An Efficient Algorithm to Compute Subsets of Points in $\mathbb{Z}^n$

Ana Pacheco and Pedro Real

Dpto. Matemática Aplicada I, ETS Ingeniería Informática, Universidad de Sevilla
{ampm,real}@us.es

Abstract. In this paper we show a more efficient algorithm than that in [8] to compute subsets of points non-congruent by isometries. This algorithm can be used to reconstruct the object from the digital image. Both algorithms are compared, highlighting the improvements obtained in terms of CPU time.

Keywords: digital image, grid, hypercube, isometry, n–xel.

1 Introduction

An *n–dimensional digital image* is a data structure typically representing a grid made up by a finite set of n–dimensional color hypercubes. The n–dimensional hypercubes of the grid are called n–xels for digital images of dimension n; particularly, pixels for $n = 2$ and voxels for $n = 3$.

By considering the central point of each n–dimensional hypercube of the grid, we construct a dual grid made up by n–dimensional hypercubes whose vertices are the central points of the hypercubes of the original grid.

In this way, the n–xels of an image are identified with vertices of n–dimensional hypercubes of the dual grid.

In Figure 1 we show more details about this construction for 2–dimensional binary digital images.

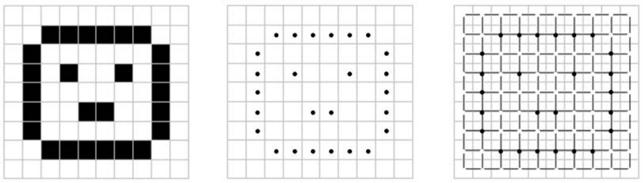

Fig. 1. From left to right: a binary digital image in a grid of size 10×9; central points of the image pixels; dual grid whose vertices are the central points of the squares of the original grid

In this sense, to represent images by using computational techniques it is necessary to fix a grid and the relations between the points.

Binary images are derived from a subdivision of the n–dimensional space into unit hypercubes of dimension n which intersect two by two in a hypercube of dimension $n-1$. This subdivided space is equivalent to use as grid the n–dimensional discrete space $\mathbb{Z}^n$. The elements $(i_1, ..., i_n) \in \mathbb{Z}^n$ are the lattice points. Once the grid has been established, it is necessary to fix the neighborhood relations between the lattice points.

For a given lattice point, a *neighborhood* is defined typically by using a distance metric (see [3]). More concretely, two lattice points in $\mathbb{Z}^n$ are *neighboring points* if they are less than *epsilon* distance away. Depending on the values of epsilon, different types of neighborhoods can be defined.

For instance, Kong and Roscoe [7] defined three standard types of neighborhood in the three-dimensional space $\mathbb{Z}^3$: the *6–neighborhood*, the *18–neighborhood* and the *26–neighborhood*. These definitions are essentially equivalent to the corresponding definitions in Rosenfeld [12]. In Figure 2 these three types of neighborhood in $\mathbb{Z}^3$ are shown.

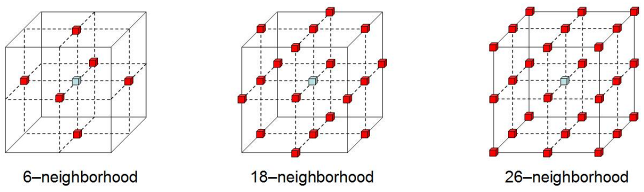

Fig. 2. A point $P \in \mathbb{Z}^3$ with each one of its: (a) six neighboring points satisfies $d_1(P,Q) = 1$; (b) eighteen neighboring points satisfies $d_1(P,Q) = 1$ or $d_1(P,Q) = 2$; and (c) twenty-six neighboring points satisfies $d_1(P,Q) = 1$, $d_1(P,Q) = 2$ or $d_1(P,Q) = 3$

For instance, a point $P \in \mathbb{Z}^4$ with each one of its: (a) eight neighboring points satisfies $d_1(P,Q) = 1$; (b) thirty-two neighboring points satisfies $d_1(P,Q) = 1$ or $d_1(P,Q) = 2$; (c) sixty-four neighboring points satisfies $d_1(P,Q) = 1$, $d_1(P,Q) = 2$ or $d_1(P,Q) = 3$; and (d) eighty neighboring points satisfies $d_1(P,Q) = 1$, $d_1(P,Q) = 2$, $d_1(P,Q) = 3$ or $d_1(P,Q) = 4$. See [5] for more details.

2 Preliminaries

In this section, we recall some basic notions about algebraic-topology, geometry, graph theory and digital images in order to do more understandable the paper.

Given a set S, an *order relation* on S is a relation $\preceq$ such that, for every $a, b, c \in S$ is held: (1) either $a \preceq b$, or $b \preceq a$; (2) if $a \preceq b$ and $b \preceq c$, then $a \preceq c$; (3) if $a \preceq b$ and $b \preceq a$, then $a = b$. Moreover, S is called *ordered set*. The *reverse order relation* $\succeq$ is the relation given by $a \succeq b$ if $b \preceq a$. Given two ordered sets S_1 and S_2, the *lexicographic order* on the Cartesian product $S_1 \times S_2$ is defined as $(a, b) \preceq (a', b')$ if and only if $a \prec a'$, or $a = a'$ and $b \preceq b'$.

A *distance* is a function $d : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ satisfying the following properties: (a) $d(x,y) \geq 0$; (b) $d(x,y) = 0$ if and only if $x = y$; (c) $d(x,y) = d(y,x)$ and (d) $d(x,z) \leq d(x,y) + d(y,z)$.

Some well-known distances are: (1) $d_1(x,y) = \sum_{i=1}^{n} |x_i - y_i|$; (2) $d_2(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$; and $d_\infty(x,y) = max_{i=1}^{n}\{|x_i - y_i|\}$.

An *n–polytope* is the closure of an n–cell with flat faces. Particularly, a *polygon* is a 2–polytope and a *polyhedron* is a 3–polytope.

An *n–dimensional hypercube* (or *hypercube of dimension* n) is an n–polytope of 2^n vertices which satisfy certain distance conditions. Particularly, 2–dimensional and 3–dimensional hypercubes are called *squares* and *cubes*, respectively.

A map $f : X \to Y$ is called *isometry* if for any $a, b \in X$ is satisfied $d(f(a), f(b)) = d(a,b)$. Two objects O, O' are called *isometric* (or *congruent by isometries*) if there exists a bijective isometry from O to O'.

The *group of isometries of a cube* are the rigid motions which leave the cube invariant. This group has 48 elements.

The *group of isometries of a 4–dimensional hypercube* are the rigid motions which leave the hypercube invariant. This group has 384 elements (see [10]).

A *graph* $G = (V(G), E(G))$ consists of two finite sets: $V(G)$, the *vertex set* of the graph, which is a nonempty set of elements called *vertices.* $E(G)$, the *edge set* of the graph, which is a possibly empty set of elements called *edges*, such that every edge $e \in E(G)$ is assigned an unordered pair of vertices $\{u, v\}$, $(u \neq v)$ called the *end-vertices* of e, and e is said to *join* u and v. If there exists more than one edge between each pair of vertices, the graph is called *multi-graph.*

A *subgraph* of a graph G is a graph having all its vertices and edges in G.

Two graphs G and G' are called *isomorphic graphs* if there exists an isomorphism (bijective morphism) between them.

Let G be a multi-graph of vertices $v_1, v_2, \ldots, v_n$. The *adjacency matrix* of G is a $n \times n$ matrix $M(G) = (m_{ij})$ where the element m_{ij} is given by the number of edges which join the vertex v_i to the vertex v_j.

A *n–dimensional digital image* is a representation of an image of dimension n as a finite set of digital values, called *picture elements* or *n–xels.* Particularly, these elements are called *pixels* and *voxels* for digital images of dimension 2 and 3, respectively. Moreover, if the set of digital values is $\{0,1\}$ then the image is called *binary digital image.*

3 Computing Subsets of Points in $\mathbb{Z}^n$

The first stage of this section consists in constructing subsets of points starting from the vertices of an n–dimensional unit hypercube. Then, the congruent ones by isometries of the n–dimensional space are ignored.

We assume that all the points in $\mathbb{Z}^n$ are assigned binary values, one or zero. The points whose value is 1 (resp. 0) are called 1–points (resp. 0–points). Given a finite subset of points, V, constructed starting from the vertices of an n–dimensional unit hypercube, we also assume that the points in V have a value of 1 while the points in the complement of V have a value of 0.

These subsets of points are determined as follows: the n–dimensional unit hypercube has 2^n vertices and each one of them can be a 1–point or a 0–point, so there exist 2^{2^n} subsets of points which can be constructed starting from the vertices of the n–dimensional unit hypercube. More concretely, there exist $C(2^n, c)$ subsets with $0 \leq c \leq 2^n$ 1–points. By using properties of combinatorial numbers, $C(2^n, 2^n - c) = C(2^n, c)$ is also the number of subsets with $2^n - c$ 1–points. In this way, the number of subsets with c 1–points is the same as the number of subsets with $2^n - c$ 1–points.

Below, we show the extension of the method shown in [8] to ignore congruent subsets that differ by isometries of the n–dimensional space. This method consisted in associating each subset with a multi-graph. The vertices of the multi-graph were the points of the subset and the number of edges between each pair of vertices u, v was determined by the square of Euclidean distance between u, v.

By considering the previous association between multi-graphs and subsets of points of the n–dimensional unit hypercube, it was natural to identify subsets with their respective associated multi-graphs.

A similar proof of Theorem 1 in [8] shows that two isomorphic subsets with at least 2^{n-1} points are isometric. The converse implication is obvious, taking into account that isometry is a stronger concept than isomorphism.

By considering previous results, Algorithm 1 in [8] (whose pseudocode is extended in Algorithm 3.1 for any dimension) was implemented.

Algorithm 3.1

```
Input: set of vertices of the $n$–dimensional unit hypercube with an order relation $\prec$.
// $V$: empty list to save the vertices of the non-isomorphic multi-graphs.
Output: non-congruent subsets by isometries of the $n$–dimensional space.
begin
  for $c = 2^{n-1}, ..., 2^n$ do
    Construct an ordered set $(V_c, \prec)$ containing to the $C(2^n, c)$ subsets with $c$ 1–points
    for $(V_c)_i \subset V_c$ do
      Determine the multi-graph associated with $(V_c)_i$, $(G_c)_i$, whose adjacency matrix is $M_{(G_c)_i} = ((m_{(G_c)_i})_{pq})$ where $(m_{(G_c)_i})_{pq} = a_{pq}$, being $a_{pq}$ the square Euclidean distance between $v_p, v_q$
      while $(V_c)_{i_1} \subset V_c$ & $(V_c)_{i_2} \subset V_c$ & $(V_c)_{i_1} \prec (V_c)_{i_2}$ do
        if $(G_c)_{i_1}$ and $(G_c)_{i_2}$ are isomorphic then
          $(V_c)_{i_1}$ and $(V_c)_{i_2}$ are congruent by isometries
          $V_c = V_c - \{(V_c)_{i_2}\}$
        end if
      end while
    end for
    $V = V \bigcup V_c$
  end for
  return $V$
end
```

By using as input the set of vertices of the n–dimensional unit hypercube arranged on an order relation $\prec$, for $2^{n-1} \leq c \leq 2^n$, this algorithm: (a) constructs an ordered set, $(V_c, \prec)$, containing to the $C(2^n, c)$ subsets with c 1–points; (b) associates each subset $(V_c)_i \subset V_c$ with a multi-graph $(G_c)_i$. The vertices of $(G_c)_i$ are the points of $(V_c)_i$ and the element a_{pq} of the adjacency matrix of $(G_c)_i$ corresponds with the square of Euclidean distance between v_p, v_q; and (c) checks if there exists an isomorphism between each pair of multi-graphs $(G_c)_{i_1}, (G_c)_{i_2}$, associated with two subsets $(V_c)_{i_1}, (V_c)_{i_2} \subset V_c$ which satisfy $(V_c)_{i_1} \prec (V_c)_{i_2}$.

Algorithm 3.1 allows us to ignore the subsets with $2^{n-1} \leq c \leq 2^n$ points obtained by isometries of the n–dimensional space. The subsets with $0 \leq c < 2^{n-1}$ points are determined by complementation.

Remark 1. Algorithm 3.1 only constructs subsets of vertices of the n–dimensional unit hypercube with at least 2^{n-1} points.

Remark 2. Given an order relation, $\prec$, on the vertices of the n–dimensional unit hypercube, Algorithm 3.1 determines the smallest non-congruent subsets with respect to $\prec$. Moreover, by changing the order relation, subsets congruent with these ones are obtained.

By using Algorithm 3.1, the following results can be proved.

Theorem 1. *In $\mathbb{Z}^3$, there exist (up to isometry): (a) six subsets with four vertices; (b) three subsets with five vertices; (c) three subsets with six vertices; (d) one subset with seven vertices; and (e) one subset with eight vertices.*

Taking into account the complementation, we can formulate Corollary 1.

Corollary 1. *In $\mathbb{Z}^3$, there exist (up to isometry): (b') three subsets with three vertices; (c') three subsets with two vertices; (d') one subset with one vertex; and (e') one subset with zero vertices.*

Remark 3. Let us observe that the twenty-two subsets obtained by using Algorithm 3.1 for $n = 3$ coincide (up to rotations of the 3–dimensional space) with the twenty-two types of unit cell presented by Kong and Roscoe in Figure 1 in [6].

Theorem 2. *In $\mathbb{Z}^4$, there exist (up to isometry): (a) seventy-four subsets with eight vertices; (b) fifty-six subsets with nine vertices; (c) fifty subsets with ten vertices; (d) twenty-seven subsets with eleven vertices; (e) nineteen subsets with twelve vertices; (f) six subsets with thirteen vertices; (g) four subsets with fourteen vertices; (h) one subset with fifteen vertices; and (i) one subset with sixteen vertices.*

Taking into account the complementation, we can formulate Corollary 2.

Corollary 2. *In $\mathbb{Z}^4$, there exist (up to isometry): (b') fifty-six subsets with seven vertices; (c') fifty subsets with six vertices; (d') twenty-seven subsets with five vertices; (e') nineteen subsets with four vertices; (f') six subsets with three vertices; (g') four subsets with two vertices; (h') one subset with one vertex; and (i') one subset with zero vertices.*

Remark 4. The results shown in Theorem 2 and Corollary 2 confirm Pólya's count in 1940 (see Table II in [9]), whose main difficulty to count the different 2–colorings of the 4–dimensional hypercube was the derivation of the appropriate cycle indices (see [2] for more details).

Algorithm 3.1 is based on graph isomorphisms, which is a problem in NP (see [1,4] for more details). For this reason, a more efficient algorithm to ignore the subsets of points that differ by isometries of the n–dimensional space has been implemented. This new algorithm computes the group of isometries of the n–dimensional unit hypercube in $\mathbb{R}^n$ and uses it to ignore the subsets of points of it that differ by isometries of the n–dimensional space. Proposition 7 in [13] proves that an algorithm of this type returns the subsets of points non-congruent that differ by isometries of the n–dimensional space. A scheme of this algorithm is shown in Figure 3.

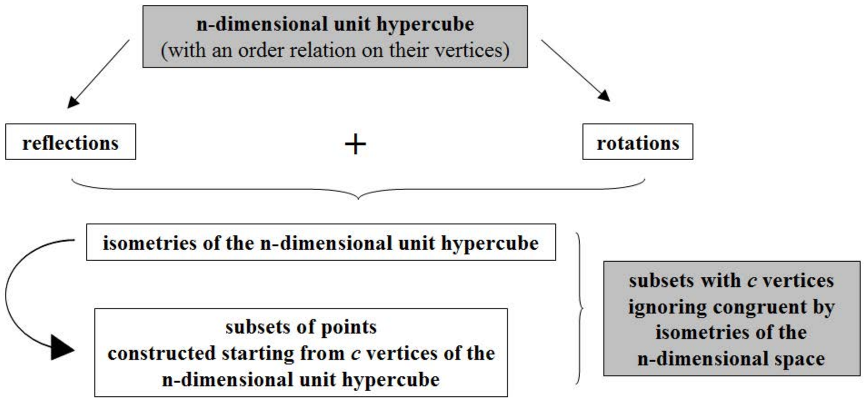
n-dimensional unit hypercube
(with an order relation on their vertices)
reflections
+
rotations
isometries of the n-dimensional unit hypercube
subsets of points
constructed starting from c vertices of the
n-dimensional unit hypercube
subsets with c vertices
ignoring congruent by
isometries of the
n-dimensional space

Fig. 3. Scheme of a more efficient algorithm than Algorithm 3.1 to ignore the subsets of vertices of the n–dimensional unit hypercube that differ by isometries of the n–dimensional space

Algorithm 3.2 ignores the subsets of points that differ by isometries from the group *iso_ncube* of isometries of the n–dimensional unit hypercube.

Algorithm 3.2

```
Input: $(V, \prec)$ set of vertices of the $n$–dimensional unit hypercube arranged on the order relation $\prec$.
        $iso\_ncube$: group of isometries of the $n$–dimensional unit hypercube.
// $NIS$: empty list to save the non-isometric subsets with $0 \leq c \leq 2^n$ points.
Output: non-isometric with $0 \leq c \leq 2^n$ points.
begin
  for $c = 0, ..., 2^n$ do
    Construct an ordered set $(V_c, \prec)$ containing to the $C(2^n, c)$ subsets with $c$ points
    for $(V_c)_i \subset V_c$ do
      $iso\_vci = \emptyset$
      {Empty list to save the subsets isometric to $(V_c)_i$.}
      for $\sigma \in iso\_ncube$ do
        $iso\_vci = iso\_vci \bigcup \sigma((V_c)_i)$
      end for
      for all $(V_c)_j \subset V_c$ satisfying $(V_c)_j \in iso\_vci$ do
        $(V_c)_i$ and $(V_c)_j$ are isometric
        $V_c = V_c - \{(V_c)_j\}$
      end for
    end for
    $NIS = NIS \bigcup V_c$
  end for
  return $NIS$
end
```

Firstly, Algorithm 3.2 constructs an ordered set $(V_c, \prec)$ containing to the $C(2^n, c)$ subsets with $0 \leq c \leq 2^n$ vertices of the n–dimensional unit hypercube. Next, for $(V_c)_i \subset V_c$, it computes all the subsets isometric to $(V_c)_i$ by applying the group of isometries of the n–dimensional unit hypercube. If there exists a subset $(V_c)_j \subset V_c$ which coincides with any of the subsets isometric to $(V_c)_i$, then the algorithm removes $(V_c)_j$ from (V_c). In this way, Algorithm 3.2 allows us to obtain a representative subset of each isometry class.

Remark 5. This alternative algorithm not only improves the computational time of Algorithm 3.1 (see Table 1 for results in $n = 3$ and Table 2 for $n = 4$) but it can be used for constructing non-isometric subsets of vertices of the n–dimensional unit hypercube, regardless of its cardinality.

Remark 6. In the same way as Algorithm 3.1, the new algorithm determines the smallest non-congruent subsets with respect to the order relation given on the set of vertices of the n–dimensional unit hypercube; so that, by changing the order relation, subsets congruent with these ones are obtained.

Remark 7. Theorems 1 and 2, and Corollaries 1 and 2 are held by using the algorithm whose scheme is shown in Figure 3.

Table 1. Algorithm 3.1 constructs and checks $C(8,4) = 70$, $C(8,5) = 56$, $C(8,6) = 28$ and $C(8,7) = 8$ multi-graphs in 3.12, 3.83, 2.8 and 1.58 seconds of CPU time; respectively. These results are improved to 0.12, (practically) 0, (practically) 0 and (practically) 0 seconds of CPU time; respectively.

	Algorithm 3.1	**Alternative algorithm**
***c* = 4**	3.12 sec.	0.12 sec.
***c* = 5**	3.83 sec.	–
***c* = 6**	2.8 sec.	–
***c* = 7**	1.58 sec.	–

Table 2. Algorithm 3.1 constructs and checks $C(16,8) = 12870$, $C(16,9) = 11440$, $C(16,10) = 8008$, $C(16,11) = 4368$, $C(16,12) = 1820$, $C(16,13) = 560$, $C(16,14) = 120$ and $C(16,15) = 16$ multi-graphs in 11092.3, 9482.27, 11652.5, 8382.22, 2384.01, 734.34, 175.04 and 25.786 seconds of CPU time; respectively. These results are improved to 18.8, 18.56, 14.63, 10.23, 5.43, 2.76, 1.55 and 1.73 seconds of CPU time; respectively.

	Algorithm 3.1	**Alternative algorithm**
***c* = 8**	11092.3 sec.	18.8 sec.
***c* = 9**	9482.27 sec.	18.56 sec.
***c* = 10**	11652.5 sec.	14.63 sec.
***c* = 11**	8382.22 sec.	10.23 sec.
***c* = 12**	2384.01 sec.	5.43 sec.
***c* = 13**	734.34 sec.	2.76 sec.
***c* = 14**	175.04 sec.	1.55 sec.
***c* = 15**	25.78 sec.	1.73 sec.

4 Conclusion and Examples

In this paper, we have shown an algorithm more efficient than the extension of that in [8] to compute subsets of points non-congruent by isometries of the n–dimensional space. By using this algorithm for $n = 3$ and $n = 4$, 22 and 402 non-isometric subsets (see Figures 4 and 5 for some examples of these subsets), respectively, have been computed by using a low CPU time. This algorithm allows us to reconstruct objects from the n–xels of n–dimensional binary digital images (see Figure 6 for 2 and 3–dimensional examples).

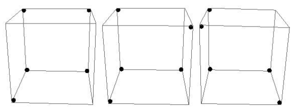

Fig. 4. Non-isometric subsets with five points of $\mathbb{Z}^3$ computed by using Algorithm 3.1 and its alternative using 3.83 and 0 seconds of CPU time, respectively

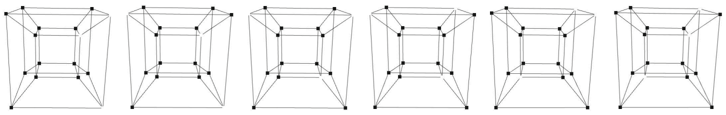

Fig. 5. Non-isometric subsets with thirteen points of $\mathbb{Z}^4$ computed by using Algorithm 3.1 and its alternative using 734.34 and 2.76 seconds of CPU time, respectively

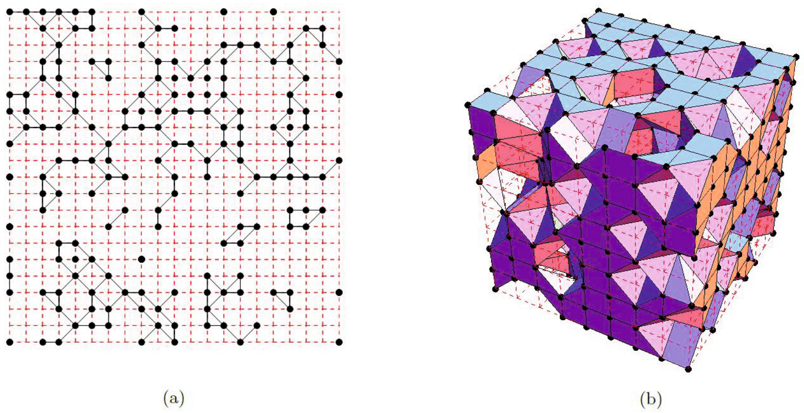

Fig. 6. Reconstruction of an object from (a) the 172 pixels (points) of the digital image localized on a grid of size 20×20, (b) the 495 voxels (points) of the digital image localized on a grid of size $8 \times 8 \times 8$

Remark 8. Let us note that the pictures in Figure 6 represent a 2–dimensional (resp. 3–dimensional) object by extracting the boundary of the cell complex whose cells are constructed from the convex hull of the black vertices of each square (resp. cube). Moreover, this technique to represent objects from the boundary of cell complexes can be extended to higher dimensions.

References

1. Arias-Fisteus, J., Fernández-García, N., Sánchez-Fernández, L., Delgado-Kloos, C.: Hashing and canonicalizing Notation 3 graphs. Journal of Computer and System Sciences 76(7), 663–685 (2010)
2. Banks, D.C., Linton, S.A., Stockmeyer, P.K.: Counting Cases in Substitope Algorithms. IEEE Transactions on Visualization and Computer Graphics 10(4), 371–384 (2004)
3. Dörksen-Reiter, H.: Shape representations of digital sets based on convexity properties. Dissertation, Universität Hamburg (2005)
4. Gross, J.L., Yellen, J.: Handbook of Graph Theory. CRC Press, Boca Raton (2003)
5. Han, S.E.: A generalized digital (k_0, k_1)-homeomorphism. Note di Matematica 22(2), 157–166 (2003)

6. Kong, T.Y., Roscoe, A.W.: A theory of binary digital pictures. Computer Vision, Graphics, and Image Processing 32(2), 221–243 (1985)
7. Kong, T.Y., Roscoe, A.W.: Continuous Analogs of Axiomatized Digital Surfaces. Computer Vision, Graphics, and Image Processing 29(1), 60–86 (1985)
8. Pacheco, A., Mari, J.-L., Real, P.: Obtaining cell complexes associated to four dimensional digital objects. Imagen-a 1(2), 57–64 (2010)
9. Pólya, G.: Sur Les Types Des Propositions Composées. The Journal of Symbolic Logic 5(3), 98–103 (1940)
10. Rodrigues Costa, S., Gerônimo, J.R., Palazzo Jr, R., Carmelo Interlando, J., Muniz Silva Alves, M.: The Symmetry Group of $\mathbb{Z}_q^n$ in the Lee Space and the $\mathbb{Z}_{q^n}$-Linear Codes. In: Mattson, H.F., Mora, T. (eds.) AAECC 1997. LNCS, vol. 1255, pp. 66–77. Springer, Heidelberg (1997)
11. Rosenfeld, A.: Connectivity in Digital Pictures. Journal of The ACM - JACM 17(1), 146–160 (1970)
12. Rosenfeld, A.: Three-dimensional digital topology. Information and Control 50(2), 119–127 (1981)
13. Ziegler, G.M.: Lectures on 0/1-polytopes. Oberwolfach Seminars 29, 1–41 (2000)