Extracting cell complexes from digital images

Ana Maria Pacheco Martinez, Pascal Lienhardt, Pedro Real Juarez

Extracting cell complexes from digital images

A. Pacheco[1], P. Lienhardt[2] and P. Real[3]

[1]Dpto. Matemáticas e Ingeniería, Universidad Loyola Andalucía, Sevilla, Spain
ampacheco@uloyola.es
[2]Département XLIM-SIC, Université de Poitiers, Poitiers, France
pascal.lienhardt@univ-poitiers.fr
[3]Dpto. Matemática Aplicada I, Universidad de Sevilla, Sevilla, Spain
real@us.es

Abstract: In this paper, we define a method for constructing cell complexes from 4–dimensional binary digital images on a dual grid. First, we revisit a method similar to Kenmochi et al. method [6, 7, 8] for treating with images of dimension 3. Then, we extend this method to 4–dimensional images. The idea consists in considering the black 4–xels of the image as 0–cells of a cell complex. The cells of higher dimension of the complex are constructed by deforming the 4–cubes of the dual grid. Finally, the resulting complex can be simplified, for instance, by merging adjacent 4–cells which share a common 3–cell. More concretely, 0,1,2,3–cells non-incident to 4–cells are stored, together with 3–cells (and their boundary) incident to exactly one 4–cell.

Keywords: cell complex, dual binary digital image, convex hull, degeneracy operator.

1 Introduction

A *n–dimensional digital image* can be defined as a set of n–xels on a grid made up by n–cubes. The n–xels can be identified with (1) the n–cubes of the grid; or with (2) the central points of these n–cubes. In the first case, we work with a *primal grid*; whereas in the second one, we work with a *dual grid* constructed from the primal one.

Segmentation consists in computing a partition of an image into regions. Every n–xel is assigned a label and each region is made up by n–xels with the same label. If the only labels allowed for the n–xels are “white” and “black”, the segmentation is said *binary*.

Methods as *Marching cubes* [4, 10, 11] and *Kenmochi et al.* [6, 7, 8] construct complexes whose cells are predefined, i.e. they are regular cells as simplices, polyhedra, etc. These complexes represent the topology of the region of interest of a 3–dimensional binary digital image. In the first method, the algorithm constructs a simplicial complex, whose 0–cells are points of the edges of the dual grid. In the second one, the authors construct a cell complex on a dual grid, i.e. the 0–cells of the complex are vertices of the dual grid. In order to construct the complex, Kenmochi et al. compute (up to rotations) the different configurations of white and black vertices of a cube and, then, they construct the convex hulls of the black points of these configurations. These convex hulls define the cells of the complex, up to rotations.

The work developed in this paper extends Kenmochi et al. method to dimension 4. The goal is to construct a cell complex from a binary digital image defined on a dual grid. First, we determine (up to isometries) the subsets of points which can be constructed from the vertices of a 4–cube, i.e. the *pattern subsets*, and we store them in a look-up table. Later, we construct the convex hulls defined by the pattern subsets, i.e. the *pattern cells*, and we store them in another look-up table. The construction of a pattern cell is made by deforming the unit cube which contains the subset associated to it. This construction method represents to a novelty respect to Kenmochi et al. method since it can be generalized and extended to any dimension. Then, we determine the pattern subsets of the image and we identify each of these subsets with its pattern cell. Latter define (up to isometries) the cells of the cell complex. So, by inverting the isometries, we obtain the cells of the complex. Finally, the cell complex is simplified by extracting its boundary.

The construction of the cell complex from the pattern subsets and pattern cells can be massively parallelised. More concretely, each 4–cube of the grid is scanned, the pattern subset of that 4–cube is determined and it is identified with its pattern cell.

This paper is structured as follows: in Section 2, we define the procedure above-described for constructing cell complexes from dual binary digital images; in Section 3, (a) we describe Kenmochi et al. method for constructing cell complexes from 3–dimensional dual binary digital images, (b) we implement algorithms for determining both the pattern subsets and the pattern cells. More concretely, once the pattern subsets are determined, we compute the pattern cell associated to each subset by deforming the unit cube which contains it. In this way, we obtain an alternative to Kenmochi et al. method, and (c) we compare the results obtained by using both methods; in Section 4, we extend the algorithms implemented in Section 3(b) for determining both the pattern subsets and the pattern cells in dimension 4, obtaining a method for constructing cell complexes from 4–dimensional dual binary digital images; and finally, in Section 5, we show some conclusions about the method.

2 Method description

Given a n–dimensional binary digital image I on a dual grid G, in this section, we describe a method for constructing a cell complex $CC(I)$ from I. The method is divided into two stages.

2.1 Pattern subsets and pattern cells

The first stage is a preprocessing made only once for each dimension. Let C be a n–cube of G. The non-isometric subsets which can be constructed from the vertices of C are computed. In this way, we obtain the *pattern subsets* which are stored in a look-up table.

Each pattern subset defines a *pattern cell*. This cell is an open set made up by all the points inside the convex hull of the points of the subset which defines the cell. The boundary of this convex hull is given by its vertices, edges, faces, etc. These convex hulls are constructed by deforming C and we distinguish several basic construction operations (deformation, degeneracy of cells, etc.). The pattern cells (together with their boundary) are stored in a look-up table.

2.2 Cell complex construction

The second stage consists of six general steps (non-depending on the dimension) which allow us to construct $CC(I)$ from the pattern subsets and pattern cells. Below, we describe each of these steps.

2.2.1 Subsets of n–xels of I

We split the set of n–xels of I in such a way that each subset of n–xels coincides with the vertices of a n–cube of G. More concretely, we scan each n–cube of G and we check which is the subset of n–xels of I corresponding to its vertex set. A naive algorithm, as Algorithm A.1 (in Appendix), allows us to implement this step.

2.2.2 Pattern subsets of I

We associate each subset of n–xels of I with one of the pattern subsets determined in the first stage. More precisely, we search the isometry between every subset of n–xels of I and the corresponding pattern subset. A naive algorithm, as Algorithm A.2 (in Appendix), allows us to implement this step.

2.2.3 Pattern cells of $CC(I)$

We choose the pattern cells defined by the pattern subsets of I in the look-up table in the first stage.

2.2.4 Cells of $CC(I)$

The cells of $CC(I)$ are obtained by inverting the isometry σ between each subset X of n–xels of I and its associated pattern subset Y. Let $C(Y)$ be the pattern cell corresponding to the pattern subset Y. If we replace every point of $C(Y)$ with its pre-image under σ, then we obtain the cell $C(X)$ corresponding to the subset X.

2.2.5 Construction of $CC(I)$

$CC(I)$ is formed by attaching the cells obtained in Section 2.2.4 along their boundaries.

2.2.6 Simplifying $CC(I)$

$CC(I)$ can be simplified by removing the $(n-1)$–cells incident to two n–cells. More concretely, the *simplified cell complex* is given by: (a) the 0,1,...,$(n-1)$–cells of $CC(I)$ which are not incident to a n–cell; and (b) the set of $(n-1)$–cells (together with their boundary) incident to exactly one of the n–cells of $CC(I)$. A naive algorithm, as Algorithm A.3 (in Appendix), allows us to obtain the simplified cell complex from $CC(I)$.

In Figure 1, we present a diagram summarizing the steps above-described for constructing a cell complex from a n–dimensional dual binary digital image.

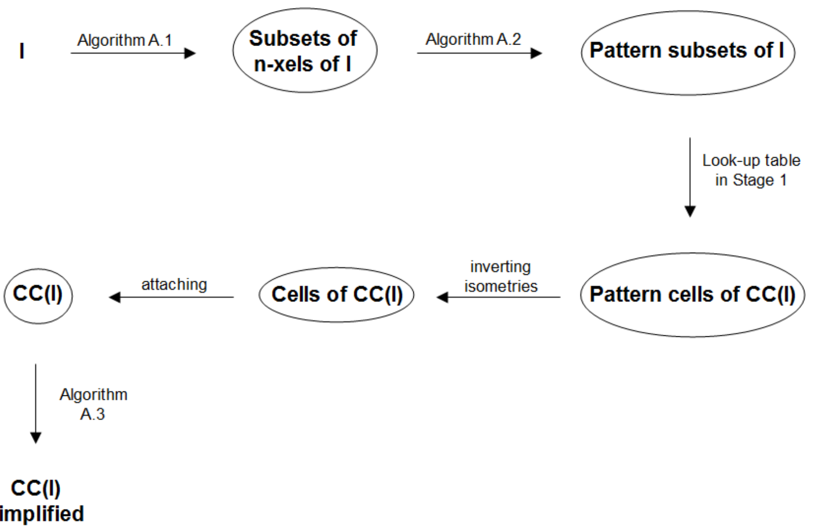


Figure 1: The diagram shows the procedure which allows us to construct the cell complex from the image.

In Examples 1 and 2 (see Appendix), we show the results obtained by applying this method to two dual binary digital images of dimension 3 and 4, respectively.

3 Extracting cell complexes from 3–dimensional digital images

In this section, (1) we recall Kenmochi et al. method (see [6, 7, 8]) for constructing cell complexes from 3–dimensional binary digital images on a dual grid; (2) we construct the look-up tables with the pattern subsets and pattern cells (respectively) in dimension 3, obtaining an alternative to Kenmochi et al. method; and finally, (3) we compare both methods.

3.1 Kenmochi et al. method

Kenmochi et al. define 3–dimensional digital images as subsets of a dual grid (see, for instance, [6, 7, 8]). The voxels of these images are the vertices of the cubes of the dual grid. They work with binary images, so that the voxels are white or black. Kenmochi et al. construct a *cell complex* from the black voxels of the image, in such a way that the boundary of the complex represents the object. The *cell complex* consists of a collection of *cells* together with the information on how they are attached to each other. The black voxels are the 0–cells of the complex. Every cell is defined by the black vertices of a cube of the grid. More concretely, every cell is an open set including the points inside the convex hull of the vertices which define it. Moreover, the boundary of the cell is given by the vertices, edges and faces of the convex hull. The cells are attached to each other along their boundaries.

There exist $2^8 = 256$ configurations of white and black voxels on a cube. Kenmochi et al. consider that two configurations on a cube which have the same number of black voxels are identical if there exists a rotation which sends the first configuration to the second one. In this way, they reduce the 256 configurations of white and black voxels on a cube to 23 configurations. In Table 1, extracted of [6], these 23 configurations of white and black voxels on a cube are shown.

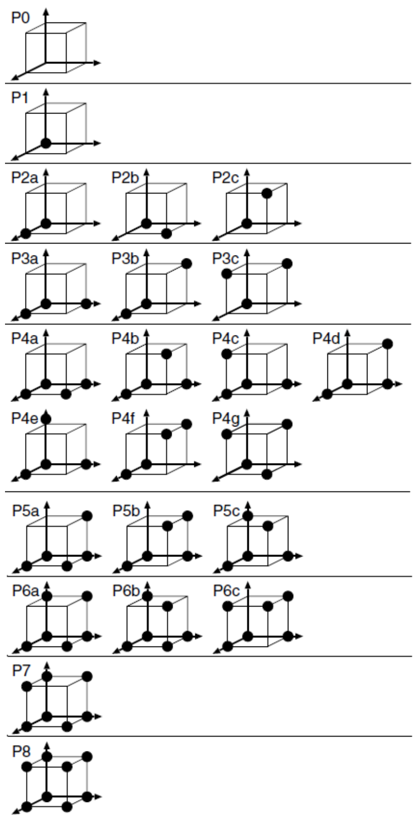


Table 1: Configurations of white and black voxels on a cube obtained by Kenmochi et al.

In their works, Kenmochi et al. state that any 3–dimensional binary digital image is made up by combining (up to rotations) the 23 configurations of white and black voxels shown in Table 1, taking into account that it is not necessary to use all them and any of them can be used more than once. In this way, given a 3–dimensional binary digital image on a dual grid, every configuration of white and black voxels on a cube of the grid coincides (up to rotations) with one of the 23 configurations shown in Table 1.

Next, for each configuration in Table 1, Kenmochi et al. construct a cell defined by the black voxels of the configuration. We recall that each of these cells is an open set including the points inside the convex hull of the vertices which define it and the boundary of the cell is given by the vertices, edges and faces of the convex hull. In this way, Kenmochi et al. obtain the 23 cells shown in Table 2 from the black voxels of the configurations shown in Table 1. These cells are used later for constructing the cell complex whose boundary represents the object encoded by the image.

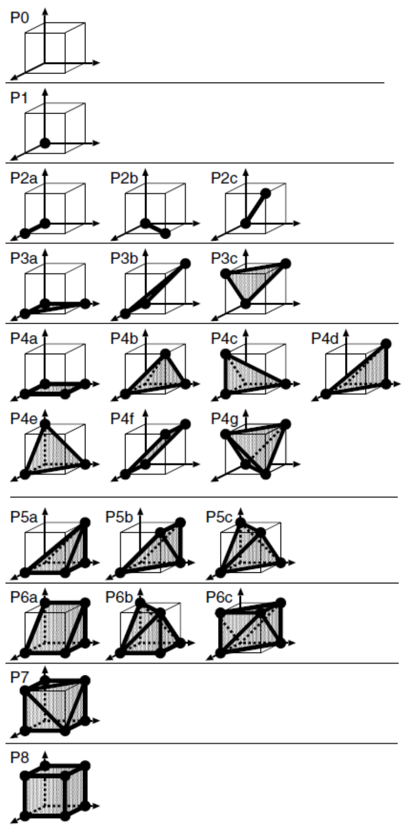


Table 2: The 23 cells obtained by Kenmochi et al. (extracted of [6]) from the black voxels of the configurations shown in Table 1.

In their works, Kenmochi et al. state that the cell complex corresponding to a 3–dimensional binary digital image is made up by combining (up to rotations) the 23 cells shown in Table 2. In this way, every cell of the complex coincides (up to rotations) with one of the 23 cells shown in Table 2.

3.2 Look-up tables of the method described in Section 2 for dimension 3

The method described in Section 2 generalizes Kenmochi et al. method to any dimension, although the cells of the complex are defined up to isometries and not up to rotations. In this section, we construct the look-up tables containing the pattern subsets and the pattern cells in dimension 3.

3.2.1 Pattern subsets

We compute the non-isometric subsets which can be made up from the vertices of a cube. There exist 256 subsets which can be constructed from the vertices of a cube. We determine the isometric subsets and we store a representative of each isometry class. They are the *pattern subsets.*

A naive procedure for computing the isometric subsets which can be made up from the vertices of a cube consists in (1) determining the group of isometries of a cube and (2) applying it to each subset. The group of isometries of the unit cube can be computed with **Automorphisms[Hypercube[3]]** in the *Mathematica* package **Combinatorica'**. This command gives a list with the 48 permutations of vertices which leave the unit cube invariant. Moreover, by permuting the vertices of each subset of the unit cube according to this list, we obtain the subsets isometric to it.

In this way, in order to compute the pattern subsets: (a) we consider an ordered list with the 256 subsets which can be constructed from the vertices of a cube and (b) for each subset of the list, we compute and remove the subsets of the list isometric to it. In the last stage, we obtain a list with the pattern subsets.

Theorem 1 and Table 3 summarize the results.

Theorem 1 *In $\mathbb{Z}^3$, there exist: (a) one pattern subset with zero points, (b) one pattern subset with one point, (c) three pattern subsets with two points, (d) three pattern subsets with three points, (e) six pattern subsets with four points, (f) three pattern subsets with five points, (g) three pattern subsets with six points, (h) one pattern subset with seven points, and (i) one pattern subset with eight points.*

Remark 1 *If we change the order relation, we obtain pattern subsets isometric to those shown in Table 3.*

As main result, we can establish Theorem 2.

Theorem 2 *Any 3–dimensional binary digital image on a dual grid is made up by combining (up to isometries) the 22 pattern subsets shown in Table 3, taking into account that it is not necessary to use all them and any of them can be used more than once.*

Below, we associate each of the 22 pattern subsets with a *pattern cell.* Every pattern cell consists in the convex hull of the points of the corresponding pattern subset. More precisely, every pattern subset is associated with (1) an open cell made up by the points inside the convex hull and (2) a boundary made up by the boundary cells of the convex hull.

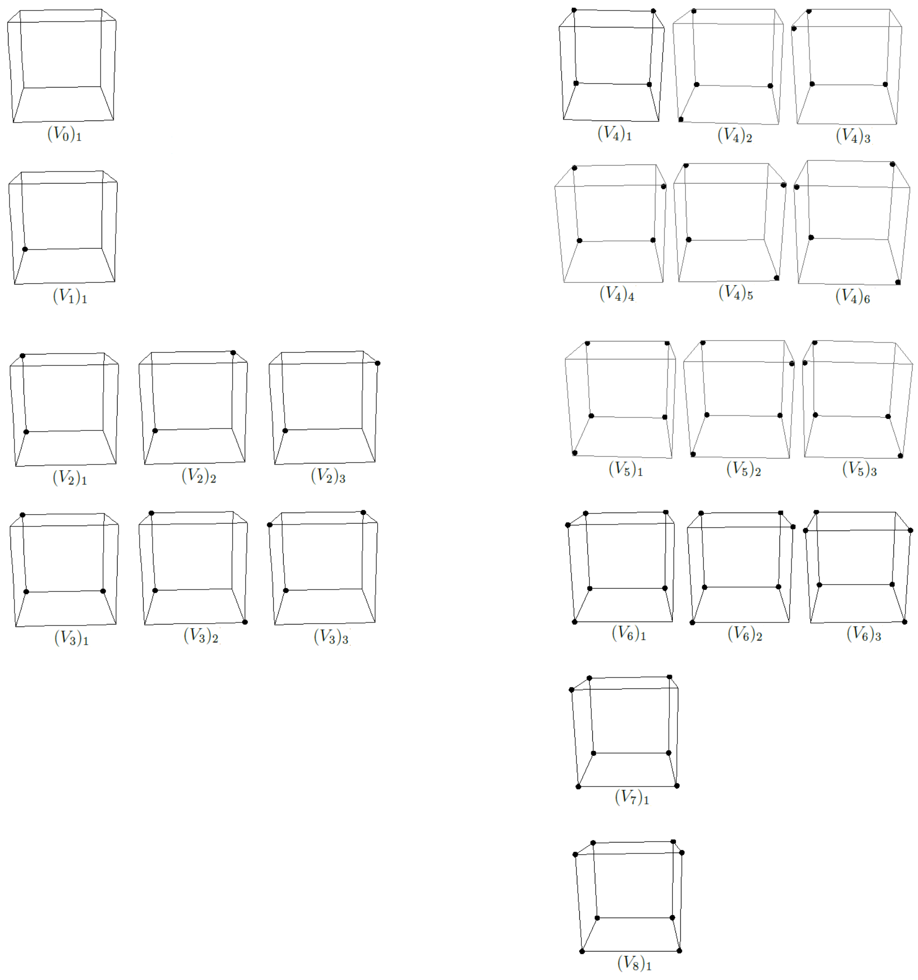


Table 3: Pattern subsets in $\mathbb{Z}^3$ obtained by using the lexicographic order.

3.2.2 Pattern cells

We define a computational method for determining the convex hull of each of the pattern subsets. The technique consists in deforming the cube which contains the subset. Moreover, we deform the faces of the cube which contain vertices non-belonging to the pattern subset, from now on *white vertices* (see Figure 2 (a)). These deformations are a consequence of degenerating edges incident to white vertices. These edge degeneracies can lead to the fact that a face (resp. volume) degenerates into an edge (resp. face) incident to it (see Figure 2 (b) and (c)).

3.2.2.1 Finding the edges to degenerate

We show that by degenerating edges incident to a white vertex which do not belong to the cube, we do not always obtain the convex hull of the pattern subset. Moreover, we justify that if during the deformation of the cube we only

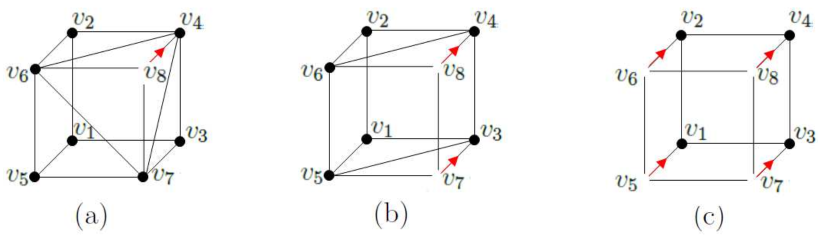


Figure 2: (a) Degeneracy of the edge v_4v_8 into the vertex v_4. This edge degeneracy leads to the fact that the square faces $v_2v_4v_6v_8$ and $v_3v_4v_7v_8$ are deformed into the triangular faces $v_2v_4v_6$ and $v_3v_4v_7$, respectively. Moreover, the square face $v_5v_6v_7v_8$ is geometrically deformed into a non-planar face; it leads to the fact that this face has to be subdivided into two triangular faces $v_4v_6v_7$ and $v_5v_6v_7$. (b) Degeneracies of the edges v_3v_7 and v_4v_8 into the vertices v_3 and v_4, respectively. These edge degeneracies lead to the fact that the square face $v_3v_4v_7v_8$ degenerates into the edge v_3v_4. (c) Degeneracies of the edges v_1v_5, v_2v_6, v_3v_7 and v_4v_8 into the vertices v_1, v_2, v_3 and v_4, respectively. These edge degeneracies lead to the fact that the cube degenerates into the square face $v_1v_2v_3v_4$.

degenerate edges belonging to the cube, then we always obtain the convex hull of the subset.

Degenerating edges non-belonging to the cube

We suppose that we want to compute the convex hull of the subset $\{v_1, v_2, v_3, v_4, v_8\}$ of vertices of the cube (see Figure 3 (a)) and we suppose that at the penultimate step of the deformation we have obtained the polyhedron shown in Figure 3 (b). The last step of the procedure consists in degenerating one of the edges incident to v_5.

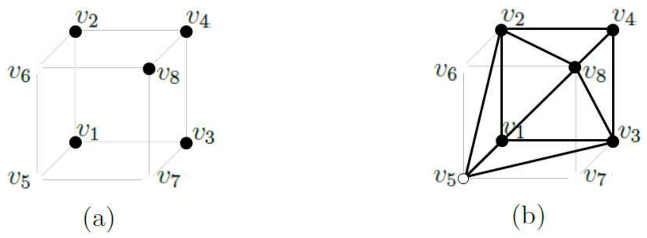
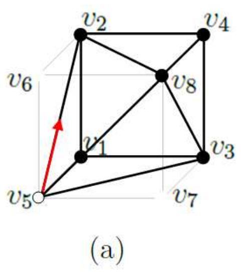


Figure 3: (a) Subset $\{v_1, v_2, v_3, v_4, v_8\}$ of vertices of the cube. (b) Result obtained at the penultimate step of the deformation.

If we degenerate the edge v_2v_5 (see Figure 4 (a)), we do not obtain the convex hull of the subset of black points. Actually, the black points define a volume contained in the cube. However, in Figure 4 (b) we have obtained an object made up by the square face $v_1v_2v_3v_4$ (containing the triangular face $v_1v_2v_3$) and three non-coplanar triangular faces $v_2v_3v_8$, $v_2v_4v_8$ and $v_3v_4v_8$ attached each other by an edge, containing the volume incident to these three faces and a half of the square. It is because to degenerate the edge v_2v_5, the edge v_3v_5 goes through the polyhedron shown in Figure 3 (b), sending it to the volume

$v_2v_3v_4v_8$ together with a face $v_1v_2v_3$ out of the volume.



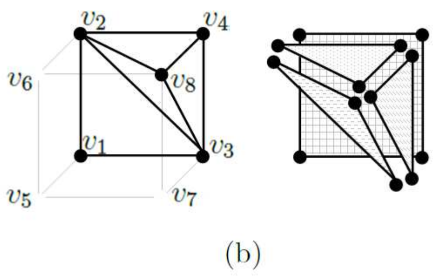


Figure 4: (a) By degenerating the edge v_2v_5, we do not obtain the convex hull of the points in Figure 3 (a). (b) We obtain an object made up by a square face $v_1v_2v_3v_4$ and three non-coplanar triangular faces $v_2v_3v_8$, $v_2v_4v_8$ and $v_3v_4v_8$, containing the volume incident to these three faces and a half of the square.

Let us note that we obtain a similar result by degenerating the edge v_3v_5. On the other hand, if we degenerate the edge v_1v_5 or v_5v_8, we obtain the convex hull of the black points (see Figure 5).

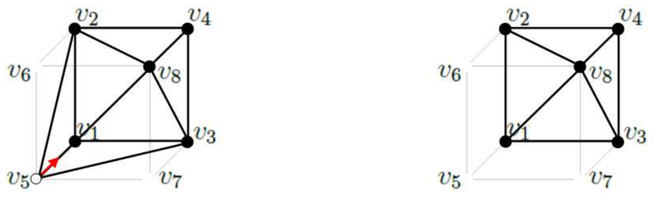


Figure 5: By degenerating the edge v_1v_5, we obtain the convex hull of the points in Figure 3 (a).

Degenerating edges belonging to the cube

We show that by degenerating an edge belonging to the cube, at each step, we obtain the convex hull. Moreover, taking into account that a cube can be decomposed into two prisms with triangular bases (see Figure 6), by symmetries, it suffices to prove that by degenerating an edge of these prisms, at each step, we obtain the convex hull.

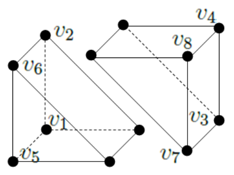


Figure 6: A cube decomposed into two prisms with triangular bases.

At the first step of the deformation of a cube C, we consider the prism D containing the white vertex B. If we degenerate an edge $e \in C \bigcap D$ incident to B, we obtain a pyramid (first two pairs of pictures in Figure 7) or a "pyramid" with a non-planar foursquare face (last four pairs of pictures in Figure 7). Latter appears by deforming the square face of D which contains B and it is not incident to e. We can interpret this deformation as a fold face, so it is necessary to introduce an edge for representing the fold. The non-planar face is replaced by two triangular faces.

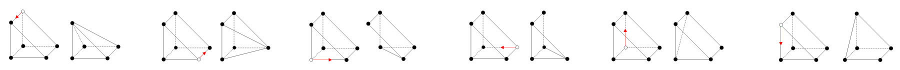


Figure 7: By degenerating any edge of $C \bigcap D$, the prism is deformed into a pyramid.

We suppose, without loss of generality, that at the following step of the deformation, B' is a white vertex of one of the pyramids P obtained in Figure 7 by deforming D. Let us note that B' can be the apex or a base vertex of the pyramid P. Below, we show that, in both cases, if we degenerate an edge $e' \in C \bigcap P$, then we obtain the convex hull of the vertices of $P - \{B'\}$.

- If B' is the apex of P, we degenerate an edge $e' \in C \bigcap P$ incident to the apex. Consequently, P degenerates into its square base and we obtain a square face divided into two coplanar triangular faces sharing an edge. In order to obtain the square face, we remove the edge shared by the two triangular faces (see Figure 8).

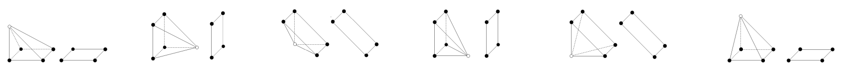


Figure 8: By degenerating any edge of $C \bigcap P$ incident to the apex, the pyramid is degenerated into a square.

- If B' is a vertex of the base of P, we degenerate any edge $e' \in C \bigcap P$ incident to B'. Consequently, P is deformed into a tetrahedron (see first four pairs of pictures in Figure 9) or into a "tetrahedron" with a non-planar foursquare face (see last pair of pictures in Figure 9). In this last case, an edge has to be added in order to keep planar faces.

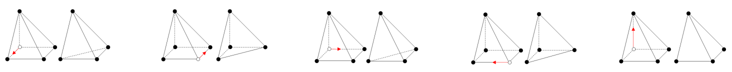


Figure 9: By degenerating any edge of C incident to two vertices of the base of P, we obtain a tetrahedron. By degenerating any edge of C which joins a vertex of the base of P with the apex, we obtain a tetrahedron (except for an edge).

Finally, the degeneracy of a tetrahedron into a triangle, a triangle into an edge, and an edge into a point, respectively, is obtained by degenerating an edge belonging to the cube incident to a white vertex (Figure 10).

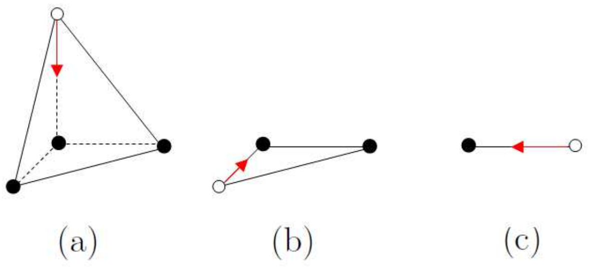


Figure 10: (a) Degeneracy of a tetrahedron into a triangle, (b) a triangle into an edge, and (c) an edge into a point.

Remark 2 *If, at each step, we degenerate an edge belonging to the cube incident to a white vertex, the previous results prove that we always obtain the convex hull.*

Remark 3 *It is also possible to obtain the convex hull by degenerating edges which do not belong to the cube; it happened in Figure 4 with the edge v_5v_8. In this sense:*

- *Figure 7 allows us to note that by degenerating any edge (belonging or not to the cube) of a prism, we obtain a pyramid.*
- *Figure 8 shows that if the apex is the white vertex, then the pyramid is degenerated into its basis by degenerating any edge (belonging or not to the cube) of the pyramid incident to the apex.*
- *If the white vertex is a vertex of the base of the pyramid, then by degenerating any edge (belonging or not to the cube) incident to it and to another base vertex, we obtain a tetrahedron. It is the case of the edge v_5v_8 in Figure 4, which the vertices v_1, v_2, v_4, v_5, v_8 define a pyramid in.*

  Moreover, the only operation which does not allow us to obtain a tetrahedron consists in degenerating an edge of the pyramid which does not belong to the cube and it joins the apex with a white vertex of the base. It is the case of the edge v_2v_5 (resp. v_3v_5) in Figure 4, which the vertices v_1, v_2, v_4, v_5, v_8 (resp. v_1, v_3, v_4, v_5, v_8) define a pyramid in. On the other hand, if v_1 were the white vertex in Figure 3 (b), then by degenerating the edge v_1v_2 (resp. v_1v_3) we would obtain the convex hull (see Figure 11).

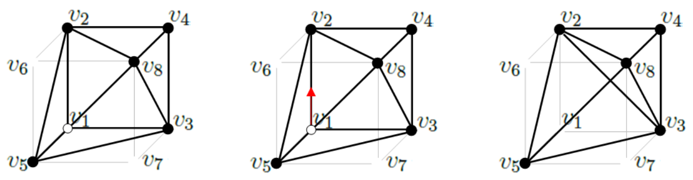


Figure 11: By degenerating the edge v_1v_2, we obtain the convex hull of v_2, v_3, v_4, v_5, v_8.

In the following paragraph, we detail a procedure to assure that at each step of the deformation of the cube, there exists an edge of the cube incident to the white vertex to treat. This edge is degenerated in order to obtain the convex hull.

3.2.2.2 An order relation on the edge degeneracies

We show that the edges of the cube have to degenerate following a certain order; otherwise, in the procedure of deformation of the cube there can be white vertices non-incident to edges of it (see Figure 12), in which case it may not be obtained the convex hull of the points of the pattern subset. Moreover, we describe a procedure for obtaining an order relation for degenerating the edges of the cube. This order relation guarantees that, at each step of the deformation of the cube, the white vertex to treat is incident to an edge of the cube, avoiding situations as that shown in Figure 12.

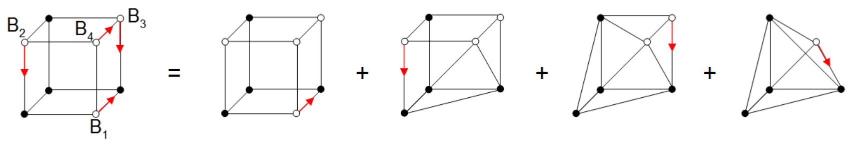


Figure 12: The first picture shows an order for degenerating the edges incident to the white vertices of the cube. The degenerated edges in the second, third, and fourth picture, respectively, lead to the deformation of the three edges of the cube incident to the vertex B_4. There are not any edge of the cube incident to the vertex B_4 for degenerating.

Let $(V_c)_i \subset \mathbb{Z}^3$ be a pattern subset with $0 \leq c \leq 8$ points, contained in a cube $C_{(V_c)_i}$. The procedure for obtaining an order relation on the edge degeneracies consists of two steps.

The first of them is to consider the white vertices of $C_{(V_c)_i}$ and the edges which join them as a subgraph S of $C_{(V_c)_i}$. More concretely, we consider $C_{(V_c)_i}$ as a graph with eight vertices and twelve edges. We delete the points of $(V_c)_i$ from $C_{(V_c)_i}$ and, consequently, the edges incident to these points. In this way, we obtain a subgraph S of $C_{(V_c)_i}$ with $8 - c$ vertices and the remaining edges of $C_{(V_c)_i}$ (an example in Figure 13).

In the second one, for each connected component of S, we construct a rooted spanning tree whose root is a white vertex adjacent to a *black vertex* (a vertex which belongs to $(V_c)_i$) of $C_{(V_c)_i}$. These trees allow us to establish a hierarchy on the set of white vertices of $C_{(V_c)_i}$ (the last vertex is the root). This hierarchy determines an order for degenerating the edges of these trees, i.e. the edges of $C_{(V_c)_i}$ which join white vertices, since the number of edges of a tree with v vertices is $v - 1$. Hence, except for the root, each vertex of a rooted tree is associated with the edge whose end-points are itself and its father. After degenerating all the edges of a tree, we degenerate the edge whose end-points are the root and a black vertex adjacent to it. The idea to avoid situations as that shown in Figure 12 is to choose as root of each tree a white vertex adjacent to a black vertex. In this way, each root is always incident to an edge of $C_{(V_c)_i}$

which has not been deformed.

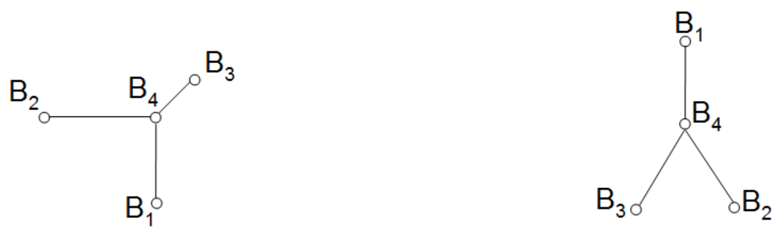


Figure 13: On the left: the subgraph obtained by deleting the black vertices of the cube shown in Figure 12. On the right: its rooted spanning tree.

Remark 4 *In the procedure described previously, we have imposed that the root of each tree is adjacent to a black vertex of the cube. This condition is essential for obtaining the order relation since after degenerating the edges of the tree, it allows us that the root is still incident to an edge of the cube. This edge is the last one in degenerating. Let us observe that this condition is not satisfied in Figure 12, since the last edge to degenerate is incident to the white vertex B_4 which is not adjacent to any black vertex of the cube.*

On the other hand, the procedure assures that the degenerated edges are always edges of the cube. It is because they are edges of a spanning tree of a subgraph of the cube. In this way, we avoid situations as that shown in Figure 12.

3.2.2.3 Convex hull of a pattern subset

We describe a procedure for constructing convex hulls of pattern subsets in $\mathbb{Z}^3$. These convex hulls allow us to determine (up to isometries) the cells (together with their boundary) which can be obtained by deforming a cube. The procedure is based on degenerating edges of the cube incident to white vertices.

As commented at the beginning of Section 3.2.2, the degeneracy of an edge of the cube incident to a white vertex can lead to the degeneracy of faces and/or volumes contained in the cube. Below, we relate the existence of degenerated faces and/or volumes to the *star* of the degenerated edge and to that of the white vertex to treat. We recall that the *star of a cell* c is the set of cells whose boundary contains c.

Let $(V_c)_i$ be a pattern subset contained in the cube $C_{(V_c)_i}$ and S the subgraph of $C_{(V_c)_i}$ constructed by deleting the points of $(V_c)_i$. For each white vertex B of $C_{(V_c)_i}$, we consider the rooted spanning tree t of the connected component of S containing B and the edge $e = AB$ of t whose end-points are B and its father.

The convex hull is obtained by deforming the cube. If B is a white vertex belonging to a planar face T, the end-point A of the edge e cannot be coplanar with the vertices of T. If not, the vertices of T and the point A define a planar face containing T.

Let $S(B)$ be the star of the white vertex to treat and $S(e)$ the star of the edge e. Next, we degenerate e.

1. Let f be a square face contained in $S(B)$.

   (a) If $f \notin S(e)$, then f is deformed into a face of four non-coplanar vertices. In this case, in order to avoid convexity problems related to this non-planar face, we attach the edge whose vertices are the other end-points of the edges of f incident to B. This edge allows us to cut up the non-planar face into two triangular faces sharing the attached edge. See Figure 14.

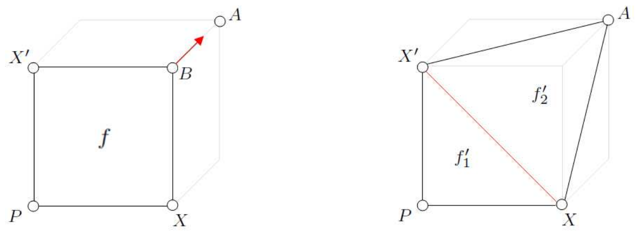


Figure 14: The square face f is deformed into a non-planar face with four vertices, as a consequence of degenerating an edge which is not incident to f. To avoid convexity problems, the red edge is attached. This edge cuts up the non-planar face into two triangular faces, f'_1 and f'_2.

   (b) If $f \in S(e)$, then f is deformed into a triangular face f'. The vertices of f' are those of f except B (see Figure 15).

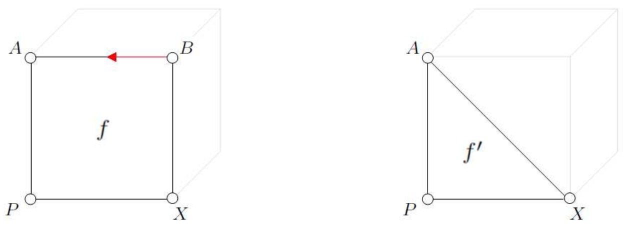


Figure 15: The square face f is deformed into the triangular face f', as a consequence of degenerating an edge which is incident to f.

2. Let T be a triangular face contained in $S(B)$.

   (a) If $T \notin S(e)$, then T is deformed into another triangular face T'. The vertices of T' are those of T except B, which is replaced with A. Figure 16 is an example.

   (b) If $T \in S(e)$, then T degenerates into an edge $e' \neq e$ incident to T. The end-points of e' are the vertices of T except B. The degenerated edge leads to a degenerated triangular face. This degenerated triangular face must be removed starting from the third edge $a \neq e, e'$ incident to T (see Figure 17).

3. Let V be a volume contained in $C_{(V_c)_i}$ and let $\mathcal{F}$ be the set of faces of V.

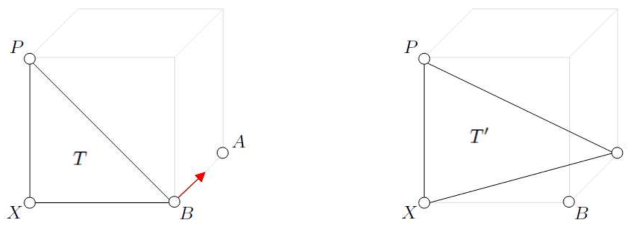


Figure 16: The triangular face T is deformed into another triangular face T', as a consequence of degenerating an edge which is not incident to T.

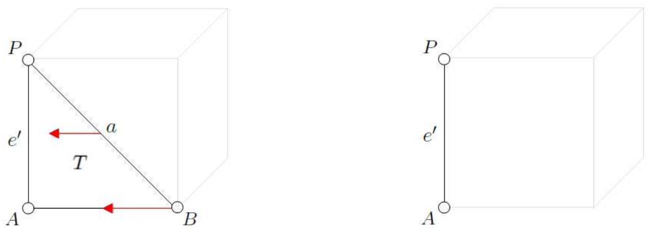


Figure 17: The triangular face T degenerates into an edge e' which is incident to T. This degenerated face appears as a consequence of degenerating an edge of the cube which is incident to T. The degenerated face is removed starting from the edge a.

If there exists only one face $f \in \mathcal{F} - S(B)$, then V degenerates into f. The degenerated edge leads to a degenerated volume. This degenerated volume must be removed starting from a face incident to it (see Figures 18 and 19).

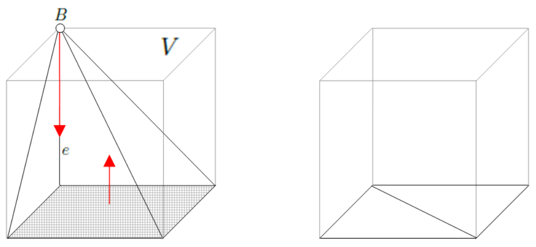


Figure 18: The only face of the pyramid which does not belong to $S(B)$ is its square base, so the pyramid degenerates into it. This degenerated pyramid is removed starting from its square base.

Remark 5 *Algorithm A.4 (in Appendix) has been implemented including the previous cases. Since for each pattern subset this algorithm returns its convex hull, we get a constructive proof of having considered all the possible cell deformations and degeneracies.*

As a consequence of these operations, two coplanar triangular faces sharing an edge may have appeared (see Figure 18). In this case, we must remove the

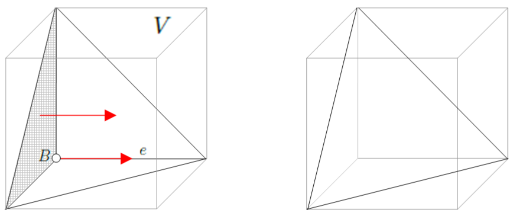


Figure 19: There exists only one face of the tetrahedron which does not belong to $S(B)$, so the tetrahedron degenerates into it. This degenerated tetrahedron is removed starting from the only face contained in $S(B) - S(e)$.

shared edge. The two coplanar triangular faces become only one square face. Hence, the vertices of this square face are those of the two triangular faces.

Summarizing, the developed technique for constructing the convex hull of a pattern subset consists in: (1) attaching edges to convert non-planar faces with four vertices into two triangular faces sharing an edge; (2) studying the degenerated i–cells, for $i = 1, 2, 3$; and (3) removing edges for converting two coplanar triangular faces into only one square face.

In Algorithm A.4 (in Appendix), we define the procedure for: (1) deforming the unit cube into the convex hull of any pattern subset $(V_c)_i \subset \mathbb{Z}^3$; (2) constructing the cell defined by this pattern subset; and (3) computing its boundary.

Remark 6 *Given a pattern subset $(V_c)_i$, Algorithm A.4 (in Appendix) deforms and degenerates the cells contained in the unit cube C_V for computing the convex hull of the points of $(V_c)_i$. In this way, the cell $C((V_c)_i)$ defined by $(V_c)_i$ is determined by the points inside this convex hull. Moreover, the boundary of $C((V_c)_i)$ is computed in terms of the vertices, edges and faces of the convex hull. More concretely, $\partial(C((V_c)_i))$ is given by Ve', Ed', Fa'.*

By using as input the 22 pattern subsets shown in Table 3, Algorithm A.4 (in Appendix) returns the 22 pattern cells shown in Table 4. Let us note that 12 of them are 3–cells (see also Figure 1 in [1]).

Remark 7 *The pattern cells shown in Table 4 are stored in a look-up table. As commented in Section 2, this table is used to construct a cell complex from a 3–dimensional binary digital image. Moreover, Theorem 3 can be proved.*

Theorem 3 *The cell complex constructed from a given 3–dimensional binary digital image on a dual grid is made up by combining (up to isometries) the 22 pattern cells shown in Table 4, taking into account that it is not necessary to use all them and any of them can be used more than once.*

3.3 Comparison with Kenmochi et al. method

In a similar way as Kenmochi et al. in [6, 7, 8], the above-described method constructs cell complexes from 3–dimensional binary digital images on a dual grid, although there exist some differences.

A minor difference between the results obtained by Kenmochi et al. and our results is the number of different subsets on a cube. They consider that two

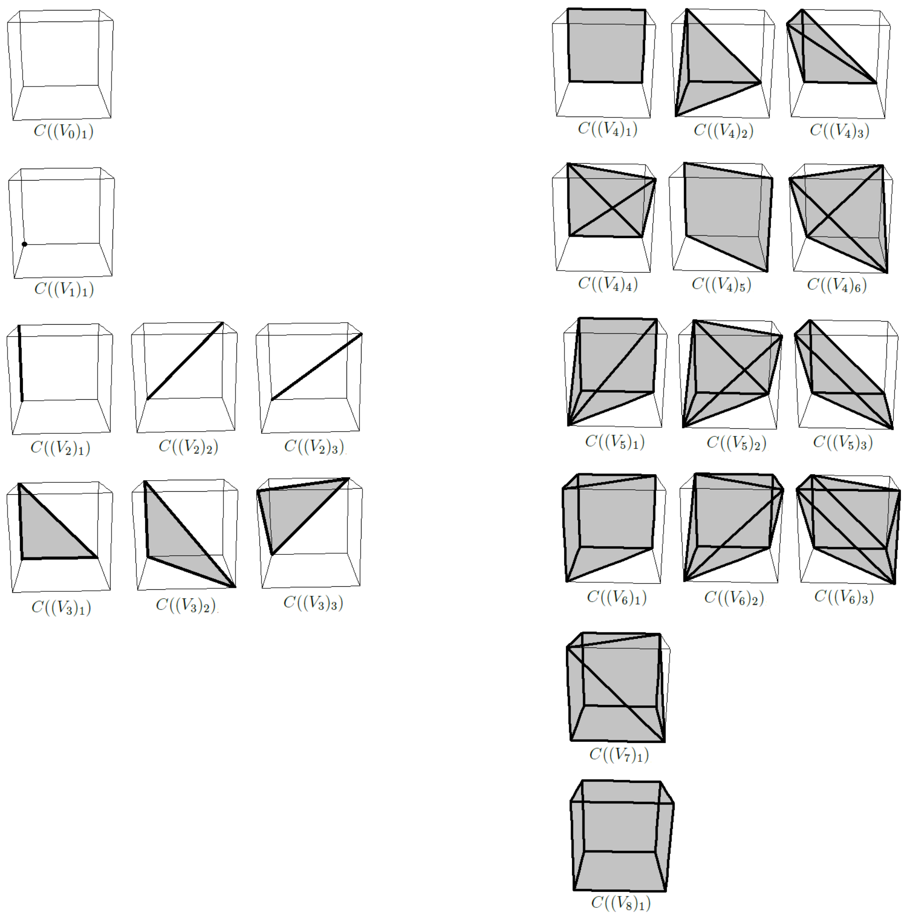


Table 4: Pattern cells defined by the pattern subsets shown in Table 3.

subsets are the same if one is a rotation of the other one; whereas we consider that two subsets are the same if one is an isometry of the other one. In this way, Kenmochi et al. obtain 23 different subsets (see Table 1 at page 6); whereas we obtain 22 (see Table 3 at page 18). Moreover, each subset in Table 1 corresponds (up to rotations) to a pattern subset in Table 3, except those with four points. In this case, Kenmochi et al. obtain 7 subsets; whereas we obtain 6. More concretely, P4c and P4d in Table 1 coincide (up to isometries) with the pattern subset $C((V_4)_3)$ in Table 3. Kenmochi et al. consider that P4c and P4d are different subsets because there does not exist a rotation which sends P4c into P4d; whereas we consider that they are identical because there exists a reflection from P4c into P4d.

The main difference with Kenmochi et al. method is that we propose a procedure which allows us to construct the pattern subsets and the pattern cells. Moreover, this procedure can be implemented and generalized to dimension 4, as we show in the following section.

4 Extracting cell complexes from 4–dimensional digital images

In this section, we extend the procedure developed in Section 3.2 for constructing the pattern subsets and pattern cells in dimension 4. In this way, as commented in Section 2, we obtain a method for constructing cell complexes from 4–dimensional dual binary digital images.

4.1 Pattern subsets

We compute (up to isometries) the different subsets of vertices of a 4–cube. There exist 65536 subsets of points which can be constructed from the vertices of a 4–cube. We extend to dimension four the procedure described in Section 3.2.1 for computing the isometric subsets.

In order to compute the isometric subsets which can be made up from the vertices of a 4–cube, (1) we determine the group of isometries of a 4–cube and (2) we apply it to each subset. More concretely, **Automorphisms[Hypercube[4]]** in the *Mathematica* package **Combinatorica'** returns a list with the 384 permutations of vertices which leave the unit 4–cube invariant. Moreover, by permuting the vertices of each subset of the unit 4–cube according to this list, we obtain the subsets isometric to it.

In this way, (a) we consider an ordered list with the 65536 subsets which can be constructed from the vertices of a 4–cube and (b) for each subset of the list, we compute and remove the subsets of the list isometric to it. In the last stage, we obtain a list with the pattern subsets.

In Appendix, we show a table with the obtained pattern subsets.

As main result, we can establish Theorem 4 (extension to dimension 4 of Theorem 2).

Theorem 4 *Any 4–dimensional binary digital image on a dual grid is made up by combining (up to isometries) the 402 pattern subsets shown in Appendix, taking into account that it is not necessary to use all them and any of them can be used more than once.*

4.2 Pattern cells

We generalize the computational method shown in Section 3.2.2 in order to determine the convex hull of the pattern subsets in $\mathbb{Z}^4$. In a similar way, the technique consists in deforming the 4–cube which contains to the subset. The deformation is a consequence of degenerating the edges of the 4–cube incident to white vertices. In this case, the degeneracies of the edges of the 4–cube can lead to the deformations and/or degeneracies of faces, volumes and hypervolumes contained in the 4–cube (see Figure 20).

Remark 8 *Let us recall that in dimension 3, the convex hull may not be obtained by degenerating edges not belonging to the cube. In dimension 4, this problem is inherited. In Figure 21, we show a 4–dimensional example obtained by extending a similar case to that shown in Figure 4 in Section 3.2.2.1.*

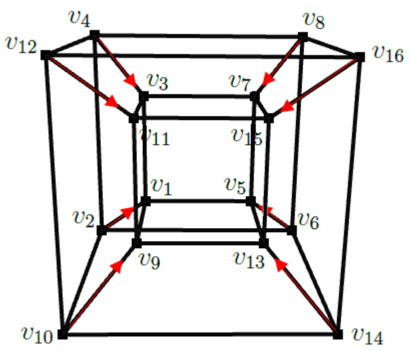


Figure 20: The degeneracies of the edges $v_1v_2, v_3v_4, v_5v_6, v_7v_8, v_9v_{10}, v_{11}v_{12}, v_{13}v_{14}, v_{15}v_{16}$ into the vertices $v_2, v_4, v_6, v_8, v_{10}, v_{12}, v_{14}, v_{16}$, respectively, lead to the fact that the 4–cube is degenerated into the cube $v_1v_3v_5v_7v_9v_{11}v_{13}v_{15}$.

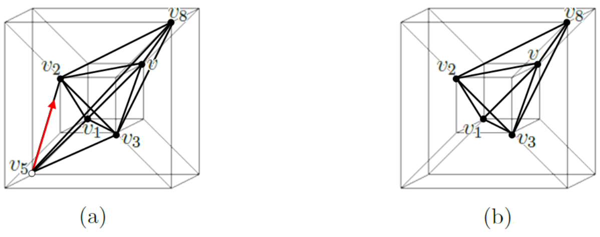


Figure 21: (a) The hypervolume $v_1v_2v_3v_5v_8v$, which is bordered by the volumes $v_1v_2v_3v_5$, $v_1v_2v_3v$, $v_1v_2v_5v$, $v_1v_3v_5v$, $v_2v_3v_5v_8$, $v_2v_3v_5v$, $v_2v_3v_8v$, $v_2v_5v_8v$, $v_3v_5v_8v$, is deformed by degenerating the edge v_2v_5. More concretely, the tetrahedra $v_1v_2v_3v, v_2v_3v_8v$ are invariant, the tetrahedra $v_1v_3v_5v, v_3v_5v_8v$ are deformed into the tetrahedra $v_1v_2v_3v, v_2v_3v_8v$, respectively, and the tetrahedra $v_1v_2v_3v_5, v_1v_2v_5v, v_2v_3v_5v_8, v_2v_3v_5v, v_2v_5v_8v$ are degenerated into the triangles $v_1v_2v_3, v_1v_2v, v_2v_3v_8, v_2v_3v, v_2v_8v$, respectively. (b) The vertices v_1, v_2, v_3, v_8, v define a hypertetrahedron, but the convex hull of these vertices cannot be obtained from (a). Moreover, we obtain similar results by degenerating the edge v_5v or v_3v_5, but if we degenerate the edge v_1v_5 or v_5v_8, we obtain the convex hull.

Remark 9 *Analogously to dimension 3, the edges of the 4–cube have to degenerate following a certain order; otherwise, at one of the steps of the deformation of the 4–cube, the white vertex to treat may not be incident to any edge of the 4–cube. Figure 22 shows a generalization of the example presented in Figure 12 in Section 3.2.2.2.*

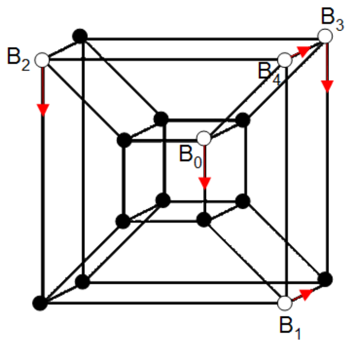


Figure 22: If we degenerate (1) the edge incident to B_0 marked with an arrow, (2) the edge incident to B_1 marked with an arrow, (3) the edge incident to B_2 marked with an arrow, and (4) the edge incident to B_3 marked with an arrow, then there is not any edge of the 4–cube incident to B_4 for degenerating.

A procedure similar to that shown in Section 3.2.2.2 allows us to determine an order relation on the edge degeneracies, in such a way that at each step of the deformation of the 4–cube, there exists an edge of it incident to the white vertex to treat. More concretely, the procedure consists in: (1) considering the white vertices of the 4–cube and the edges which join them as a subgraph S of the 4–cube; and (2) constructing a rooted spanning tree of each connected component of S, whose root is a white vertex adjacent to a black vertex of the 4–cube. These trees establish a hierarchy on the set of white vertices and, consequently, they determine an order for degenerating the edges incident to white vertices, where the last edge in degenerating is that made up by the root and by a black vertex adjacent to it. Let us note that in dimension 4, a vertex of the rooted spanning tree can be adjacent to at most four vertices of the tree, i.e. a father can have at most three children. See Figure 23 for an example.

4.2.1 Convex hull of a pattern subset

We extend the procedure shown in Section 3.2.2.3 in order to construct the convex hull of any pattern subset in $\mathbb{Z}^4$. In this way, we determine (up to isometries) the cells (together with their boundary) which can be obtained by deforming a 4–cube. Analogously, we degenerate the edges of the 4–cube incident to white vertices.

As commented at the beginning of Section 4.2, the degeneracies of edges of the 4–cube incident to white vertices can lead to the degeneracies of faces, volumes, and/or hypervolumes contained in the 4–cube. Below, we relate the existence of degenerated faces, volumes, and/or hypervolumes to the star of the degenerated edge and to that of the white vertex to treat.

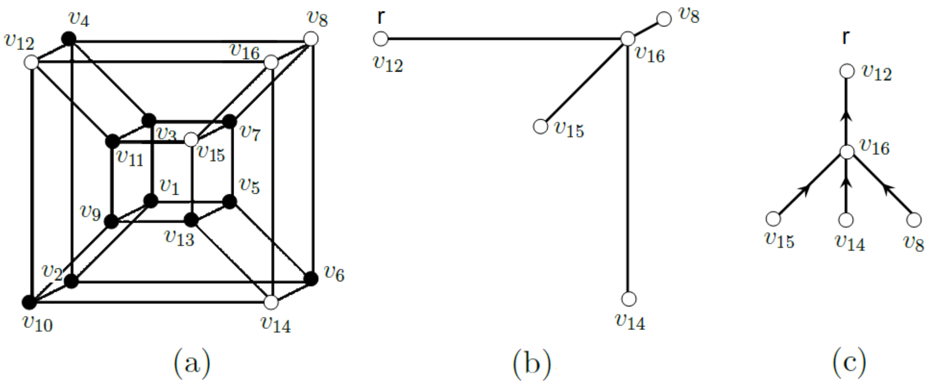


Figure 23: From left to right: (a) one of the pattern subsets with eleven points; (b) the subgraph of the 4–cube obtained by deleting the black vertices (and, obviously, the edges incident to these vertices); (c) the rooted spanning tree of the only connected component of the subgraph. The root of the tree is denoted by r and the arrows indicate how to degenerate the edges.

Let $(V_c)_i$ be a pattern subset contained in the 4–cube $HC_{(V_c)_i}$ and let S be the subgraph of $HC_{(V_c)_i}$ constructed by deleting the points of $(V_c)_i$. For each white vertex B of $HC_{(V_c)_i}$, we consider the rooted spanning tree t of the connected component of S containing B and the edge $e = AB$ of t whose end-points are B and its father.

The convex hull is obtained by deforming the 4–cube. If B is a white vertex of a spatial volume v, the end-point A of the edge e cannot be cospatial with the vertices of v. If not, the vertices of v and the point A define a spatial volume containing v.

Let $S(B)$ be the star of the white vertex to treat and $S(e)$ the star of the edge e. Next, we degenerate e.

1. Let v be a tetrahedron contained in $S(B)$.
   (a) If $v \notin S(e)$, then v is deformed into another tetrahedron v'. The vertices of v' are those of v except B which is replaced with A. In this case, we repeat the step 2(a) in Section 3.2.2.3 for each face T of v contained in $S(B)$. Figure 24 is an example.
   (b) If $v \in S(e)$, then, as commented in the three-dimensional case, v is degenerated into its only one face non-belonging to $S(B)$. In this case, we repeat the step 2 in Section 3.2.2.3 for each face T of v contained in $S(B)$ and the step 3 for v. The degenerated edge leads to a degenerated volume. This degenerated volume must be removed starting from a face incident to it (see Figure 19 in Section 3.2.2.3).
2. Let v be a pyramid contained in $S(B)$.
   (a) If $v \notin S(e)$, then v is deformed into another (spatial or not) volume with five vertices. Below, we detail both cases:

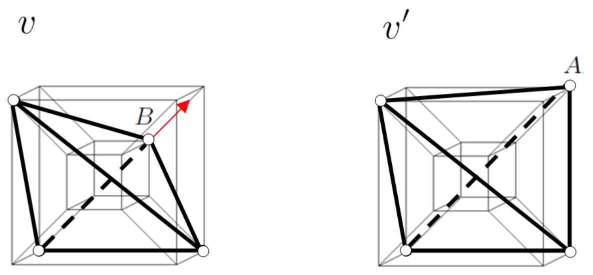


Figure 24: The tetrahedron v is deformed into another tetrahedron v', as a consequence of degenerating an edge e satisfying that $v \notin S(e)$.

i. If B is the apex of v, then v is deformed into another pyramid v' whose base is that of v and whose apex is A. In this case, we repeat the steps 1(a) and 2(a) in Section 3.2.2.3 for each face of v contained in $S(B)$. See Figure 25.

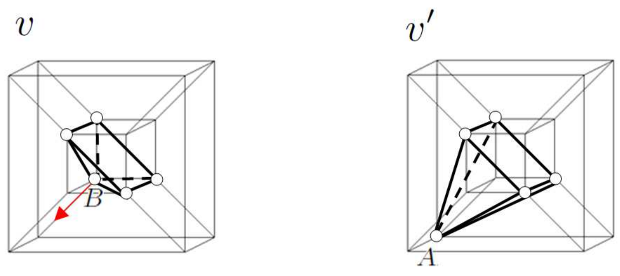


Figure 25: A pyramid v is deformed into another pyramid v', as consequence of degenerating an edge incident to the apex of v.

ii. If B is a vertex of the base of v, then v is deformed into a volume v' of five non-cospatial vertices. In this case, we repeat the steps 1(a) and 2(a) in Section 3.2.2.3 for each face of v contained in $S(B)$. Moreover, in order to avoid convexity problems related to the non-spatial volume v', we attach the triangular face whose vertices are the other end-points of the edges of v incident to B. This face allows us to cut up the non-spatial volume into two tetrahedra sharing a face. The shared face is the attached face. See Figure 26.

(b) If $v \in S(e)$, then, as commented in the three-dimensional case, v is degenerated into its only one face non-belonging to $S(B)$. In this case, we repeat the steps 1 and 2 in Section 3.2.2.3 for each face of v contained in $S(B)$, the step 3 for v, and we treat the appearance of coplanar triangular faces. The degenerated edge leads to a degenerated volume. This degenerated volume must be removed starting from a face incident to it (see Figure 18 in Section 3.2.2.3).

3. Let v be a volume (different from a pyramid) with $p \geq 5$ vertices contained in $S(B)$.

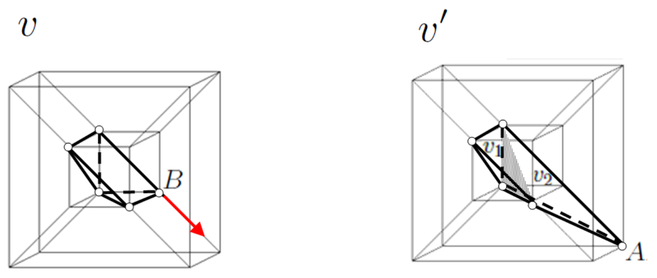


Figure 26: A pyramid v is deformed into a non-spatial volume v' with five vertices, as a consequence of degenerating an edge e satisfying that $v \notin S(e)$. To avoid convexity problems, the shaded face is attached. This face cuts up the non-spatial volume into two tetrahedra v_1 and v_2.

(a) If $v \notin S(e)$, then v is deformed into a volume v' of $p \geq 5$ non-cospatial vertices. In this case, we repeat the steps 1(a) and 2(a) in Section 3.2.2.3 for each face of v contained in $S(B)$. Moreover, in order to avoid convexity problems related to the non-spatial volume v', we attach the face f whose vertices are the other end-points of the edges of v incident to B. This face allows us to cut up the non-spatial volume v' into two volumes v_1 and v_2 (see Figure 27). More concretely, v_1 is the volume consisted of $p-1$ cospatial vertices which coincide with those of v except B; and v_2 is the volume whose vertices are A and the vertices of the attached face. Moreover, the relation between the vertices of f and those of v_2 is shown in Property 1.

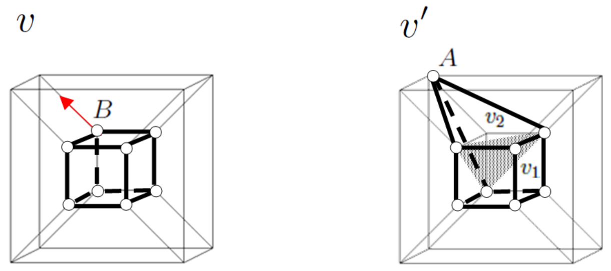


Figure 27: The volume v' of non-cospatial vertices is cut up into two volumes with a common face.

Property 1 *f is a face of four non-coplanar vertices if and only if v_2 is a volume of five non-cospatial vertices. Moreover, by replacing A with B in the set of vertices of v_2, we obtain (up to isometry) the pattern cell $C((V_5)_2)$ (see Table 8 in Section 3.2.2.3).*

If f is a face of four non-coplanar vertices, then, in order to avoid convexity problems related to this non-planar face, we attach the edge whose end-points are the only two vertices of f which do not belong to any face of v. This edge allows us to cut up the non-planar face into two triangular faces sharing an edge. The shared edge is the attached edge.

Moreover, Property 1 assures that v_2 is a volume of five non-cospatial vertices. In order to avoid convexity problems related to this non-spatial volume, we attach the face whose vertices are A and the vertices of the attached edge. This face allows us to cut up the non-spatial volume v_2 into two tetrahedra sharing a face. The shared face is the attached face. Furthermore, each of these tetrahedra is attached to v_1 by one of the triangular faces forming f. All this has allowed us to cut up the non-spatial volume v' into three volumes sharing a face two by two. See Figure 28.

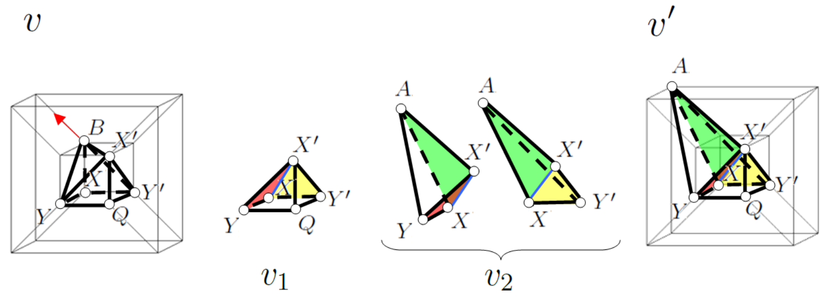


Figure 28: The volume $v' = AXX'YY'Q$ of non-cospatial vertices is cut up into the volumes $v_1 = XX'YY'Q$ of cospatial vertices and $v_2 = AXX'YY'$ of non-cospatial vertices sharing the non-planar face $f = XX'YY'$. Moreover, f is cut up into two triangular faces sharing the edge XX', and v_2 is cut up into two tetrahedra $AXX'Y$ and $AXX'Y'$ sharing the face AXX'.

(b) If $v \in S(e)$, then v is deformed into a volume v' of $p-1 \geq 4$ cospatial vertices. The vertices of v' are those of v except B. In this case, we repeat the steps 1 and 2 in Section 3.2.2.3 for each face of v and we treat the appearance of coplanar triangular faces. See Figure 29 for an example.

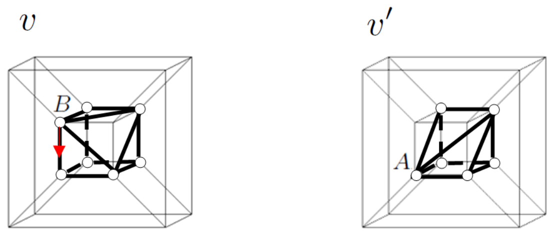


Figure 29: A volume v with 7 vertices is deformed into a volume v' with 6 vertices.

4. Let HV be a hypervolume contained in $HC_{(V_c)_i}$, and let $\mathcal{V}$ be the set of volumes of HV.

   If there exists only one volume $O \in \mathcal{V} - S(B)$, then HV degenerates into O. This degenerated hypervolume must be removed starting from a volume incident to it (see Figure 30). The degenerated edge leads to a degenerated hypervolume.

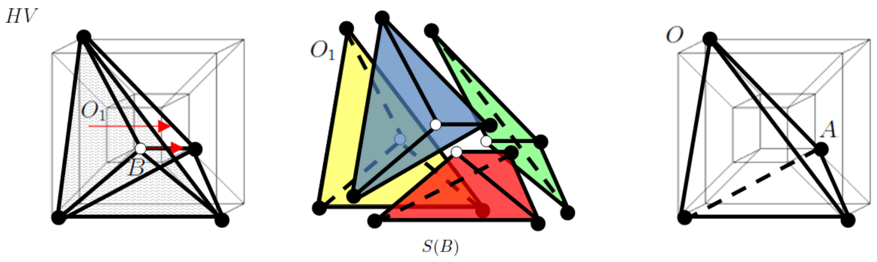


Figure 30: There exists only one tetrahedron O of the hypertetrahedron HV which does not belong to $S(B)$, so HV degenerates into O. This degenerated hypertetrahedron is removed starting from the only volume O_1 in $S(B) - S(e)$.

Remark 10 *In an analogous way to the three–dimensional case, an algorithm (Algorithm A.5, in Appendix) including the previous cases has been implemented. Since for each pattern subset this algorithm returns its convex hull, we get a constructive proof of having considered all the possible cell deformations and degeneracies.*

The corresponding operations in the three-dimensional case led to the fact that two coplanar triangular faces sharing an edge may appear. In the four-dimensional case, cospatial volumes (with up to seven vertices) sharing a face may have appeared. By reasoning in a similar way as Section 3.2.2.3, we remove the shared face and the edges which are inside of another volume. More concretely, in dimension four, it may have appeared (a) three cospatial volumes with common faces two by two sharing an edge. In this case, we remove the shared edge, three common faces, and three volumes, and we attach a new volume made by the union of those three volumes (see Figure 31); and it may also have appeared (b) two cospatial volumes sharing a face. In this case, we remove the common face, and both volumes, and we attach a new volume made up by the union of those two volumes (see Figure 32).

Summarizing, the technique developed for constructing the convex hull of a pattern subset in $\mathbb{Z}^4$ includes, in addition to the operations of the three-dimensional case, the steps of: (1) attaching faces for converting non-spatial volumes into spatial volumes sharing a face two by two; (2) studying the degenerated 4–cells; and (3) removing faces and edges for converting cospatial volumes into only one volume.

The previous results allow us to define an algorithm (Algorithm A.5, in Appendix), which generalizes Algorithm A.4 (in Appendix), for: (1) deforming the unit 4–cube into the convex hull of any pattern subset $(V_c)_i \subset \mathbb{Z}^4$; (2) constructing the cell defined by this pattern subset; and (3) computing its boundary.

Remark 11 *Given a pattern subset $(V_c)_i$, Algorithm A.5 (in Appendix) deforms and degenerates the cells contained in the unit 4–cube HC_V for computing the convex hull of the points of $(V_c)_i$. In this way, the cell $C((V_c)_i)$ defined by*

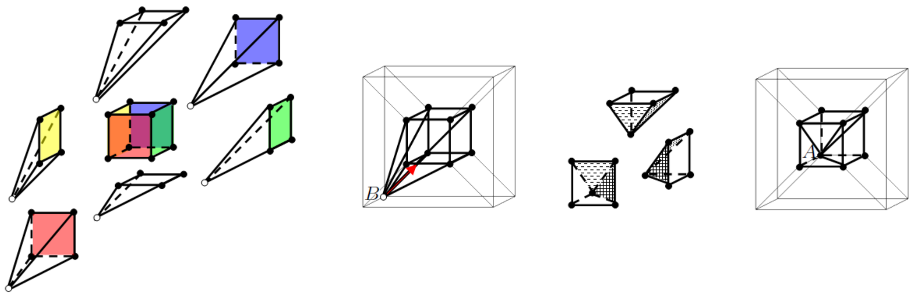


Figure 31: By degenerating an edge of a hypervolume with nine vertices, which consists of a cube with one pyramid attached on each of its faces, we obtain three pyramids forming a cube with an edge inside of it.

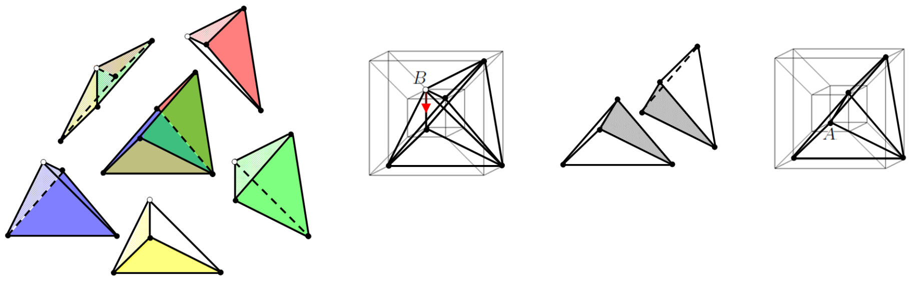


Figure 32: By degenerating an edge of a hypervolume with six vertices, which consists of one pyramid attached to an other on its square face, and four tetrahedra attached to the corresponding pairs of triangular faces of both pyramids, we obtain two cospatial tetrahedra sharing a face. By removing this face, we obtain a pyramid.

$(V_c)_i$ is determined by the points inside this convex hull. Moreover, the boundary of $C((V_c)_i)$ is computed in terms of the vertices, edges, faces, and volumes of the convex hull. More concretely, $\partial(C((V_c)_i))$ is given by Ve', Ed', Fa', Vo'.

By using as input the 402 pattern subsets shown in Appendix, Algorithm A.5 (in Appendix) returns the 402 pattern cells. In Figure 33 [1], we show an example of some of them.

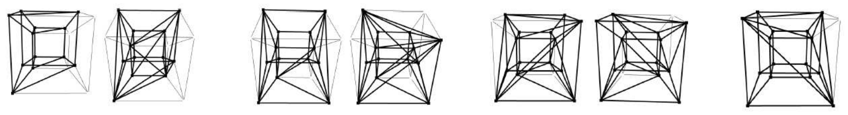


Figure 33: Some of the 4–dimensional pattern cells with 12, 13, 14 and 15 vertices, respectively.

[1]screenshots of "**Hypercube**" software developed by Régis Meyssonnier in his Final Master Project supervised by Prof. Jean-Luc Mari from University of Marseille (see [13])

Remark 12 *The pattern cells are stored in a look-up table. As commented in Section 2, this table is used to construct a cell complex from a 4–dimensional binary digital image. Moreover, Theorem 5 can be proved.*

Theorem 5 *The cell complex constructed from a given 4–dimensional binary digital image on a dual grid is made up by combining (up to isometries) the 402 pattern cells, taking into account that it is not necessary to use all them and any of them can be used more than once.*

5 Conclusions

In this paper, we have defined a method for constructing cell complexes from n–dimensional binary digital images on a dual grid. Moreover, this method has been implemented for $n = 3, 4$.

The first stage of the method consists in determining the pattern subsets which can be obtained from the vertices of a n–cube and constructing the convex hull of these subsets, i.e. the pattern cells. The second stage is scanning each subset of vertices of a n–cube of the grid and associating it with one of the pattern subset determined in the first stage. We consider the convex hull of the pattern subset and we invert the isometry between this one and its corresponding subset of vertices, in this way, we obtain the cells of the cell complex. The cell complex is formed by attaching the cells along their boundaries. Finally, this complex is simplified by extracting its boundary.

The pattern subsets are determined by using an algorithm whose input are (1) all the subsets of points which can be constructed from the vertices of a n–cube and (2) the group of isometries of a n–cube.

The convex hull of the pattern subsets is constructed in two steps: (1) a first step consists in determining an order on the edge degeneracy; and (2) a second step is giving a list of rules for establishing (2.1) the stages of "deformation" and "degeneracy" of cells of a n–cube and (2.2) the stage of "correction" which allows us to assure the convexity.

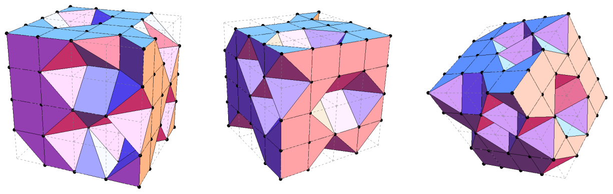
We think that, in a theoretical way, the method developed in this paper can be generalized to dimension n. In order to get this goal, it is necessary: (a) to determine the pattern subsets in nD, by computing the group of isometries of an n–cube and (b) to construct the convex hull of these subsets, by giving an order on the edge degeneracy and by deforming/degenerating (b.1) cells of an $(n-1)$–cube and (b.2) $(n-1)$–cells of an n–cube. In practice, it is not possible to store the pattern subsets in nD, for $n \geq 6$, since the number of pattern subsets in $6D$ is of the order of 10^{14} (see [5], where the number of irreducible Boolean functions of n variables coincides with the number of pattern subsets in nD).

A Appendix

Example 1 *Let* $\{\{0,0,0\},\{0,0,1\},\{0,2,1\},\{1,1,0\},\{1,1,1\},\{1,2,1\},\{2,0,1\},$ $\{2,1,1\},\{2,2,0\},\{3,3,4\},\{3,3,5\},\{3,4,3\},\{3,4,4\},\{3,5,4\},\{4,4,3\},\{4,4,5\},$

$\{4,5,3\}, \{4,5,4\}, \{5,0,0\}, \{5,0,4\}, \{5,0,5\}, \{5,1,0\}, \{5,1,5\}, \{5,3,3\}, \{5,3,4\}, \{5,3,5\}, \{5,4,3\}, \{5,4,4\}, \{5,5,0\}, \{5,5,3\}, \{5,5,4\}\}$ *be the set of the 31 black voxels of a binary digital image I on a dual grid of size $6 \times 6 \times 6$.*

The simplified cell complex obtained by applying the method above-described to I is shown in the following figure. Moreover, Algorithm A.3 determines that all the 0,1,2–cells of $CC(I)$ are incident to a 3–cell of $CC(I)$ and it computes the 174 2–cells of $CC(I)$ incident to exactly one of the 3–cells of $CC(I)$.



Three viewpoints of the simplified cell complex obtained from the image.

Example 2 *Let* $\{\{0,0,0,0\}, \{0,1,0,0\}, \{0,1,0,1\}, \{0,1,1,0\}, \{0,2,0,0\}, \{1,1,0,0\}\}$ *be the set of the 6 black 4–xels of a binary digital image I on a dual grid G made up by $2 \times 2 \times 2 \times 2$ 4–cubes.*

The cells of the cell complex obtained by applying the method above-described to I are:

Six singleton subsets corresponding to 0–cells.
$\{\{\{0,0,0,0\}\}, \{\{0,1,0,0\}\}, \{\{0,1,0,1\}\}, \{\{0,1,1,0\}\}, \{\{0,2,0,0\}\}, \{\{1,1,0,0\}\}\}$

Fourteen subsets made up by two points (1–cells).
$\{\{\{0,0,0,0\}, \{0,1,0,1\}\}, \{\{0,0,0,0\}, \{0,1,1,0\}\}, \{\{0,0,0,0\}, \{1,1,0,0\}\}, \{\{0,1,0,0\}, \{0,0,0,0\}\}, \{\{0,1,0,0\}, \{0,1,0,1\}\}, \{\{0,1,0,0\}, \{0,1,1,0\}\}, \{\{0,1,0,0\}, \{0,2,0,0\}\}, \{\{0,1,0,0\}, \{1,1,0,0\}\}, \{\{0,1,0,1\}, \{0,1,1,0\}\}, \{\{0,1,0,1\}, \{0,2,0,0\}\}, \{\{0,1,0,1\}, \{1,1,0,0\}\}, \{\{0,1,1,0\}, \{0,2,0,0\}\}, \{\{0,1,1,0\}, \{1,1,0,0\}\}, \{\{0,2,0,0\}, \{1,1,0,0\}\}\}$

Sixteen subsets made up by three points (2–cells).
$\{\{\{0,0,0,0\}, \{0,1,0,1\}, \{0,1,1,0\}\}, \{\{0,0,0,0\}, \{0,1,0,1\}, \{1,1,0,0\}\}, \{\{0,0,0,0\}, \{0,1,1,0\}, \{1,1,0,0\}\}, \{\{0,1,0,0\}, \{0,0,0,0\}, \{0,1,0,1\}\}, \{\{0,1,0,0\}, \{0,0,0,0\}, \{0,1,1,0\}\}, \{\{0,1,0,0\}, \{0,0,0,0\}, \{1,1,0,0\}\}, \{\{0,1,0,0\}, \{0,1,0,1\}, \{0,1,1,0\}\}, \{\{0,1,0,0\}, \{0,1,0,1\}, \{0,2,0,0\}\}, \{\{0,1,0,0\}, \{0,1,0,1\}, \{1,1,0,0\}\}, \{\{0,1,0,0\}, \{0,1,1,0\}, \{0,2,0,0\}\}, \{\{0,1,0,0\}, \{0,1,1,0\}, \{1,1,0,0\}\}, \{\{0,1,0,0\}, \{0,2,0,0\}, \{1,1,0,0\}\}, \{\{0,1,0,1\}, \{0,1,1,0\}, \{0,2,0,0\}\}, \{\{0,1,0,1\}, \{0,1,1,0\}, \{1,1,0,0\}\}, \{\{0,1,0,1\}, \{0,2,0,0\}, \{1,1,0,0\}\}, \{\{0,1,1,0\}, \{0,2,0,0\}, \{1,1,0,0\}\}\}$

Nine subsets made up by four points (3–cells).
$\{\{\{0,0,0,0\}, \{0,1,0,1\}, \{0,1,1,0\}, \{1,1,0,0\}\}, \{\{0,1,0,0\}, \{0,0,0,0\}, \{0,1,0,1\}, \{0,1,1,0\}\}, \{\{0,1,0,0\}, \{0,0,0,0\}, \{0,1,0,1\}, \{1,1,0,0\}\}, \{\{0,1,0,0\}, \{0,0,0,0\}, \{0,1,1,0\}, \{1,1,0,0\}\}, \{\{0,1,0,0\}, \{0,1,0,1\}, \{0,1,1,0\}, \{0,2,0,0\}\}, \{\{0,1,0,0\}, \{0,1,0,1\}, \{0,1,1,0\}, \{1,1,0,0\}\}, \{\{0,1,0,0\}, \{0,1,0,1\}, \{0,2,0,0\}, \{1,1,0,0\}\}, \{\{0,1,0,0\}, \{0,1,1,0\}, \{0,2,0,0\}, \{1,1,0,0\}\}, \{\{0,1,0,1\}, \{0,1,1,0\}, \{0,2,0,0\}, \{1,1,0,0\}\}\}$

Two subsets made up by five points (4–cells).
$\{\{\{0,0,0,0\}, \{0,1,0,0\}, \{0,1,0,1\}, \{0,1,1,0\}, \{1,1,0,0\}\}, \{\{0,1,0,0\}, \{0,1,0,1\}, \{0,1,1,0\}, \{0,2,0,0\}, \{1,1,0,0\}\}\}$

Algorithm A.1

```
Input: $I$ and $G$.
Output: subsets of $n$–xels of $I$.
begin
// $V(I)$: empty set to store the lists of $n$–xels of $I$.
  for each $n$–cube $C_i$ of $G$ do
    Let $L$ be the subset of $n$–xels of $I$ corresponding to the vertex set of $C_i$
    $V(I) = V(I) \bigcup \{L\}$
  end for
  return  $V(I)$
end
```

Algorithm A.2

```
Input: $V(I)$ and look-up table with the pattern subsets.
Output: look-up table with the pattern subsets of $I$ and the isometries between the subsets
of $V(I)$ and the pattern subsets.
// $P(I)$: empty table to store the pattern subsets of $I$ and the isometries between the subsets
of $V(I)$ and the pattern subsets.
begin
  for $X \in V(I)$ do
    if $Y$ is a pattern subset isometric to $X$ then
      $P(I) = P(I) \bigcup \{(Y, \sigma)\}$ where $\sigma$ is the isometry that satisfies $\sigma(X) = Y$
    end if
  end for
  return  $P(I)$
end
```

Algorithm A.3

```
Input: $C_i(V(I))$: $i$–cells of $CC(I)$, for $i = 0, 1, ..., n$.
       $\partial(C_n(V(I)))$: boundary of the cells in $C_n(V(I))$.
Output: simplified cell complex.
begin
// $Boundary$: empty list to store the cells of the simplified complex.
  for $i \in \{0, 1, ..., n-1\}$ do
    for each $i$–cell $c \in C_i(V(I))$ do
      if $c$ is not incident to a $n$–cell then
        $Boundary = Boundary \bigcup \{c\}$
      else
        if $i \neq n-1$ or $c$ is incident to exactly one $n$–cell then
          $Boundary = Boundary \bigcup \{c\}$
        end if
      end if
    end for
  end for
  return  $Boundary$
end
```

Algorithm A.4

```
Input: the unit cube $C_V$.
       a pattern subset $(V_c)_i \subset C_V$ with $c$ points.
Output: convex hull of $(V_c)_i$.
```

```
begin
// $Ve$: vertices of $C_V$.
// $Ed$: edges of $C_V$.
// $Fa$: faces of $C_V$.
// $Vo$: volume of $C_V$.
// $S$: subgraph of $C_V$ constructed by deleting the points of $(V_c)_i$.
// $WV$: set of white vertices of $C_V$ ordered according to Section 3.2.2.2.
 1: $Ed' = Ed$
 2: Construct the set $\mathcal{T}$ of all the rooted spanning trees of all the connected components of $S$
 3: for each $t \in \mathcal{T}$ do
 4:     for each $B \in t$ do
 5:         • Degenerate the edge incident to $B$ and to its father in $t$
            {This edge degeneracy replaces $B$ with $A$, where $A$ is the father of $B$ in $t$.}
 6:         • Obtain the lists $Ve'$, $Ed'$, $Fa'$, $Vo'$ of vertices, edges, faces and volumes, respec-
            tively, by replacing $B$ with $A$
 7:         for each $f' \in Fa'$ do
 8:             if $f'$ is a face of 4 non-coplanar vertices then
 9:                 • $Ed' = Ed' \bigcup \{XX'\}$, where $X, X' \in Ve'$ satisfy that $XA, X'A \in \partial(f')$
                    {We attach an edge for cutting up the non-planar face into two triangular
                    faces.}
10:                 • $Fa' = Fa' \bigcup \{XX'A\}$
11:                 • $Fa' = Fa' \bigcup \{XX'P\}$, where $P \neq A$ is such that $P \in Ve'$ and it satisfies
                    $XP, X'P \in \partial(f')$
                    {We attach the two triangular faces.}
12:                 • $Fa' = Fa' - \{f'\}$
                    {We remove the non-planar face.}
13:             end if
14:             if $f'$ is a face degenerated into an edge $e' \in Ed'$ then
15:                 $Fa' = Fa' - \{f'\}$
                    {We remove $f$ starting from $a$, where $f$ is the triangular face which became $f'$
                    when $B$ was replaced with $A$ and $a \neq e, e'$ is the third edge of $f$.}
16:             end if
17:             if $v' \in Vo'$ is a volume degenerated into $f'$ then
18:                 • $Vo' = Vo' - \{v'\}$
                    {We remove $v$ starting from $g$, where $v$ is the volume which became $v'$ when $B$
                    was replaced with $A$ and $g$ is a face incident to $v$.}
19:             end if
20:             if $f' \neq f'' \in Fa'$ are two coplanar triangular faces sharing an edge $e'' \in Ed'$
                then
21:                 if $\{f' + f''\} \notin Fa'$, where $f' + f''$ denotes the closure of the all the points
                    inside the convex hull of the vertices of $f'$ and $f''$ then
22:                     • $Fa' = Fa' \bigcup \{f' + f''\}$
                        {We attach the square face made up by the two coplanar triangular faces.}
23:                 end if
24:                 • $Fa' = Fa' - \{f'\}$
25:                 • $Fa' = Fa' - \{f''\}$
26:                 • $Ed' = Ed' - \{e''\}$
                    {We remove the two coplanar triangular faces and the edge shared by both
                    faces.}
27:             end if
28:         end for
29:     end for
30: end for
31: return $Ve', Ed', Fa'$
end
```

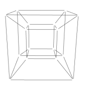
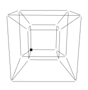
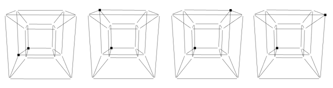
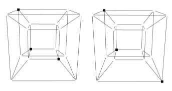
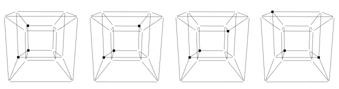
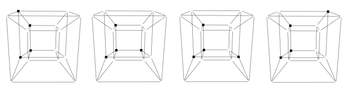
Algorithm A.5

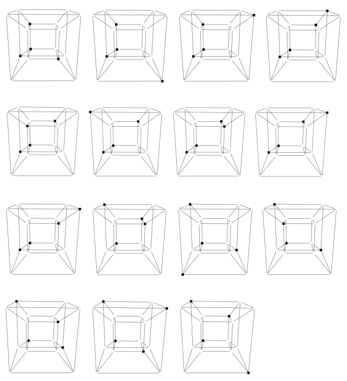
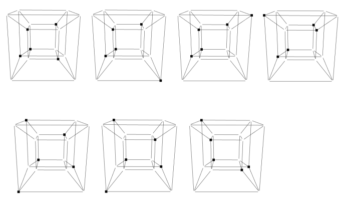
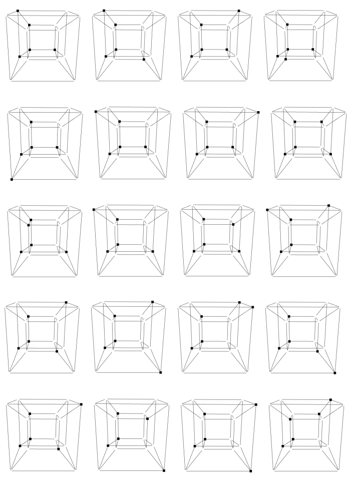
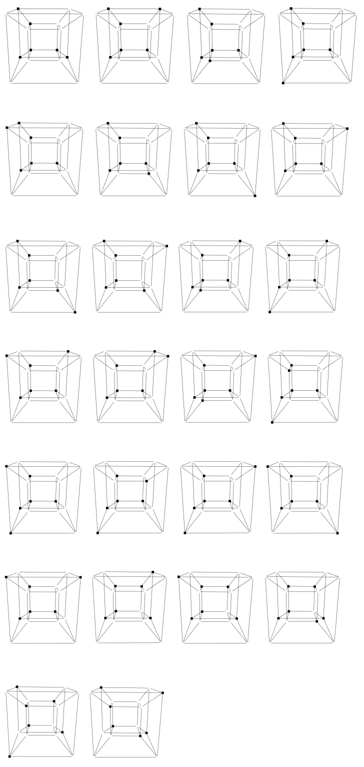
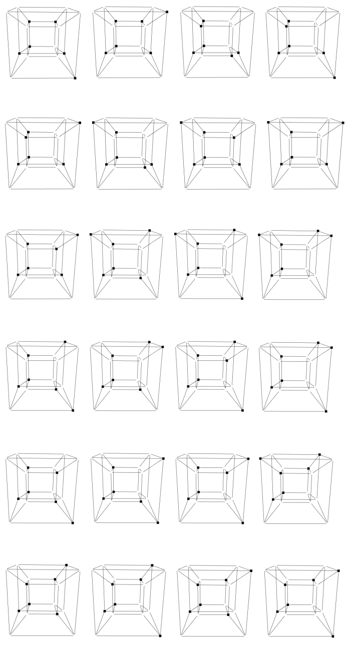
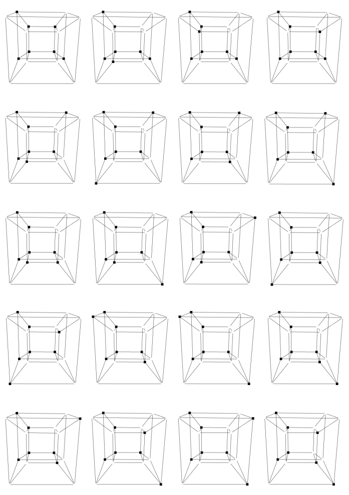
Input: the unit 4–cube HC_V.
a pattern subset $(V_c)_i \subset HC_V$ with c points.

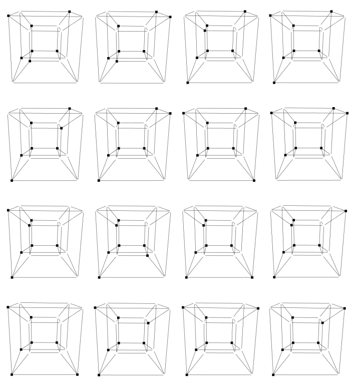
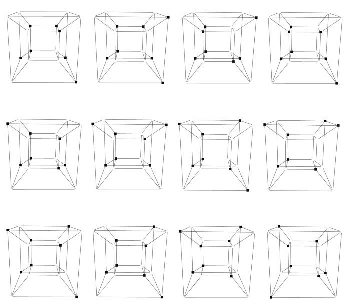
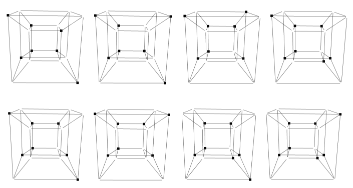
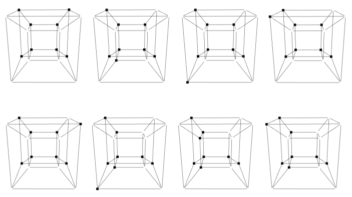
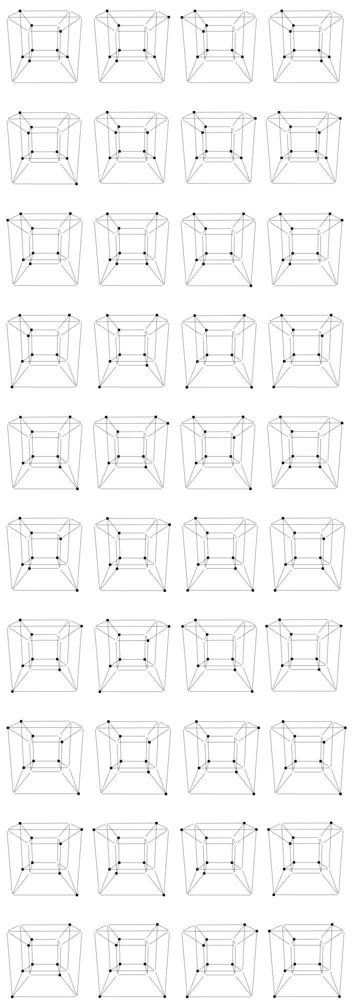
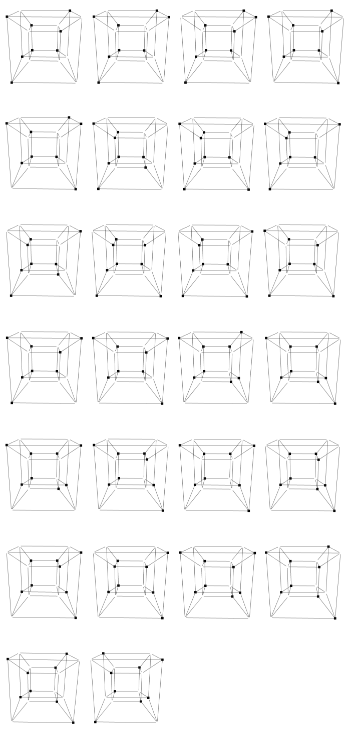
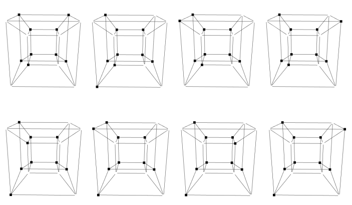
```
Output: convex hull of $(V_c)_i$.
begin
// $Ve$: vertices of $HC_V$.
// $Ed$: edges of $HC_V$.
// $Fa$: faces of $HC_V$.
// $Vo$: volumes of $HC_V$.
// $Hv$: hypervolume of $HC_V$.
// $S$: subgraph of $HC_V$ constructed by deleting the points of $(V_c)_i$.
// $WV$: set of white vertices of $HC_V$ ordered according to Remark 9.
  $Ed' = Ed$
  Construct the set $\mathcal{T}$ of all the rooted spanning trees of all the connected components of $S$
  for each $t \in \mathcal{T}$ do
    for each $B \in t$ do
      • Degenerate the edge incident to $B$ and to its father in $t$
      • Obtain the lists $Ve'$, $Ed'$, $Fa'$, $Vo'$, $Hv'$ of vertices, edges, faces, volumes, and
      hypervolumes, respectively, by replacing $B$ with $A$
      for $v' \in Vo'$ do
        if $v'$ is a non-spatial volume then
          for each $f' \in Fa' \bigcap \partial(v')$ do
            Lines 8–13 in Algorithm A.3
          end for
          {We define a face starting from the vertices of $v'$ which are attached to $A$ by an
          edge and we cut up $v'$ into two volumes}
          • $f' = \bigcup_i \{P_i\}$, where $P_i \in Ve' \bigcap \partial(v')$ is attached to $A$ by an edge
          • $v_1 = v' - \{A\}$
          • $v_2 = f' \bigcup \{A\}$
          • $Vo' = Vo' \bigcup \{v_1\}$
          • $Vo' = Vo' - \{v'\}$
          {Corollary 2 asserts that $f'$ is a non-planar face iff $v_2$ is a non-spatial volume.}
          if $f'$ is a face of 4 non-coplanar vertices then
            • $Ed' = Ed' \bigcup \{XX'\}$, where $X, X' \neq A$ are such that $X, X' \in Ve' \bigcap \partial(f')$
            and they satisfy $X, X' \notin \partial(g)$, $\forall g \neq f'$ such that $g \in Fa' \bigcap \partial(v')$
            • $Fa' = Fa' \bigcup \{XX'Y\}$, where $Y \neq X, X'$ is such that $Y \in Ve'$ and it satisfies
            $XY, X'Y \in \partial(f')$
            • $Fa' = Fa' \bigcup \{XX'Y'\}$, where $Y' \neq X, X'$ is such that $Y' \in Ve'$ and it
            satisfies $XY', X'Y' \in \partial(f')$
            {We attach the face defined by the vertex $A$ and the two end-points of the edge
            which cuts up $f'$ in two triangular faces and we cut up $v_2$ into two tetrahedra}
            • $Fa' = Fa' \bigcup XX'A$
            • $Vo' = Vo' \bigcup XX'YA$
            • $Vo' = Vo' \bigcup XX'Y'A$
          end if
          if $f'$ is a planar face then
            • $Fa' = Fa' \bigcup \{f'\}$
            • $Vo' = Vo' \bigcup \{v_2\}$
          end if
        end if
        if $v'$ is a spatial volume then
          for each $f' \in Fa' \bigcap \partial(v')$ do
            Lines 8–16 in Algorithm A.3
            Lines 20–27 in Algorithm A.3, taking into account that $f'' \in Fa' \bigcap \partial(v')$
          end for
        end if
        if $v'$ is a volume degenerated into a face then
          for each $f' \in Fa' \bigcap \partial(v')$ do
            Lines 14–27 in Algorithm A.3, taking into account that $f'' \in Fa' \bigcap \partial(v')$ in
            Lines 20–27
          end for
        end if
        if $hv' \in Hv'$ is a hypervolume degenerated into $v'$ then
          $HV' = HV' - \{hv'\}$
        end if
```

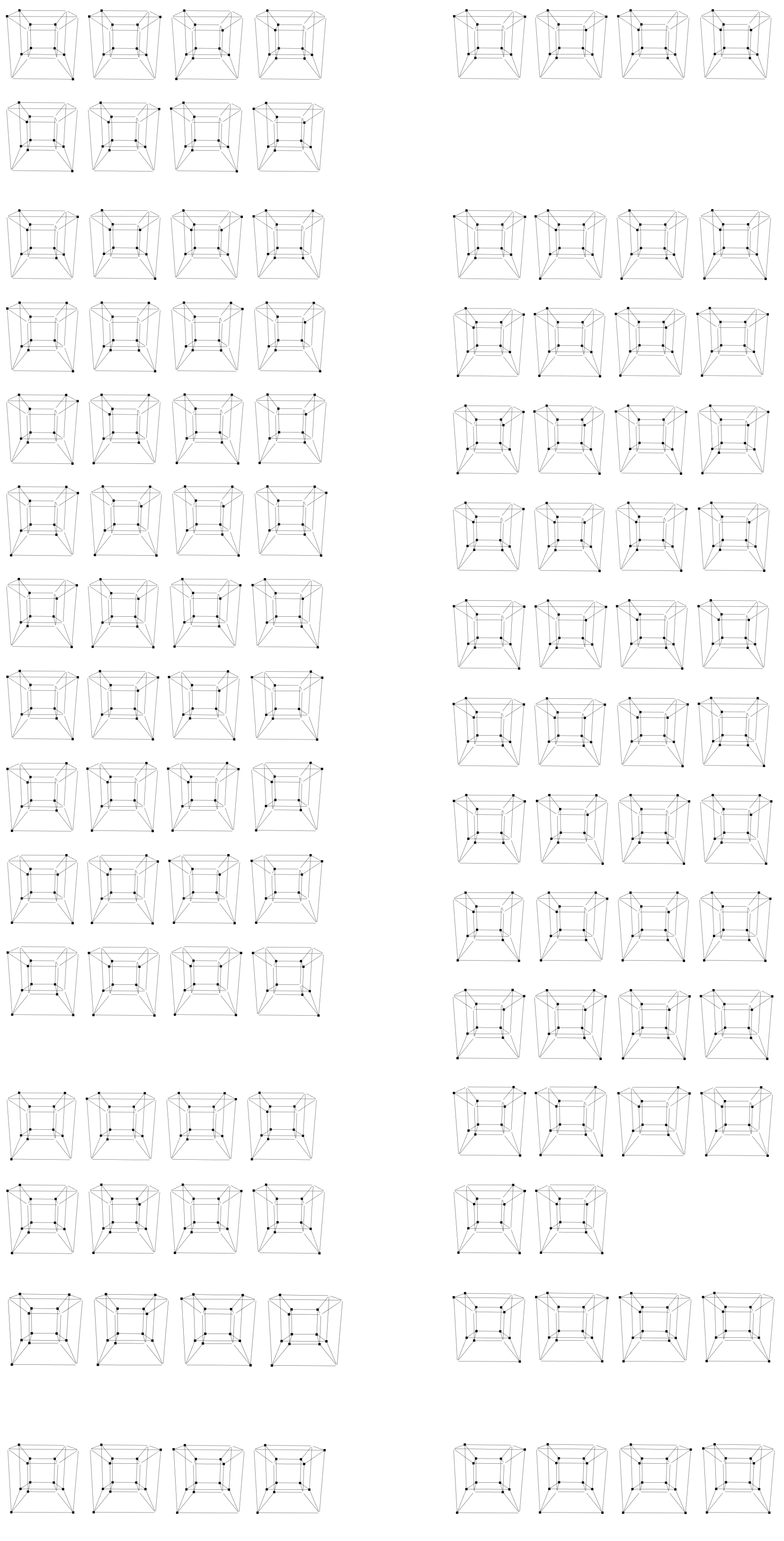
```
            if $v' \neq v'' \in Vo'$ are two cospatial volumes sharing a face $f'' \in Fa'$ then
                if $\exists\ v''' \in Vo'$ cospatial with $v'$ and $v''$ then
                    $\{v', v'''$ share a face $f''' \in Fa'.\}$
                    $\{v'', v'''$ share a face $f'''' \in Fa'.\}$
                    $\{v', v'', v'''$ share an edge $e'' \in Ed'$. Moreover, $e''$ is inside the volume $\{v' + v'' + v'''\}$, where $v' + v'' + v'''$ denotes the closure of all the points inside the convex hull of the vertices of $v'$, $v''$ and $v'''$.$\}$
                    if $\{v' + v'' + v'''\} \notin Vo'$ then
                        $Vo' = Vo' \bigcup \{v' + v'' + v'''\}$
                        for each $f' \in Fa' \bigcap \partial(v' + v'' + v''')$ do
                            Lines 20–27 in Algorithm A.3, taking into account that $f'' \in Fa' \bigcap \partial(v' + v'' + v''')$
                        end for
                    end if
                    • $Vo' = Vo' - \{v'\}$
                    • $Vo' = Vo' - \{v''\}$
                    • $Vo' = Vo' - \{v'''\}$
                    • $Fa' = Fa' - \{f''\}$
                    • $Fa' = Fa' - \{f'''\}$, where $f''' \in Fa' \bigcap \partial(v') \bigcap \partial(v''')$
                    • $Fa' = Fa' - \{f''''\}$, where $f'''' \in Fa' \bigcap \partial(v'') \bigcap \partial(v''')$
                    • $Ed' = Ed' - \{e''\}$, where $e'' \in Ed' \bigcap \partial(v') \bigcap \partial(v'') \bigcap \partial(v''')$
                end if
                if $\nexists\ v''' \in Vo'$ cospatial with $v'$ and $v''$ then
                    if $\{v' + v''\} \notin Vo'$, where $v' + v''$ denotes the closure of all the points inside the convex hull of the vertices of $v'$ and $v''$ then
                        $Vo' = Vo' \bigcup \{v' + v''\}$
                        for each $f' \in Fa' \bigcap \partial(v' + v'')$ do
                            Lines 20–27 in Algorithm A.3, taking into account that $f'' \in Fa' \bigcap \partial(v' + v'')$
                        end for
                    end if
                    • $Vo' = Vo' - \{v'\}$
                    • $Vo' = Vo' - \{v''\}$
                    • $Fa' = Fa' - \{f''\}$
                end if
            end if
        end for
    end for
end for
return $Ve', Ed', Fa', Vo'$
end
```

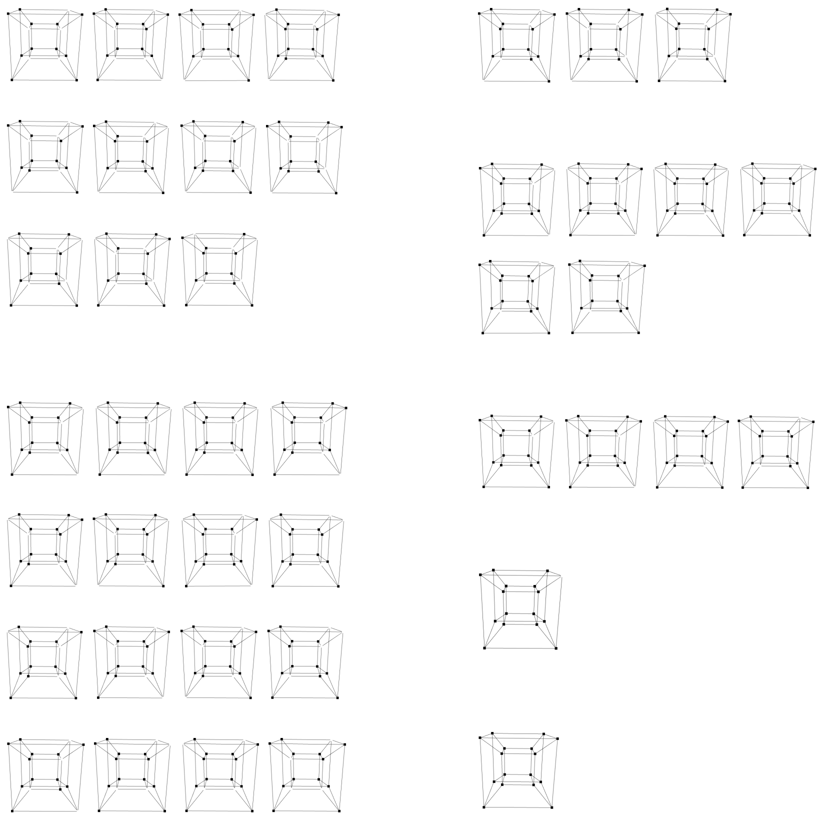
Table with the pattern subsets in dimension 4





References

[1] O. Aichholzer, *Extremal properties of 0/1-polytopes of dimension 5*, In G. Ziegler and G. Kalai (eds.), Polytopes - Combinatorics and Computation, Birkhäuser, 111–130, 2000.

[2] G. Damiand, *Définition et étude d'un modèle topologique minimal de représentation d'images 2d et 3d*, PhD thesis, Montpellier II University, 2001.

[3] G. Damiand, *Topological model for 3D Image Representation: Definition and Incremental Extraction Algorithm*, Computer Vision and Image Understanding 109(3): 260–289, 2008.

[4] F. Hanisch, *Marching square*, CGEMS: Computer graphics educational materials source, 2008.

[5] M. A. Harrison, *The Number of Transitivity Sets of Boolean Functions*, Journal of the Society for Industrial and Applied Mathematics 11(3): 806–828, 1963.

[6] Y. Kenmochi and A. Imiya, *Combinatorial boundary of a 3D lattice point set*, Visual Communication and Image Representation 17(4): 738–766, 2006.

[7] Y. Kenmochi, A. Imiya and N. F. Ezquerra, *Polyhedra generation from lattice points*, In S. Miguet, A. Montanvert and S. Ubéda (eds.), Discrete Geometry for Computer Imagery, Berlin Heidelberg: Springer-Verlag, 127–138, 1996.

[8] Y. Kenmochi, A. Imiya and A. Ichikawa, *Boundary Extraction of Discrete Objects*, Computer Vision and Image Understanding 71(3): 281–293, 1998.

[9] W. G. Kropatsch, *Building irregular pyramids by dual-graph contraction*, Vision, Image and Signal Processing, 142(6): 366–374, 1995.

[10] W. E. Lorensen and H. E. Cline, *Marching cubes: A high-resolution 3D surface construction algorithm*, Computer Graphics 21(4): 163–169, 1987.

[11] B. K. Natarajan, *On generating topologically consistent isosurfaces from uniform samples*, The Visual Computer, 11(1): 52–62, 1994.

[12] A. Rosenfeld, *Adjacency in digital pictures*, Information and Control 26(1): 24–33, 1974.

[13] R. Meyssonnier, *Régis Meyssonnier - France | LinkedIn[Proffesional Network]*, Website: http://fr.linkedin.com/pub/r%C3%A9gis-meyssonnier/41/77b/866/en