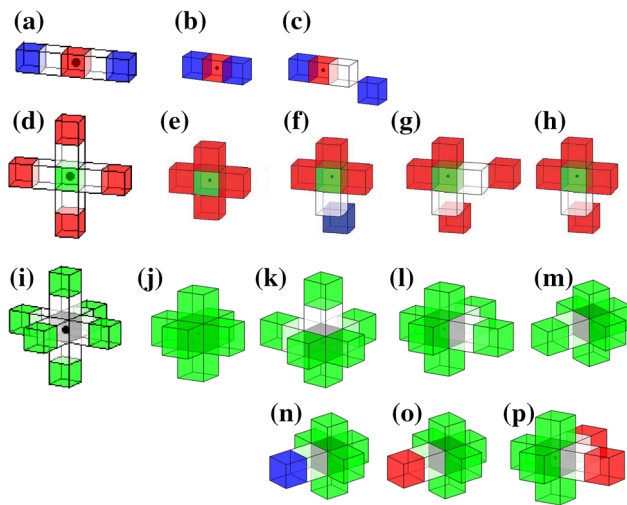


Fig. 5 (Color online) Set B_P of structuring elements (modulo reflections and 90° rotations) for computing boundary face relations (see [5]): three structuring elements for the boundary of a 1-cell (first row); five structuring elements for the boundary of a 2-cell (second row); eight structuring elements for the boundary of a 3-cell (third and fourth rows)

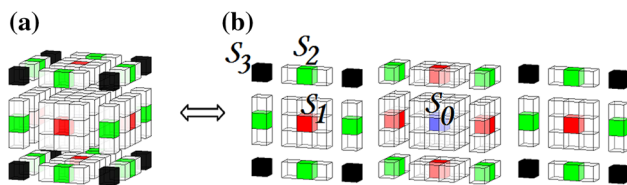


Fig. 6 (Color online) **a** Voxel colors on $N^{\leq 2}(p)$ of the canonical ECM representation of the well-composed polyhedral complex $P(I)$ over a picture I composed by eight cubes sharing a common vertex p . **b** The different types of S_ℓ -blocks, $\ell = 0, 1, 2, 3$, have been spaced for a better visualization

to directly construct a map $g_P : \mathbb{Z}^3 \rightarrow \mathbb{Z}$ (by “coloring” the S -blocks centered at critical cells in $Q(I)$) and a set D_P of points in \mathbb{Z}^3 . In [5, Proc. 2] we showed that, the one-to-one correspondence $h_P : D_P \rightarrow P(I)$ can be univocally defined given the points in D_P and the cells in the well-composed polyhedral complex $P(I)$ over the picture I . Besides, in [5, Proposition 4] we showed that for each point $p \in D_P$ with $g_P(p) \geq 1$, exactly one of the structuring elements in B_P given in Fig. 5 fits around p . This way, the set (g_P, B_P, h_P) obtained after applying Procedure 1 is the ECM representation E_P of the well-composed polyhedral complex $P(I)$ over the picture I .

Procedure 1 [5] *Computing the ECM representation of the well-composed polyhedral complex $P(I)$ over a picture I .*

INPUT: The 3D binary image $J = (\mathbb{Z}^3, F_J)$ encoding the cubical complex $Q(I)$ and the set $R \subset F_J$ of critical vertices of $Q(I)$.

OUTPUT: The grayscale image g_P and the set D_P .

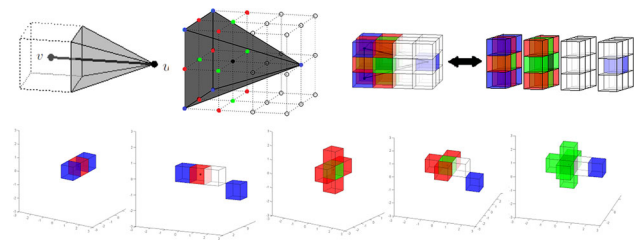


Fig. 7 (Color online) *Top row* A pyramid $k(v, w)$; a point-based picture of the values of g_P for the cells of the pyramid; a voxel-based picture of the values of g_P . *Bottom row* set of structuring elements (modulo 90° rotations) that can be applied

Initialize $D_P := F_J$ and $g_P(p) := \ell$ if p is an ℓ -cell in F_J and $g_P(p) := -1$ otherwise.

For each $p \in R$:

For each $q \in N^{\leq 2}(p) \cap F_J$:

For each $s \in S(q)$:

Set $g_P(s)$ to 3 if $s = q$;

Set $g_P(s)$ to 2 if $s \in N_6^1(q) \cap S(q)$;

Set $g_P(s)$ to 1 if $s \in N_{12}^1(q) \cap S(q)$;

Set $g_P(s)$ to 0 if $s \in N_8^1(q) \cap S(q)$.

Add s to D_P .

Notice that if q encodes an ℓ -critical cell in $Q(I)$, then all the points in $S(q)$ are colored.

Concerning time and space complexity of Procedure 1, observe that if I has n points in its foreground, then F_J has at most $(8+12+6+1)n = 27n$ points, and the set R can have at most $8n$ points (which is the number of critical vertices in $Q(I)$ in the worst case). Then, D_P can have at most $9 \cdot 27 \cdot 8n$ points since, in the worst case, we add to D_P all the points in $S(q)$ (which are 9 at most) for all the points in $N^{\leq 2}(p) \cap F_J$ (27 at most) for $p \in R$. Finally, the operation that assigns a color to a point and set operations are considered constant time. Then, worst-time complexity is $O(9 \cdot 27 \cdot 8n)$.

For a better understanding of the concept of ECM representation, observe Fig. 7 (top row) where, for a pyramid $k(v, w)$, both point-based and voxel-based pictures illustrate, with colors, the values of the image g_P overlapped with the pyramid itself. All the structuring elements that provide the boundary faces of each type of cell are depicted at the bottom row.

3.2 3D Binary Images for Storing Well-Composed Polyhedral Complexes over Pictures

Now, we will show that geometric and topological information of the well-composed polyhedral complex $P(I)$ over a picture I can be stored just by a 3D binary image $L = (\mathbb{Z}^3, F_L)$ being $F_L = D_P$, in such a way that neither

grayscale image g_P nor set of structuring elements B_P are needed. We will say then that L encodes $P(I)$.

For the sake of brevity **we will identify the cells of $P(I)$ with the points that encode them**. Notice that all the coordinates used for encoding the cubical complex $Q(I)$ as a 3D binary image were even coordinates. Now, odd coordinates will also be needed to encode boundary cells of the polyhedra that have been added to $P(I)$ to replace the critical cells in $Q(I)$. We denote:

- $\mathcal{O}_0 = \{(2i + 1, 2j + 1, 2k + 1)\}_{i,j,k \in \mathbb{Z}}$;
- $\mathcal{O}_1 = \{(2i, 2j + 1, 2k + 1), (2i + 1, 2j, 2k + 1), (2i + 1, 2j + 1, 2k)\}_{i,j,k \in \mathbb{Z}}$;
- $\mathcal{O}_2 = \{(2i, 2j, 2k + 1), (2i, 2j + 1, 2k), (2i + 1, 2j, 2k)\}_{i,j,k \in \mathbb{Z}}$.

Another notation that will be useful in the sequel is the one of the sets $E(p), F(p), C(p)$, defined as follows:

- If $p \in \mathcal{E}_1$, then $E(p) = N_6^2(p) \cap \mathcal{E}_0$ consists of the two endpoints of the size-4 **edge** p . For example, if $p = (4i, 4j + 2, 4k + 2)$, then $E(p) = \{(4i \pm 2, 4j + 2, 4k + 2)\}$.
- If $p \in \mathcal{E}_2$, then $F(p) = N_{12}^2(p) \cap \mathcal{E}_0$ consists of the four corner points of the size-4 **square face** p . For example, if $p = (4i, 4j, 4k + 2)$, then $F(p) = \{(4i \pm 2, 4j \pm 2, 4k + 2)\}$.
- If $p \in \mathcal{E}_3$, then $C(p) = N_8^2(p)$ consists of the eight corner points of the size-4 **cube** p . This way, if $p = (4i, 4j, 4k)$, then $C(p) = \{(4i \pm 2, 4j \pm 2, 4k \pm 2)\}$.

In [7, Proposition 2], we proved that, for each point $p \in D_P$, we can identify the cell of $P(I)$ encoded by examining the type of coordinates of p and the configuration of points in the neighborhood $N^{\leq 2}(p)$, showing that color information stored in g_P is redundant. Now, Proposition 1 improves that result since it provides a characterization of each type of polyhedron p of $P(I)$ in terms of the neighborhoods $N^{\leq 1}(q)$ of specific values $q \in N^{\leq 2}(p)$, depending on the coordinates of p , what will allow to obviate a big part of information stored, as we will see later.

Proposition 1 *Let $L = (\mathbb{Z}^3, F_L)$ be the 3D binary image encoding the polyhedral complex $P(I)$ over a picture I . The polyhedron that a given point $p \in F_L$ encodes can be characterized as follows:*

- (a) *If $p \in \mathcal{E}_0$: p is the size-2 cube $c(p)$ described in Definition 2.b iff $N^{\leq 1}(p) \subseteq F_L$.*
- (b) *If $p \in \mathcal{E}_1$:*
 - *p is the pyramid $k(\dots)$ described in Definition 2.c iff $N^{\leq 1}(q) \subseteq F_L$ for exactly one of the two points $q \in E(p)$;*

- *p is the size-2 cube $c(p)$ described in Definition 2.b iff $N^{\leq 1}(q) \subseteq F_L$ for the two points $q \in E(p)$.*

- (c) *If $p \in \mathcal{E}_2$:*
 - *p is the polyhedron $p_1(\dots)$, described in Definition 2.d iff $N^{\leq 1}(q) \subseteq F_L$, for exactly one of the four points $q \in F(p)$;*
 - *p is the polyhedron $p_\ell(\dots)$, $\ell = 2, 3$, described in Definition 2.d iff $N^{\leq 1}(q) \subseteq F_L$ for exactly two of the four points $q \in F(p)$. In fact, $\ell = 2$ if the distance between the two points is 4, and $\ell = 3$ otherwise;*
 - *p is the polyhedron $p_4(\dots)$ described in Definition 2.d iff $N^{\leq 1}(q) \subseteq F_L$ for exactly three of the four points $q \in F(p)$.*
 - *p is a size-2 cube iff $p \in \mathcal{E}_2$ and $N^{\leq 1}(q) \subseteq F_L$ for the four points $q \in F(p)$.*

- (d) *If $p \in \mathcal{E}_3$: p is one of the 22 hexahedra listed in Definition 2.e. In fact,*
 - *p is the voxel described in Definition 2.a iff none of the eight points $q \in C(p)$ satisfy that $N^{\leq 1}(q) \subseteq F_L$.*
 - *p is one of the 20 hexahedra listed in Tables 4 and 5 iff at least one and no more than seven points $q \in C(p)$ satisfy that $N^{\leq 1}(q) \subseteq F_L$. The specific hexahedron that encodes p is determined by the number and relative position of the points $q \in C(p)$ satisfying that $N^{\leq 1}(q) \subseteq F_L$;*
 - *p is a size-2 cube iff all the eight points $q \in C(p)$ satisfy that $N^{\leq 1}(q) \subseteq F_L$.*

Now, we review the coordinate types of the faces of all the polyhedra described in Definition 2.

Proposition 2 *Let $L = (\mathbb{Z}^3, F_L)$ be the 3D binary image encoding the polyhedral complex $P(I)$ over a picture I . Then, the ℓ -cells of $P(I)$, for $\ell = 0, 1, 2$, are encoded by either points in \mathcal{O}_ℓ or in \mathcal{E}_ℓ .*

In [5], the $(\ell - 1)$ -faces of a given ℓ -cell $p \in F_L$ were obtained by running over the set B_P of 121 structuring elements listed in Fig. 5 (mod reflections and 90 degree rotations) and looking for the ones that fitted around p . The following procedure computes the $(\ell - 1)$ -faces of p without making use of the set B_P .

Procedure 2 (Proposition 3 in [7]) *Computing the $(\ell - 1)$ -faces of a given cell $p \in F_L$.*

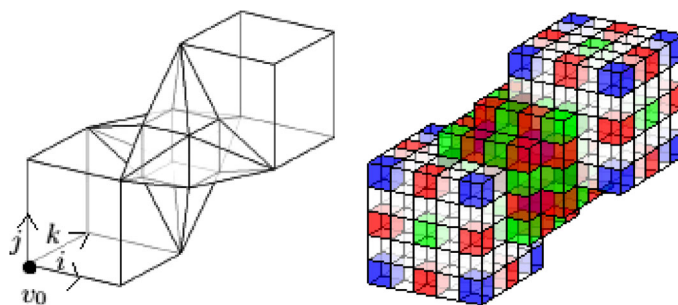
INPUT: The 3D binary image $L = (\mathbb{Z}^3, F_L)$ encoding the well-composed polyhedral complex $P(I)$ over a picture I and a cell $p \in F_L$.

OUTPUT: The $(\ell - 1)$ -faces of p .

For each $q \in N_6^1(p)$ do:

If $q \in F_L \cap \mathcal{O}_{\ell-1}$, then q is an $(\ell - 1)$ -face of p .

Fig. 8 (Color online) On *top-left*, a naive example of the well-composed polyhedral complex $P(I)$ over a picture I (where vertex v_0 has coordinates $(2, 2, 2)$). On *top-right*, the grayscale image of its ECM codification. *Color* illustrates the dimension of the cell. On *bottom*, the 3D binary image L encoding $P(I)$ seen as a set of binary matrices. Notice that values on the i -axis correspond to rows, j -axis to columns and k -axis to sequences of matrices. The set of points in the foreground of the minimal encoding M of $P(I)$ are the *underlined ones*



$\begin{pmatrix} 00000000 \\ 00000000 \\ 00000000 \\ 00000000 \\ 10101000 \\ 00000000 \\ 10101000 \\ 00000000 \\ 10101000 \end{pmatrix}$	$\begin{pmatrix} 00000000 \\ 00000000 \\ 00000000 \\ 00000000 \\ 00000000 \\ 00000000 \\ 00000000 \\ 00000000 \\ 00000000 \end{pmatrix}$	$\begin{pmatrix} 00000000 \\ 00000000 \\ 00000000 \\ 00111100 \\ 10\underline{111100} \\ 00111100 \\ 10\underline{111100} \\ 00000000 \\ 10101000 \end{pmatrix}$	$\begin{pmatrix} 00000000 \\ 00000000 \\ 00011110 \\ 00111110 \\ 00111110 \\ 00111110 \\ 00111110 \\ 00000000 \\ 00000000 \end{pmatrix}$	$\begin{pmatrix} 000010101 \\ 00000000 \\ 0001\underline{11101} \\ 00111110 \\ 10\underline{1111101} \\ 00111110 \\ 10\underline{1111000} \\ 00000000 \\ 10101000 \end{pmatrix}$
$\begin{pmatrix} 00000000 \\ 00000000 \\ 00011110 \\ 00111110 \\ 00111110 \\ 00111110 \\ 00111100 \\ 00000000 \\ 00000000 \end{pmatrix}$	$\begin{pmatrix} 000010101 \\ 00000000 \\ 0001\underline{11101} \\ 00011110 \\ 0001\underline{11101} \\ 00011110 \\ 00000000 \\ 00000000 \\ 00000000 \end{pmatrix}$	$\begin{pmatrix} 00000000 \\ 00000000 \\ 00000000 \\ 00000000 \\ 00000000 \\ 00000000 \\ 00000000 \\ 00000000 \\ 00000000 \end{pmatrix}$	$\begin{pmatrix} 000010101 \\ 00000000 \\ 000010101 \\ 00000000 \\ 000010101 \\ 00000000 \\ 00000000 \\ 00000000 \\ 00000000 \end{pmatrix}$	

If $q \notin F_L$, then
 If $q' = p + 2(q - p) \in F_L \cap \mathcal{E}_{\ell-1}$, then q' is an $(\ell - 1)$ -face of p ;
 Else,
 If there exists $q'' \in F_L \cap N_8^1(q) \cap \mathcal{E}_{\ell-1}$ or $q'' \in F_L \cap N_{12}^1(q) \cap \mathcal{E}_{\ell-1}$, then q'' is an $(\ell - 1)$ -face of p .

I , provides geometric information of all the cells in $P(I)$ as well as their boundary face relations.

Now, we go further and define the minimal encoding of the well-composed polyhedral complex $P(I)$ over a picture I showing that just storing one point per polyhedron is enough to recover geometric and topological information of $P(I)$.

Definition 4 A 3D binary image $M = (\mathbb{Z}^3, F_M)$ is a *minimal encoding* of a 3D well-composed polyhedral complex $P(I)$ over a picture I , if F_M is composed of all the points in F_L encoding a polyhedron.

Observe that Procedure 2 describes, in fact, the way to obtain all of structuring elements in Fig. 5. Figure 7 shows an example of a polyhedron together with the set of structuring elements describing the $(\ell - 1)$ -faces of each cell involved. Procedure 2 provides those configurations.

In Fig. 8, a small example of a well-composed polyhedral complex $P(I)$ over a picture I is shown together with its codification under the form of a 3D binary image L given as a set of matrices. The set of points in F_M (the foreground of the minimal codification of $P(I)$) are the underlined ones. Observe that F_M is composed of just 15 points that correspond to the 15 polyhedra that form $P(I)$. On the other hand, F_L (the foreground of L) is composed of 139 points.

Finally, the worst-time complexity of Procedure 2 is $O(6 \cdot 5)$, which is constant, since for each $q \in N_6^1(p)$ we have to check at most 5 if-conditions and an ℓ -cell $p \in F_L$ can have at most six $(\ell - 1)$ -faces.

The following result guarantees that neither grayscale image, nor structuring elements are needed and, hence, storing just the 3D binary image $L = (\mathbb{Z}^3, F_L)$ is enough to retrieve both geometric and topological information of $P(I)$.

Now, we show how to compute the minimal encoding of $P(I)$, directly from the 3D binary image $J = (\mathbb{Z}^3, F_J)$ encoding the cubical complex $Q(I)$. For this aim, critical vertices have to be detected using the critical structuring elements (modulo reflections and 90-degree rotations) shown in

Theorem 1 The 3D binary image $L = (\mathbb{Z}^3, F_L)$, encoding the well-composed polyhedral complex $P(I)$ over a picture

Fig. 2 under their binary form. Notice that, in $P(I)$, each cell comes either from a cube (of the initial cubical complex) or from a polyhedra replacing a critical cell.

Procedure 3 Computing the minimal encoding $M = (\mathbb{Z}^3, F_M)$ of the well-composed polyhedral complex $P(I)$ over a picture I .

INPUT: The 3D binary image $J = (\mathbb{Z}^3, F_J)$ encoding the cubical complex $Q(I)$ and the set $R \subset F_J$ of critical vertices of $Q(I)$.

OUTPUT: The 3D binary image $M = (\mathbb{Z}^3, F_M)$.

Initially, $F_M := F_J \cap \mathcal{E}_3$.

For each point $p \in R$ do:

Add p to F_M .

Add to F_M the points of the set $\mathcal{E}_1 \cap N_6^2(p) \cap F_J$.

Add to F_M the points of the set $\mathcal{E}_2 \cap N_{12}^2(p) \cap F_J$.

Suppose I has n points. Then, F_J has at most $27n$ points and R at most $8n$. Since $F_M \subset F_J$ then F_M has at most $27n$ points. Besides, worst-time complexity is $O(8n)$, since we assume that adding points to a set can be done in constant time. Notice that the computation of the set R is also linear.

Proposition 3 The result of applying Procedure 3 to the 3D binary image $J = (\mathbb{Z}^3, F_J)$ encoding the cubical complex $Q(I)$ is a minimal encoding of $P(I)$.

In the following procedure, we show how to retrieve the 3D binary image $L = (\mathbb{Z}^3, F_L)$ encoding the well-composed polyhedral complex $P(I)$ over a picture I from the minimal encoding $M = (\mathbb{Z}^3, F_M)$ of $P(I)$.

Procedure 4 Computing the 3D binary image $L = (\mathbb{Z}^3, F_L)$ from the minimal encoding $M = (\mathbb{Z}^3, F_M)$ of the well-composed polyhedral complex $P(I)$ over a picture I .

INPUT: The minimal encoding $M = (\mathbb{Z}^3, F_M)$ of $P(I)$.

OUTPUT: A 3D binary image $L = (\mathbb{Z}^3, F_L)$.

For each $p \in F_M$ do:

1. Add to F_L the points in the block $S(p)$.
2. If $p \in \mathcal{E}_1$, let v, w denote the vertices in $E(p)$.
Then:
 - 2.1 If $E(p) \cap F_M = \{v\}$, add to F_L the vertex w .
3. If $p \in \mathcal{E}_2$, let v, v_1, v_2, v_3 denote the vertices in $F(p)$ (with relative positions as shown in Table 3). Then:
 - 3.1 If $F(p) \cap F_M = \{v\}$, add to F_L the vertex v_3 and the edges $\frac{v_1+v_3}{2}$ and $\frac{v_2+v_3}{2}$.
 - 3.2 If $F(p) \cap F_M = \{v, v_1\}$, add to F_L the edge $\frac{v_2+v_3}{2}$.
4. If $p \in \mathcal{E}_3$, let $v_i, i = 1 \dots 8$ denote the vertices in $C(p)$. Then:

4.1 For each vertex v_i in $C(p)$, do:

If $v_i \notin F_M$, add v_i to F_L .

4.2 For each pair of vertices $\{v_i, v_j\} \subset C(p)$, do:

If $\{v_i, v_j\} \cap F_M = \emptyset$ and $\frac{v_i+v_j}{2} \in \mathcal{E}_1$, add $\frac{v_i+v_j}{2}$ to F_L .

4.3 For each set of vertices $\{v_i, v_j, v_k, v_\ell\} \subset C(p)$, do:

If $\{v_i, v_j, v_k, v_\ell\} \cap F_M = \emptyset$ and the point $\frac{v_i+v_j+v_k+v_\ell}{4} \in \mathcal{E}_2$, add $\frac{v_i+v_j+v_k+v_\ell}{4}$ to F_L .

Suppose I has n points. Then, F_M has at most $27n$ points and F_L at most $(1 + 6 + 12 + 8)27n$ (each polyhedron has at most 6 faces, 12 edges and 8 vertices). For analyzing time complexity, first observe that there are at most $27n$ points in F_M . For each of them, we have to add points to F_M and evaluate at most (Step 4) $(8 + \binom{8}{2} + \binom{8}{4})$ if-conditions, that we assume we can do in constant time. Then, worst-time complexity is $O(107n)$.

The following one is the main result of the paper, which demonstrates that just storing one point per polyhedron of $P(I)$ is enough to recover geometric and topological information of $P(I)$.

Theorem 2 Having a minimal encoding $M = (\mathbb{Z}^3, F_M)$ of the well-composed polyhedral complex $P(I)$ over a picture I as input of Procedure 4, the output is the 3D binary image $L = (\mathbb{Z}^3, F_L)$ that encodes $P(I)$.

The following statement is a direct consequence of Proposition 3.

Proposition 4 The points of a minimal encoding $M = (\mathbb{Z}^3, F_M)$ of $P(I)$ lie in $2\mathbb{Z}^3$, that is, $F_M \subset 2\mathbb{Z}^3$.

Consequently, we could divide by 2 the coordinates of the points in F_M before storing the 3D binary image M . Besides, sparse matrix compression techniques could also be used.

4 Computing the Set of 2-Cells in $\partial P(I)$ from the Minimal Encoding of $P(I)$

In this section, we provide an algorithm to compute a 3D binary image encoding the boundary surface of the well-composed polyhedral complex $P(I)$ over a picture I . Recall that $\partial P(I)$ is composed of the 2-cells of $P(I)$ which are faces of exactly one 3-cell in $P(I)$, together with all its faces.

First, in the following proposition we give a very simple condition to detect if a 2-cell of $P(I)$ belongs to $\partial P(I)$ or not.

Proposition 5 Let $P(I)$ be the well-composed polyhedral complex $P(I)$ over a picture I encoded by the 3D binary image $L = (\mathbb{Z}^3, F_L)$. Then, a 2-cell $p \in F_L$ belongs to

$\partial P(I)$ if and only if p satisfies one of the following “boundary conditions”:

- (1) $p = (4i + 2, 4j, 4k) \in \mathcal{E}_2$, $p + d \notin F_L$ and $p + 2d \notin F_L$ for some $d \in \{(\pm 1, 0, 0)\}$.
Analogous conditions apply for $p = (4i, 4j + 2, 4k)$ (with $d \in \{(0, \pm 1, 0)\}$) and $p = (4i, 4j, 4k + 2)$ (with $d \in \{(0, 0, \pm 1)\}$).
- (2) $p = (2i + 1, 2j, 2k) \in \mathcal{O}_2$ and $p + d \notin F_L$ for some $d \in \{(\pm 1, 0, 0)\}$.
Analogous conditions apply for $p = (2i, 2j + 1, 2k)$ (with $d \in \{(0, \pm 1, 0)\}$) and $p = (2i, 2j, 2k + 1)$ (with $d \in \{(0, 0, \pm 1)\}$).

Procedure 5 computes a 3D binary image $H = (\mathbb{Z}^3, F_H)$ encoding the 2-cells in $\partial P(I)$ directly from the minimal encoding of $P(I)$. Let $D = \{(\pm 1, 0, 0), (0, \pm 1, 0), (0, 0, \pm 1)\}$.

Procedure 5 Computing the 3D binary image $H = (\mathbb{Z}^3, F_H)$ encoding the 2-cells of $\partial P(I)$.

INPUT: The minimal encoding $M = (\mathbb{Z}^3, F_M)$ of the well-composed polyhedral complex $P(I)$ over a picture I .

OUTPUT: A 3D binary image $H = (\mathbb{Z}^3, F_H)$.

For each point $p \in F_M$, do:

If $p \in \mathcal{E}_3$, $p + 2d \notin F_M$ and $p + 4d \notin F_M$ for some $d \in D$, add $p + 2d$ to F_H .

If $p \in \mathcal{E}_2$ then:

If $p = (4i + 2, 4j, 4k)$ and $p + 2d \notin F_M$ for some $d \in \{(\pm 1, 0, 0)\}$, add $p + d$ to F_H .

If $p = (4i, 4j + 2, 4k)$ and $p + 2d \notin F_M$ for some $d \in \{(0, \pm 1, 0)\}$, add $p + d$ to F_H .

If $p = (4i, 4j, 4k + 2)$ and $p + 2d \notin F_M$ for some $d \in \{(0, 0, \pm 1)\}$, add $p + d$ to F_H .

If $p \in \mathcal{E}_1$ then:

If $p = (4i + 2, 4j + 2, 4k)$ and $p + 2d \notin F_M$ for some $d \in \{(\pm 1, 0, 0), (0, \pm 1, 0)\}$, add $p + d$ to F_H .

If $p = (4i + 2, 4j, 4k + 2)$ and $p + 2d \notin F_M$, for some $d \in \{(\pm 1, 0, 0), (0, 0, \pm 1)\}$, add $p + d$ to F_H .

If $p = (4i, 4j + 2, 4k + 2)$ and $p + 2d \notin F_M$, for some $d \in \{(0, \pm 1, 0), (0, 0, \pm 1)\}$, add $p + d$ to F_H .

If $p \in \mathcal{E}_0$ and $p + 2d \notin F_M$, for some $d \in D$, add $p + d$ to F_H .

Suppose I has n points then F_M has at most $27n$ points and F_H at most $6 \cdot 27n$ points since each polyhedron has at most 6 faces. For analyzing time complexity, observe that for

each point in F_M we have to evaluate 6 if-conditions at most. Then, worst-time complexity is $O(6 \cdot 27n)$.

Proposition 6 The result of applying Procedure 5 to the minimal encoding $M = (\mathbb{Z}^3, F_M)$ of the well-composed polyhedral complex $P(I)$ over a picture I is a 3D binary image $H = (\mathbb{Z}^3, F_H)$ that encodes the 2-cells of $\partial P(I)$.

Proof If $p \in \mathcal{E}_3$, we add to F_H all the size-4 square faces $p + 2d$, of a voxel p in $Q(I)$, that are not critical cells (since $p + 2d \notin F_M$) and are in the boundary since $p + 2d$ satisfies the first boundary condition in Proposition 5.

If $p \in \mathcal{E}_2$, p is a polyhedron of type $p_\ell(\dots)$. Let $i, j, k \in \mathbb{Z}$. Suppose, without loss of generality, that $p = (4i + 2, 4j, 4k)$. Then, both quadrangular faces that might be in $\partial P(I)$ are $p + d \in \mathcal{O}_2$, with $d \in \{(\pm 1, 0, 0)\}$. The one satisfying the second boundary condition in Proposition 5 is added to F_H .

If $p \in \mathcal{E}_1$, p is either a pyramid or a size-2 cube. Suppose, without loss of generality, that $p = (4i + 2, 4j + 2, 4k)$. Either the triangular faces of the pyramid or the square faces of the cube that are parallel to the edge given by $E(p)$, are those $p + d \in \mathcal{O}_2$ that are added to F_H when they satisfy the second boundary condition in Proposition 5. Notice that in the case of the size-2 cube, the other two square faces cannot lie in $\partial P(I)$.

If $p \in \mathcal{E}_0$, p is a size-2 cube and a square face $p + d$ is added whenever $p + d \in \mathcal{O}_2$ satisfies the second boundary condition in Proposition 5. □

5 Conclusion and Future Work

In this paper, we have made important progress on the computational treatment of the codification system developed in [5] to store and deal with well-composed polyhedral complexes over pictures. These complexes may have applications in the context of 3D visualization as well as 3D printing due to the development of a mathematical 3D model of objects whose boundary surface is a manifold while the storage system allows a quick access to the cells and boundary relations between them. These applications constitute a possible research line in future. Besides, it could be interesting to explore the extension of the construction of well-composed polyhedral complexes over pictures to higher dimensions. Finally, the development of a homology computation algorithm adapted to these type of complexes is also considered as future work, such as, for example, the direct computation of the homology of $P(I)$ via its minimal encoding or via the encoding of the 2-cells of its boundary surface.

Acknowledgements We would like to thank the anonymous reviewers for their valuable suggestions and comments.

Appendix: Proofs of the Main Results in the Paper

Proof of Proposition 1 Let us prove the statements case by case.

- (a) A size-2 cube $c(p)$ is centered at $p \in \mathcal{E}_0$ has its faces encoded by their barycenters, all of them in $N^{\leq 1}(p) \subseteq F_L$. See Table 3b.2. Conversely, if $N^{\leq 1}(p) \subseteq F_L$ for $p \in \mathcal{E}_0$, then p was a critical vertex in $Q(I)$ and, hence, it encodes a size-2 cube $c(p) \in P(I)$.
- (b) If $p \in \mathcal{E}_1$ then $p = \frac{v+w}{2}$ for $\{v, w\} = E(p) \subset \mathcal{E}_0$.
 - If p is a pyramid $k(v, w)$, then, by construction, v (but not w) is the center of a size-2 cube (see Table 3c.1). Conversely, if $N^{\leq 1}(q) \subseteq F_L$ for exactly one of the two points $q \in E(p)$, then q (and not the other point $q' \in E(p)$) was a critical vertex in $Q(I)$ and, hence, a size-2 cube is centered at q (but not at q') in $P(I)$, what means that p is a pyramid.
 - p is a size-2 cube iff both points in $E(p)$ are size-2 cubes, that is, iff they were critical vertices in $Q(I)$.
- (c) If $p \in \mathcal{E}_2$ is one of the polyhedra $p_\ell(v, v_1, v_2, v_3)$, $\ell = 1, 2, 3, 4$, then $F(p) = \{v, v_1, v_2, v_3\}$ forms a size-4 square face s being p the barycenter of s . Besides, v, v_1, v_2 and/or v_3 (but not all) are the centers of size-2 cubes (see Table 3[d,e,f,g].1). The number and relative positions of the points q in $F(p)$ such that $N^{\leq 1}(q) \subseteq F_L$ give rise to the different polyhedra $p_\ell(v, v_1, v_2, v_3)$, $\ell = 1, 2, 3, 4$. Conversely, if some points q in $F(p)$ satisfy that $N^{\leq 1}(q) \subseteq F_L$, they corresponded to critical vertices in $Q(I)$ what leads to the construction of the polyhedra $p_\ell(v, v_1, v_2, v_3)$, $\ell = 1, 2, 3, 4$. Finally, p is a size-2 cube iff all the points in $F(p)$ are size-2 cubes, that is, iff they were critical vertices in $Q(I)$.
- (d) The hexahedra $h_\ell(v_1, \dots, v_8)$, $\ell = 1, \dots, 22$ (where vertices $v_1, \dots, v_8 \in \mathcal{E}_0$ form a voxel c), is encoded by $p = \frac{v_1+\dots+v_8}{8}$, (i.e., the barycenter of c). So, if p is one of those hexahedra, then, necessarily, $\{v_1, \dots, v_8\} = C(p)$. Besides:
 - p is one of the 20 hexahedra listed in the first column of Tables 4 and 5 if a subset of at least one and no more than seven points $W \subset C(p)$ are the center of a size-2 cube. Then, for any $q \in W$, $N^{\leq 1}(q) \subseteq F_L$;
 - otherwise, p is a size-4 cube if $W = \emptyset$ and it is a size-2 cube if $W = C(p)$.

Conversely, if there is a subset of points $W \subseteq C(p)$ satisfying $N^{\leq 1}(q) \subseteq F_L$, they corresponded to critical vertices in $Q(I)$, what leads to the construction of each of the 22 hexahedra.

Proof of Proposition 2 We run over all the polyhedra and check that their faces are all encoded by either points in \mathcal{O}_ℓ

or in \mathcal{E}_ℓ . In fact, we will provide a description of their coordinates.

- The size-2 cube centered at a point $p \in 2\mathbb{Z}^3$ has square faces in $N_6^1(p) \subset \mathcal{O}_2$, edges in $N_{12}^1(p) \subset \mathcal{O}_1$ and vertices in $N_8^1(p) \subset \mathcal{O}_0$.
- The size-4 cube centered at a point $p \in 4\mathbb{Z}^3$ has square faces in $N_6^2(p) \subset \mathcal{E}_2$, edges in $N_{12}^2(p) \subset \mathcal{E}_1$ and vertices in $N_8^2(p) \subset \mathcal{E}_0$.
- The four triangle faces of the pyramid $p = (4i, 4j + 2, 4k + 2)$ are $\{(4i, 4j + 2 \pm 1, 4k + 2), (4i, 4j + 2, 4k + 2 \pm 1)\} \subset \mathcal{O}_2$ (see green voxels in Table 3c.2). If $v = (4i - 2, 4j + 2, 4k + 2)$ is the point in $E(p) \subset \mathcal{E}_0$ for which $N^{\leq 1}(v) \subset F_L$, then $w = (4i + 2, 4j + 2, 4k + 2)$ is the apex of the pyramid. The four edges of p incident to w are $\{(4i, 4j + 2 \pm 1, 4k + 2 \pm 1)\} \subset \mathcal{O}_1$. Cases $p = (4i + 2, 4j, 4k + 2)$ and $p = (4i + 2, 4j + 2, 4k)$ are analogous.
- The polyhedron $p_\ell(v, v_1, v_2, v_3)$, $\ell = 1, 2, 3, 4$, encoded by $p = (4i, 4j, 4k + 2)$, has two quadrangle faces $\{(4i, 4j, 4k + 2 \pm 1)\} \subset \mathcal{O}_2$.
 - For $p_1(v, v_1, v_2, v_3)$, if $v = (4i - 2, 4j - 2, 4k + 2)$ is the vertex in $F(p) \subset \mathcal{E}_0$ for which $N^{\leq 1}(v) \subset F_L$, then the vertex v_3 (see Table 3d.1) is $(4i + 2, 4j + 2, 4k + 2)$ and the size-4 edges $(4i, 4j + 2, 4k + 2)$ and $(4i + 2, 4j, 4k + 2)$.
 - For $p_2(v, v_1, v_2, v_3)$, if $v = (4i - 2, 4j - 2, 4k + 2)$ and $v_1 = (4i - 2, 4j + 2, 4k + 2)$ are the vertices in $F(p) \subset \mathcal{E}_0$ for which $N^{\leq 1}(v) \subset F_L$, then the size-4 edge (see Table 3e.1) is $(4i + 2, 4j, 4k + 2)$.

Cases in which $p_\ell(v, v_1, v_2, v_3)$ is encoded by $p = (4i + 2, 4j, 4k)$ and $p = (4i, 4j + 2, 4k)$ are analogous.

- For each of the 20 hexahedra $h_\ell(v_1, \dots, v_8)$ shown in Tables 4 and 5, encoded by $p = (4i, 4j, 4k)$ the vertices w_1, \dots, w_8 are $\{(4i \pm 1, 4j \pm 1, 4k \pm 1)\} \subset \mathcal{O}_0$, such that, for example, if $v_\ell = (4i + 2, 4j - 2, 4k + 2)$ then $w_\ell = (4i + 1, 4j - 1, 4k + 1)$. A size-4 square face of $h_\ell(v_1, \dots, v_8)$ is encoded by a point $q \in \mathcal{E}_2 \cap N_6^2(p)$ with $N^{\leq 1}(q) \cap F_L = \{q\}$, a size-4 edge is encoded by a point $q \in \mathcal{E}_1 \cap N_{12}^2(p)$ with $N^{\leq 1}(q) \cap F_L = \{q\}$ and a vertex not shared with a pyramid or a size-2 cube is encoded by a point $q \in C(p)$ with $N^{\leq 1}(q) \cap F_L = \{q\}$. For example, in the second case (second row) in Table 4, if $v_1 = (4i - 2, 4j - 2, 4k - 2)$ and $v_2 = (4i - 2, 4j + 2, 4k - 2)$, are the vertices in $C(p) \subset \mathcal{E}_0$ for which $N^{\leq 1}(v_i) \subset F_L$, then the two size-4 square faces are $(4i + 2, 4j, 4k)$ and $(4i, 4j, 4k + 2)$; the size-4 edges are $(4i + 2, 4j \pm 2, 4k), (4i + 2, 4j, 4k \pm 2), (4i - 2, 4j, 4k + 2), (4i, 4j \pm 2, 4k + 2)$; and the vertices not shared with a pyramid or a size-2 cube are $(4i + 2, 4j \pm 2, 4k + 2)$.

Notice that, for each polyhedron different to the cubes, some of their faces have not been explicitly given since they have been described before as part of another polyhedron. \square

Proof of Theorem 1 Proposition 1 guarantees that all the polyhedra and their faces can be univocally identified by the type of coordinates and neighboring information in D_P . Besides, Procedure 2 provides a method to retrieve the $(\ell-1)$ -faces of a given ℓ -cell, so the binary image $L = (\mathbb{Z}^3, F_L)$ can properly encode the well-composed polyhedral complex $P(I)$ over a picture I . \square

Proof of Proposition 3 First, notice that the color operation accomplished in Procedure 1, to obtain the ECM representation of $P(I)$, sets $g_P(p) = 3$ whenever p is a critical cell (i.e., a cell incident to a critical vertex). Given a critical vertex $p \in F_J \cap \mathcal{E}_0$, incident edges would lie in $\mathcal{E}_1 \cap N_6^2(p) \cap F_J$ and incident square faces in $\mathcal{E}_2 \cap N_{12}^2(p) \cap F_J$. Geometrically, each critical cell is replaced by a specific polyhedron within the ones listed in Definition 2. This way, any polyhedron of $P(I)$ is encoded by: (1) a point $p \in \mathcal{E}_3$; or (2) a point $p \in \mathcal{E}_\ell, 0 \leq \ell \leq 2$, if p encodes a critical ℓ -cell of $Q(I)$. \square

Proof of Theorem 2 By definition, the points $p \in F_M$ are encoding all the polyhedra in $P(I)$. We have to prove that Procedure 4 can retrieve all the points in F_L encoding faces of such polyhedron.

- If $p \in F_M \cap \mathcal{E}_0$, then p is a size-2 cube and its faces are encoded in $N^{\leq 1}(p) = S_0(p)$, which have been added to F_L in Step 1.
- If $p \in F_M \cap \mathcal{E}_1$, let v, w denote the two points in $E(p)$. Then,
 - If $E(p) \cap F_M = \{v\}$, then p is the pyramid $k(v, w)$ shown in Table 3c.1. (first case of Proposition 1.b), the block $S_1(p)$ added contains the triangular faces of the pyramid and the vertex w is added in Step 2.1. The square face and its faces are added together with the block $S_0(v)$ (Step 1).
 - If $E(p) \cap F_M = E(p)$, then p is a size-2 cube (second case of Proposition 1.b) and points in $S_1(p)$ encode the faces that are parallel to the edge formed by the points in $E(p)$. The rest of the faces are encoded as part of the corresponding blocks $S_0(v)$ and $S_0(w)$.
- If $p \in F_M \cap \mathcal{E}_2$, let v, v_1, v_2, v_3 denote the four points in $F(p)$. Then, the quadrangular faces are added with the blocks $S_2(p)$ in Step 1.
 - If $F(p) \cap F_M = \{v\}$, then p is the polyhedron $p_1(v, v_1, v_2, v_3)$ shown in Table 3d.1, according to the first case of Proposition 1.c. Then in Step 3.1, the vertex v_3 is encoded, as well as the edge between vertices v_1 and v_3 and the one between v_2 and v_3 ,

which are the only faces that had not been encoded in previous steps.

- If $F(p) \cap F_M = \{v, v_1\}$, then p is the polyhedron $p_2(v, v_1, v_2, v_3)$ (shown in Table 3e.1) and in Step 3.2, the edge between v_2 and v_3 is added (the rest of faces had been added in previous steps).
- If $F(p) \cap F_M = \{v, v_3\}$, then p is the polyhedron $p_3(v, v_1, v_2, v_3)$ (shown in Table 3f.1) and its faces have already been added.
- If $F(p) \cap F_M = \{v, v_1, v_3\}$, then p is the polyhedron $p_4(v, v_1, v_2, v_3)$ shown in Table 3g.1, corresponding to the fourth case of Proposition 1.c and its faces have already been added.
- If $p \in F_M \cap \mathcal{E}_3$, let $C(p) = \{v_1, \dots, v_8\}$. Then p is one of the 22 hexahedra given in Definition 2, that is p is either one of the 20 hexahedra listed in the first column of Tables 4 and 5, or a size-4 cube or a size-2 cube, depending on the number and relative positions of the critical vertices in $C(p)$. In each case, Step 4.1 adds the vertices of the hexahedra that are not critical vertices (though some of them could have been added before as apex of a pyramid); Step 4.2 adds the possible edges between these vertices; and Step 4.3 adds the possible square faces between them. The rest of the faces are part of other polyhedra and have already been added in previous steps. \square

Proof of Theorem 5 Let $i, j, k \in \mathbb{Z}$. For $p \in F_L$, we consider the two possible cases:

- $p \in \mathcal{E}_2$. Without loss of generality, suppose that $p = (4i + 2, 4j, 4k)$. Then, p is a size-4 square face iff $N^{\leq 1}(p) \cap F_L = \{p\}$ and it will lie on $\partial P(I)$ if $p + (2, 0, 0) = (4i + 4, 4j, 4k) \notin F_L$ and $p - (2, 0, 0) = (4i, 4j, 4k) \in F_L$ or viceversa.
- $p \in \mathcal{O}_2$. The 2-cells that might be in the boundary complex are those in the boundary of the polyhedra in Table 3. There are several cases: (1) p is in the square face of a size-2 cube centered for example in $(4i + 2, 4j + 2, 4k + 2)$, say $p = (4i + 2, 4j + 2, 4k + 1)$; or (2) p is a triangular face of a pyramid $(4i, 4j + 2, 4k + 2)$, say $p = (4i, 4j + 2, 4k + 1)$; or (3) p is one of the quadrangular faces of a polyhedron $q = (4i, 4j, 4k + 2)$, say $p = (4i, 4j, 4k + 1)$. Then, in any of the three cases, $p \in \partial P(I)$ iff $p + d \notin F_L$ for the right direction d . \square

References

1. Artzy, E., Frieder, G., Herman, G.T.: The theory, design, implementation and evaluation of a three-dimensional surface detection algorithm. *Comput. Graph. Image Process.* **15**, 1–24 (1981)

2. Boutry, N., Géraud, T., Najman, L.: How to make nD functions digitally well-composed in a self-dual way. In: Proceedings of the 12th International Conference on Mathematical Morphology and Its Applications to Signal and Image Processing. LNCS 9082, pp. 561–572 (2015)
3. Boutry, N., Géraud, T., Najman, L.: How to make nD images well-composed without interpolation. In: International Conference on Image Processing (ICIP 2015) IEEE, 27–30 Sept 2015, pp. 2149–2153 (2015)
4. Gonzalez-Diaz, R., Jimenez, M.-J., Medrano, B.: Well-composed cell complexes. In: Proceedings of the 16th International Conference on Discrete Geometry for Computer Imagery (DGCI 2011), LNCS 6607, pp. 153–162 (2011)
5. Gonzalez-Diaz, R., Jimenez, M.-J., Medrano, B.: 3D well-composed polyhedral complexes. *Discret. Appl. Math.* **183**, 59–77 (2015)
6. Gonzalez-Diaz, R., Jimenez, M.-J., Medrano, B.: Demo: Well-composed polyhedral complexes. http://grupo.us.es/cimagroup/DEMO_JMIV_matlab.zip
7. Gonzalez-Diaz, R., Jimenez, M.-J., Medrano, B.: Encoding specific 3D polyhedral complexes using 3D binary images. In: Proceedings of the 19th International Conference on Discrete Geometry for Computer Imagery (DGCI 2016). LNCS 9647, pp. 268–281 (2016)
8. Han, X., Xu, C., Prince, J.L.: A topology preserving deformable model using level sets. In: IEEE CVPR2001 (session 2), pp. 765–770 (2001)
9. Hatcher, A.: *Algebraic Topology*. Cambridge University Press, Cambridge (2002). ISBN 0-521-79540-0
10. Herman, G.T.: Discrete multidimensional Jordan surfaces. *CVGIP Graph. Models Image Process.* **54**, 507–515 (1992)
11. Kenmoki, Y., Imiya, A.: Combinatorial boundary of a 3D lattice point set. *J. Vis. Commun. Image Represent.* **17**, 738–766 (2006)
12. Kozlov, D.: Combinatorial algebraic topology. *Algorithms Comput. Math.* **21**, 16–27 (2008)
13. Lachaud, J.O., Montanvert, A.: Continuous analogs of digital boundaries: a topological approach to iso-surfaces. *Graph. Models* **62**, 129–164 (2000)
14. Latecki, L.J.: 3D well-composed pictures. *Graph. Models Image Process.* **59**(3), 164–172 (1997)
15. Latecki, L.J.: *Discrete Representation of Spatial Objects in Computer Vision*. Kluwer Academic, Dordrecht (1998)
16. Latecki, L., Eckhardt, U., Rosenfeld, A.: Well-composed sets. *Comput. Vis. Image Underst.* **61**(1), 7083 (1995)
17. Latecki, L., Conrad, C., Gross, A.: Preserving topology by a digitization process. *J. Math. Imaging Vis.* **8**(2), 131159 (1998)
18. Lorensen, W.E., Cline, H.E.: Marching cubes: a high resolution 3D surface construction algorithm. *ACM SIGGRAPH* **21**, 163–169 (1987)
19. Najman, L., Géraud, T.: Discrete set-valued continuity and interpolation. In: Proceedings of the 11th International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing. LNCS 7883, pp. 37–48 (2013)
20. Rosenfeld, A., Kong, T.Y., Nakamura, A.: Topology-preserving deformations of two-valued digital pictures. *Graph. Models Image Process.* **60**(1), 2434 (1998)
21. Rosenfeld, A., Kong, T.Y., Wu, A.Y.: Digital surfaces. *CVGIP, Graph. Models Image Process.* **53**, 305–312 (1991)
22. Siqueira, M., Latecki, L.J., Gallier, J.: Making 3D binary digital images well-composed. In: Proceedings of SPIE 5675, Vision Geometry XIII, pp. 150–163 (2005)
23. Siqueira, M., Latecki, L.J., Tustison, N., Gallier, J., Gee, J.: Topological repairing of 3D digital images. *J. Math. Imaging Vis.* **30**, 249–274 (2008)
24. Stelldinger, P., Latecki, L.J.: 3D object digitization: majority interpolation and marching cubes. In: Proceedings of the 18th IEEE

International Conference on Pattern Recognition (ICPR 2016), pp. 1173–1176 (2006)

25. Stelldinger, P.: *Image digitization and its influence on shape properties in finite dimensions*, vol. 312. IOS Press, Amsterdam, The Netherlands (2008)



Rocio Gonzalez-Diaz is an Associate Professor in the Department of Applied Mathematics (I) at the University of Seville. She is the head of the Combinatorial Image Analysis group since 2010. Principal Researcher of the Spanish project Computational Algebraic Topology applied to Computer Vision MTM2015-67072-P (MINECO-FEDER) and of a previous related project. She was team member of the European Science Foundation Program: Applied and Computational Topology (2011–2015). Her main interest is to investigate tools (methods) for combinatorial image analysis that build on the theory of (persistent-)homology and cohomology.



Maria-Jose Jimenez is Associate Professor in the Department of Applied Mathematics (I) at the University of Seville. She is team member of the Combinatorial Image Analysis group, the Spanish project Computational Algebraic Topology applied to Computer Vision MTM2015-67072-P (MINECO-FEDER) and was a member of the European Science Foundation Networking Research: Applied and Computational Topology (2011–2015). Her main interests are among computational and combinatorial topology applications to different image contexts, as for example, for action recognition, image sequence analysis, geometrical and topological models for 3D images.



Belen Medrano defended her Ph.D. Thesis in 2007 about Algebraic-Topological models for nD Digital Images. She is Associate Professor in the Department of Applied Mathematics (I) at the University of Seville. She is team member of the Combinatorial Image Analysis group and the Spanish project Computational Algebraic Topology applied to Computer Vision MTM2015-67072-P (MINECO-FEDER). Her main interests are among applications of algebraic topology to the analysis of digital images.