Algebraic tools in geometrical modeling with topological control

Ana Pacheco[1] and Pedro Real[1]

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

Abstract. In this paper we show a method to create a look-up table with all the possible bricks which can made up an object. Moreover, we generate a procedure to recreate any object starting from this table. More concretely, the procedure is divided into two stages: (1) choosing the bricks of the table which made up the given object; and (2) merging these bricks to recreate the object.

Keywords: digital image; grid; object; pixel.

1 Introduction

Digital images are numerical representations of objects. It means an effective way of storing and communicating visual information, although the images usually condensate or summary the information of the objects that they represent. A *digital image* is defined as a data structure representing a grid made up by a finite set of color squares. The squares of the grid are called *pixels*. These pixels are considered as cells of dimension two (2–cells) made up four vertices (0–cells) and four edges (1–cells).

Considering the central point of each pixel, we construct a dual grid made up by squares whose vertices are the central points of the squares of the primal grid. In this way, the pixels of an image are identified with the vertices of the squares of the dual grid.

In this sense, to represent images 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 plane into unit squares which intersect two by two in an edge. This subdivided plane is equivalent to use as grid the discrete plane $\mathbb{Z}^2$. The elements $(x, y) \in \mathbb{Z}^2$ 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 using a distance metric (see [2]). More concretely, two lattice points in $\mathbb{Z}^2$ are *neighboring points* if they are less than *epsilon* distance away. Depending on the values of epsilon, different types of neighborhoods can be defined. A point $P \in \mathbb{Z}^2$ with each one of its: (a) four neighboring points satisfies $d_1(P, Q) = 1$; (b) eight neighboring points satisfies $d_1(P, Q) = 1$ or $d_1(P, Q) = 2$.

The structure of the paper is as follows: in Section 2, we create a look-up table with all the bricks which can made up an object; in Section 3, given a digital image placed on a dual grid, we chose of the look-up table the bricks which made up the object represented by the image; in Section 4, we merge the bricks obtained in Section 3 in order to recreate the object; finally, in Section 5, we expose some conclusions and examples.

2 Construction of the look-up table

In this section we present a method to construct all the bricks which can made up an object. The construction is made in two steps: (a) firstly, we obtain (using combinatorial techniques) the vertices (0–cells) of these bricks; and then (b) we compute the i–cells ($0 < i \leq 2$) of these bricks from its 0–cells. The bricks are saved on a look-up table shown at the end of the section.

2.1 Vertices of the bricks

We can set down that the 0–cells of the bricks coincide with the subsets of points constructed from the vertices of a unit square, since the digital image which represent the object is placed on a dual grid made up by squares. In this sense, we assume that all the points in $\mathbb{Z}^2$ have 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 from the vertices of a unit square, 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 unit square has 4 vertices and each one of them can be a 1–point or a 0–point, so there exist $2^4 = 16$ subsets of points which can be constructed from the vertices of the unit square. More concretely, there exist $C(4, c)$ subsets with $0 \leq c \leq 4$ 1–points. On the left in Table 1, we show the 16 subsets of points which can be constructed from the vertices of the unit square. These subsets coincide with the 0–cells of the bricks which can made up an object.

2.2 Cells of the bricks

The i–cells ($0 < i \leq 2$) of the bricks are computed deforming the square which contain to the vertices of each brick. We can observe that the elements on the positions (1,1), (2,1), (3,1), (3,3), (4,1) and (5,1) on the left in Table 1 can be chosen as pattern squares, since they allow us to determine (by using rotations) the remain of elements of the table. In this way, it suffices to deform these six pattern squares. The deformation of any another element of the table is obtained by rotation of one of these six pattern squares.

By another hand, the deformation of each one of the pattern squares is obtained defining operators from each 0–point of the pattern square to the edge of the pattern square whose end-points are the 0–point and its small (using lexicographic order) 4–neighboring point. These operators can lead a double edge, which must be removed defining a new operator. The new operator sends the double edge to the square. See Figure 1 for an example.

The previous operators, called *integral operators*, are defined as follows.

Definition 1. *An* integral operator *is a linear map denoted by ϕ and defined from a k–cell a to a $(k+1)$–cell b satisfying: (1) $\phi(a) = b$, where b is a $(k+1)$–cell bounded by the k–cell a, and (2) $\phi(c) = 0$ for any k–cell c different from a.*

Fig. 1. (a) Defining operators from the 0–points of the pattern square to the edges of the pattern square can lead a double edge. (b) This edge is removed defining a new operator, which sends the double edge to the square.

An integral operator can be seen as an elementary algebraic deformation of a $(k+1)$–cell into a k–cell which bounds the $(k+1)$–cell. A pictorial description of an integral operator applied to a k–cell a which bounds a $(k+1)$–cell b can be represented by an arrow over b starting from a.

On the left in Table 1, we show the integral operators which deform the 16 subsets of vertices. These operators allow us to obtain the i–cells $(0 < i \leq 2)$ of the bricks, which are shown on the right in Table 1.

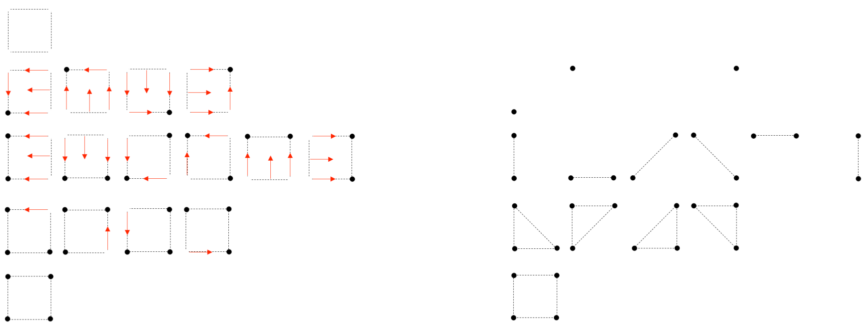

Table 1. Integral operators deforming the squares to obtain the bricks.

3 Recreating the object

In this section, we show the procedure to recreate the object from the pixels of a given digital image.

Let I be a digital image placed on a dual grid of size $n \times m$ and made up by $p_1, ..., p_r$ coordinated pixels. The first step on the recreation of the object consists on localizing the squares of the grid which contain to the pixel p_i, for $i = 1, ..., r$. This step allows us to regroup the pixels by squares, in such way as each group of pixels on a square of the grid coincides with one of the subsets of vertices shown on the left in Table 1. See Figure 2 (a) for an example. The vertices of the bricks which made up the object are determined in this step.

The second step on the recreation consists on determining the i–cells $(0 < i \leq 2)$ of these bricks from their vertices. Each subset of vertices on the left in Table 1 has associated a brick on the right in Table 1. In this sense, it suffices to associate each group of pixels of the image (see previous step) with its corresponding brick. In Figure 2 (b), we show the bricks associated with each one of the groups of pixels of the image shown in Figure 2 (a).

Finally, the bricks which made up the object are merged. It is made removing the common vertices between the co-lineal edges which belong to adjacent squares of the grid and removing the common edges between the co-planar faces which belong to adjacent squares of the grid. In Figure 2 (c), we recreate the object obtained from the image shown in Figure 2 (a).

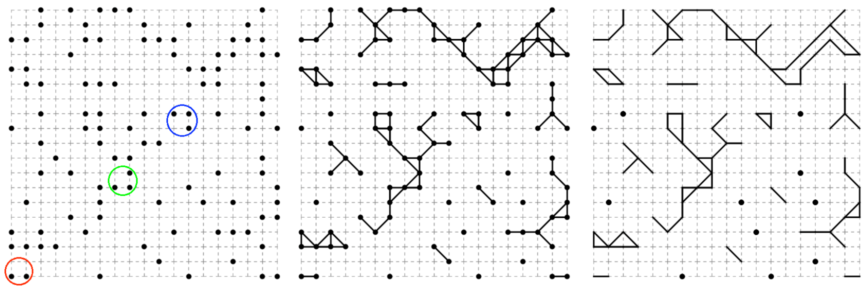

Fig. 2. (a) Digital image placed on a dual grid of size 18×18 and made up by 105 coordinated pixels. The group of pixels placed on the square circled in red, green and blue coincides with the element on the position (3,2), (4,3), and (4,4), respectively, on the left in Table 1. (b) Bricks associated with each one of the groups of pixels of the image shown in (a). (c) Object recreated from the image shown in (a).

References

1. P. Bhaniramka, R. Wenger and R. Crawfis. Isosurface construction in any dimension using convex hulls. IEEE Transactions on Visualization and Computer Graphics 10(2): 130–141, 2004.
2. H. Dörksen-Reiter. Shape representations of digital sets based on convexity properties. Dissertation, Universität Hamburg, 2005.
3. T. Y. Kong and A. W. Roscoe. A theory of binary digital pictures. Computer Vision, Graphics, and Image Processing 32(2): 221–243, 1985.
4. V. A. Kovalevsky. Finite Topology as Applied to Image Analysis. Computer Vision, Graphics, and Image Processing 46, 141–161, 1989.
5. W. E. Lorensen and H. E. Cline. Marching cubes: A high-resolution $3D$ surface construction algorithm. Computer Graphics 21(4): 163–169, 1988.
6. A. Rosenfeld. Connectivity in Digital Pictures. Journal of The ACM - JACM 17(1): 146–160, 1970.
7. G. M. Ziegler. Lectures on 0/1-polytopes. Oberwolfach Seminars 29: 1–41, 2000.