A Framework for Digital Topology

E. Domínguez
Dpto. Ing. Eléctrica e Informática
Facultad de Ciencias. U. de Zaragoza
E-50009 Zaragoza (SPAIN)

A.R. Francés
Dpto. Ing. Eléctrica e Informática
Facultad de Ciencias. U. de Zaragoza
E-50009 Zaragoza (SPAIN)

A. Márquez
Dpto. de Algebra, Geometría, Topología y Computación
Facultad de Matemáticas. U. de Sevilla
Sevilla (SPAIN)

Abstract **The main goal of this paper is to show the functional architecture of a framework for Digital Topology. This architecture has four levels, called Device, Logical, Conceptual and Continuous Levels. In each one of them we can use several models according to the particular problem. The models in the Device Level represent the physical problem whereas the models in the Continuous Level are topological spaces which allow us to use the well-known results of continuous topology (actually, the stronger results of polyhedral topology). The other two levels are used to find a digital solution. The Logical Level is closer to the Device Level and it is used for processing, for writing algorithms and showing their correctness. The Conceptual Level is the nearest to the Continuous Level and it is used to translate results and notions from the Continuous Level to the Logical Level.**

I. INTRODUCTION

The main purpose of Digital Topology is the study of topological properties of discrete objects which are gotten digitizing continuous objects. Digital Topology plays a very important role in computer vision, image processing and computer graphics. But a complete theoretical foundation of a consistent theory for digital spaces is still missing. Kong and Rosenfeld give in [5] a very good survey about this subject.

However, several theories have been devised for the analysis of the topological attributes of a digital image. Let us recall, for example, Rosenfeld's combinatorial theory [9,10,11] (generalized by Kong and Roscoe in [4]) and Khalimsky's theory based in a particular topological space [3]. The common drawback of these theories is that, in order to include well-known results of the Euclidean Topology, they need to rewrite new proofs instead of exploiting those coming from continuous topology. The reason is that these theories are far from the Euclidean plane (or space). Some other authors (Kovalevsky [7], Ankeney, Ritter [1]) can use results coming from Euclidean Topology but they have problems in the image processing because the models are far from the discrete objects which represent screen digital images.

In this paper we introduce a new point of view. We present a framework, which tries to define a general theory for the development of Digital Topology. To do this, our framework proposes a multilevel architecture whose main feature is the ability for translating concepts, statements, proofs and algorithms from continuous topology without rewriting a parallel theory. The first level represents a computer and the following levels consist of models more and more abstract. Finally, the last level represents the Euclidean Topology. This takes us, from the discrete world (a computer screen), and brings us closer to the continuous one (the Euclidean plane or space).

In our framework there is not an universal model which can be used for solving all the problems in Digital Topology. Instead, given a problem we must choose a suitable model in every level. According to our proposal, what is common for all the problems is the working methodology and the multilevel functional architecture.

To show that our theory works we present the solution to the well-known Digital Jordan Curve Problem (as many other authors have done in order to prove the

Manuscript received July 1, 1993. This work was supported in part by the Diputación General de Aragón, the DGICYT and the Junta de Andalucía, Spain.

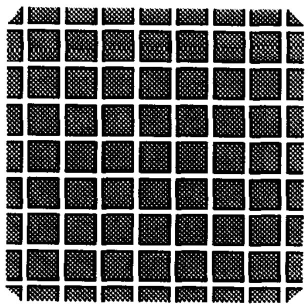


Figure 1: Screen model

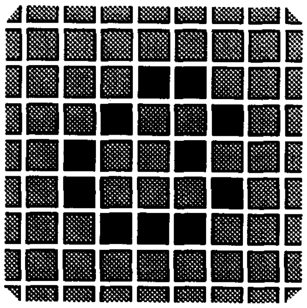


Figure 2: A digital image

consistency of their theories). So, in the second section we choose the models for each level in the architecture, and then, in the third section, we give the proof of the result that is reduced to giving the appropriate notions and translations. Finally, the forth section is devoted to explain the functional architecture in a general context.

II. THE MATHEMATICAL MODELS

We consider that the screen model is an infinite matrix S of pixels with the shape showed in Fig 1 where each pixel can have two states represented by a dotted or black small square. In this context, a digital image in the screen model S is defined by a set of black points (see a digital curve in Fig 2).

The Digital Jordan Curve Problem consists of proving that a simple closed digital curve, as that of Fig 2, divides the screen in two connected components. Evidently it is necessary to define the meaning of "simple closed digital curve" and "connected components." For this, we will start by defining the mathematical models used in each level of our framework.

Since our problems has a topological nature, it is natural to consider a transformation from the screen model S to the graph E_8 represented in the Fig 3. For the sake of simplicity we will suppose that the vertex

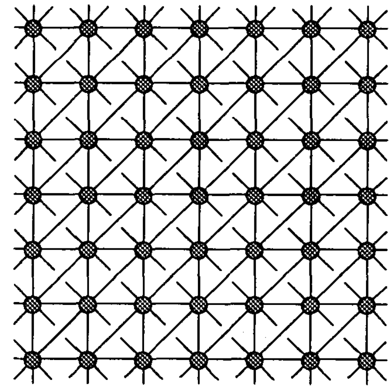


Figure 3: The graph E_8

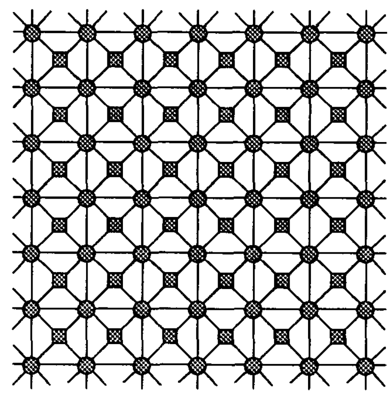


Figure 4: The graph E_8^*

set of E_8 is $\mathbf{Z}^2$, which is the set of all pairs of integer numbers. The vertices of the graph represent the pixels and two vertices are adjacent if and only if their corresponding pixels are contiguous in the obvious sense. In order to solve our problem, this mathematical model represents the Logical Level. In it the curve becomes the subgraph induced by the vertices corresponding to black pixels.

Because E_8 is not planar, it is well-known that within it we cannot represent the topology of the Euclidean plane (see Rosenfeld [9]). To solve that problem we flatten out this graph in a natural way and we get the planar graph E_8^* represented in the Fig 4. In this graph there are two different kinds of vertices. Some of them represent the pixels and the others, called middle points, represent a degree of nearness between the corresponding pixels; in fact it is the diagonal nearness. Observe that this graph is a triangulation of the Euclidean plane. This makes up the Conceptual Level to

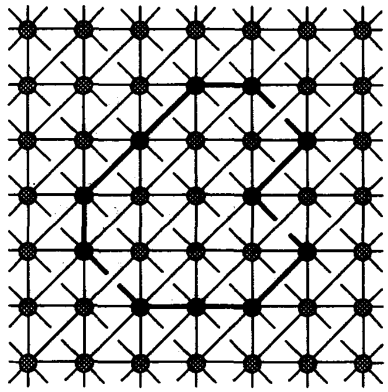


Figure 5: A digital curve C in E_8

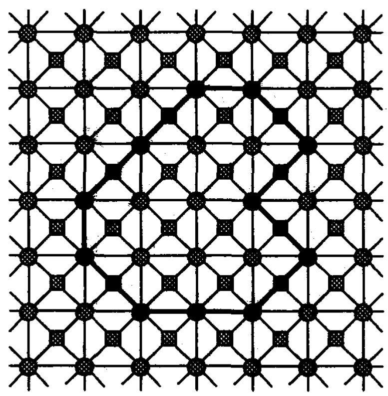


Figure 6: A digital curve C^* in E_8

solve our problem.

On the other hand, we define a digital image (or digital subspace) of one of these graphs as an induced subgraph; that is, a subgraph which contains an edge if and only if it contains the two vertices of the edge. We represent the set of digital images in E_8 and E_8^* by $\mathcal{O}(E_8)$ and $\mathcal{O}(E_8^*)$, respectively.

Now we have a natural transformation $\pi : \mathcal{O}(E_8) \longrightarrow \mathcal{O}(E_8^*)$ defined as follows: Given a digital image C in E_8, $\pi(C) = C^*$ is the subgraph induced by the vertices in C and the middle vertices defined by two diagonal vertices in C (see the Fig 5 and 6). In a natural way we have a transformation $\pi^* : \mathcal{O}(E_8^*) \longrightarrow \mathcal{O}(E_8)$. Given a digital image C^* in E_8^*, $\pi^*(C^*) = C$ is the subgraph induced by the vertices in C^* that are not middle vertices. Also we have a transformation $j : \mathcal{O}(E_8^*) \longrightarrow \mathcal{L}(\mathbf{R}^2)$ induced by the embedding of E_8^* in the Euclidean plane $\mathbf{R}^2$, where $\mathcal{L}(\mathbf{R}^2)$ is the set of polygonal subspaces.

In this way, we have the architecture represented by the following diagram

$$\begin{array}{l} \mathcal{O}(E_8) \xrightarrow{i} \mathcal{O}(S) \\ \pi \uparrow\downarrow \pi^* \\ \mathcal{O}(E_8^*) \xrightarrow{j} \mathcal{L}(\mathbf{R}^2) \end{array}$$

where $\mathcal{O}(S)$ is the set of digital images in the screen model S and i is the 1-1 transformation between $\mathcal{O}(E_8)$ and $\mathcal{O}(S)$.

III. THE DIGITAL JORDAN CURVE THEOREM

In the logical and conceptual levels, a simple digital curve is the subgraph induced by a sequence of vertices $\{p_0, \ldots, p_n\}$ so that p_i is adjacent to p_j if and only if $|i - j| \leq 1$. The curve is called closed if, in addition, $p_0 = p_n$. Then a digital object in the screen model S is called a simple closed digital curve if its image by i^{-1} is this type of curve in E_8. These definitions agree with those usually adopted in the literature (see [5]).

In this way, a simple closed digital curve C in the screen model S is, by definition, transformed through i in such a curve in E_8, denoted also by C. It is obvious that the image by π of a simple closed digital curve C in E_8 is a simple closed digital curve C^* in E_8^*. And, also, the image by the embedding j of C^* is a polygonal Jordan Curve C' in $\mathbf{R}^2$ (see Fig 5, 6). In this way, we have translated our initial Jordan Curve Problem for the Screen Model to analogous problems in E_8, E_8^* and $\mathbf{R}^2$. Now then, it is well-known that the solution to this problem in $\mathbf{R}^2$ is the Polygonal Jordan Curve Theorem. So that, our goal is to translate this theorem, by mean of the transformations j and π^*, to E_8 in order to find a solution to our problem in this model.

In the literature, there exists several equivalent statements for the Polygonal Jordan Curve Theorem. Here, we consider one of them that is appropriated to find an algorithm solving this problem and to prove its correctness. The base of this statement is the notion of transversal intersection between a half-line, which is parallel to the axis OX, and a polygonal curve.

Let D be a polygonal curve in $\mathbf{R}^2$ whose set of vertices $\{p_0, \ldots, p_n\}$ is counter-clockwise ordered. Let $r_q = \{(x, y); y = y_q \text{ and } x \geq x_q\}$ be a half-line, where $q = (x_q, y_q)$. There exists a ***transversal intersection*** between D and r_q if one of the following situations occurs:

(a) r_q intersects the edge defined by the vertices p_i and p_{i+1} in only a point $p \notin \{p_i, p_{i+1}\}$.

(b) there exists $i \in \{0, \ldots, n\}$ such that for some $k \geq 0$

1. $y_i = y_{i+1} = \cdots = y_{i+k} = y_q$

2. $y_{i-1} > y_q$ and $y_{i+k+1} < y_q$ or $y_{i-1} < y_q$ and $y_{i+k+1} > y_q$
3. $x_j \geq x_q$ for every $i \leq j \leq i+k$.

With this definition we can state the next well-known result.

Polygonal Jordan Curve Theorem. Let D be a simple closed polygonal curve in $\mathbf{R}^2$, then $\mathbf{R}^2 \setminus D$ has two connected components (one of them bounded and the other one unbounded). Moreover, a point $q \in \mathbf{R}^2 \setminus D$ belongs to the bounded component if and only if $\#(r_q \pitchfork D) \equiv 1 \pmod 2$, where $\#(r_q \pitchfork D)$ represents the number of transversal intersections between D and r_q.

It is important to point out that the result we want to translate to E_8 is applied to polygonal curves C' coming from a digital curve C in E_8 and half-lines $r_{p'}$, where p' belongs to $\mathbf{Z}^2$. In this way, it is easy to observe that the transversal intersections between one of such a half-line $r_{p'}$ and one of such a curve C' never occurs in case (a) of the definition above. That is, this kind of intersection always has a vertex of the curve. So that, we can translate the notion of transversal intersection to E_8 and E_8^* in such a way that

$$\#(r_{p'} \pitchfork C') = \#(r_{p^*} \pitchfork C^*) = \#(r_p \pitchfork C)$$

On the other hand, in the Conceptual Level, represented by E_8^*, we can consider the natural notion of connection induced by the graph structure. This notion coincides with the notion of connection induced by the topology of the Euclidean plane through the embedding j. So we can consider the connected components of $E_8^* \setminus C^*$. Since E_8^* is a triangulation of $\mathbf{R}^2$ it is not difficult to prove that the number of connected components of $E_8^* \setminus C^*$ and $\mathbf{R}^2 \setminus C'$ agree; even more, each component K^* of $E_8^* \setminus C^*$ is the finite triangulation of a component K' of $\mathbf{R}^2 \setminus C'$ in the following sense:

1. Given K^* there is one and only one component K' of $\mathbf{R}^2 \setminus C'$ such that $K^* \subset K'$.
2. K^* is induced by the vertices of E_8^* in K'.

These properties show us that the components of $E_8^* \setminus C^*$ represent the components of $\mathbf{R}^2 \setminus C'$.

Now we can translate the Jordan Curve Theorem to E_8^* in the following way.

Jordan Curve Theorem in $\mathbf{E}_8^*$. Let C^* be a simple closed digital curve in E_8^*, then $E_8^* \setminus C^*$ has two connected components (one of them bounded and the other one unbounded). Moreover, a point $p^* \in E_8^* \setminus C^*$ belongs to the bounded component if and only if $\#(r_{p^*} \pitchfork C^*) \equiv 1 \pmod 2$.

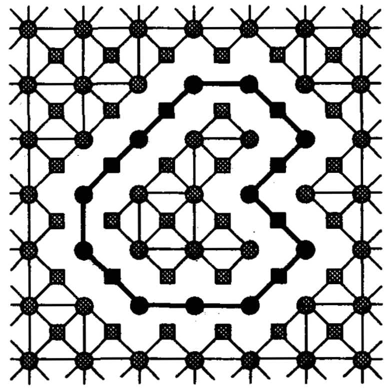


Figure 7: Components of $E_8^* \setminus C^*$

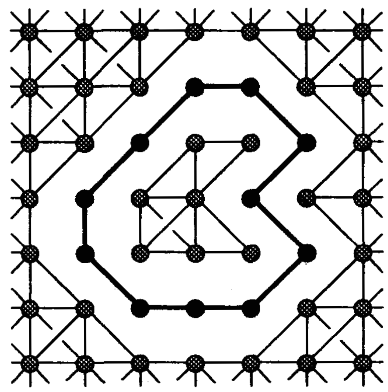


Figure 8: Top-components of $E_8 \setminus C$

Finally, we need to consider an appropriate notion of component in E_8. Let C be a simple closed digital curve in E_8. We call a *top-component* of $E_8 \setminus C$ to the image by π^* of a connected component of $E_8^* \setminus C^*$. Observe that, in general, a top-component of $E_8 \setminus C$ does not coincide with a connected component of $E_8 \setminus C$ (see Fig 7, 8).

In this way we have proved

Jordan Curve Theorem in $\mathbf{E}_8$. Let C be a simple closed digital curve in E_8, then $E_8 \setminus C$ has two connected top-components (one of them bounded and the other one unbounded). Moreover, a point $p \in E_8 \setminus C$ belongs to the bounded top-component if and only if $\#(r_p \pitchfork C) \equiv 1 \pmod 2$.

Observe that the previous proof not only proves the given problem but also allows to translate the well-known algorithm of Preparata [8] (coming from Computational Geometry) to solve the digital curve inclusion problem.

IV. THE FUNCTIONAL ARCHITECTURE

The previous sections contain a particular instance of the methodology proposed in our framework. In this section, we will present the general functional architecture of this framework. Our framework has four levels, called Device, Logical, Conceptual and Continuous Levels.

In the Device Level we represent the objects in a computer screen (typically a digital image). This level has a very small degree of abstraction and we only represent the physical aspects of the objects.

A second level of abstraction is obtained in the Logical Level. We consider in it the aspects of proximity of the objects so, we can study some properties of topological nature. The main function of this level is to be the support for writing the algorithms and to prove their correctness.

In general, the level above is far from the mathematical model in which we have a solution for our problem. So we need the Conceptual Level as an interface between the level above and the Continuous Level. To realize this interface it is necessary to translate: (1) Objects and properties from the Logical Level to the Conceptual Level and vice versa; (2) Objects and properties from the Conceptual Level to the Continuous Level; (3) Properties of objects in the Continuous Level to properties of objects in the Conceptual Level.

Finally, the Continuous Level is used to find a continuous solution. Observe that, actually, the objects and concepts obtained from the Logical Level are inside the Polyhedral Topology rather than the Continuous Topology and so we can use the more powerful tools of this field. The objects of our physical problem have been translated by consecutive abstractions from the Device Level. Now we must find a continuous solution in this level by using the well-known results of Polyhedral Topology and we translate it to the Logical Level across the Conceptual Level.

When we have a concrete problem and a particular screen model we must choose specific models in each level and functions which can support the functionality that we have described. Specifically, suppose that these chosen models are D, L, C and S for the Device, Logical, Conceptual and Continuous Level, respectively. Let $\mathcal{O}(D)$, $\mathcal{O}(L)$, $\mathcal{O}(C)$ and $\mathcal{O}(S)$ be the sets of the objects (i.e., substructures in some mathematical sense) of these models. So we have the following functional architecture

$$\begin{array}{c} \mathcal{O}(L) \xrightarrow{i} \mathcal{O}(D) \\ \pi \uparrow\downarrow \pi^* \\ \mathcal{O}(C) \xrightarrow{j} \mathcal{O}(S) \end{array}$$

We represent our physical objects in the model D and we translate it to the model L by the function i. If we have in L enough knowledge to solve the problem we do not need to use the rest of the models; when we have the solution, we interpret it in D by the function i. Otherwise, we translate the objects to the model C. If we can find a solution in it we translate it to the model L by the function π^*. But if even in C we cannot find a solution, we translate the objects to the model S where we can apply all of the quite powerful tools and results of Polyhedral Topology and, if we find a solution, we translate it to L by the functions j and π^*.

From a theoretical point of view, the particular structures that we need in the levels depend on the problem we want to solve. But, in general, there is a basic structure for a wide range of problems. For example, the basic structure for the planar Digital Topology is the one shown in the paragraphs above.

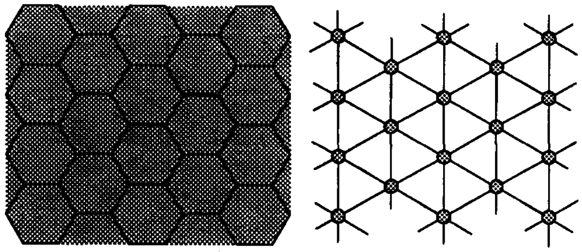
In addition, this framework can be used to solve problems which have not been proposed up till now in Digital Topology. An important example is the digital Shoenflies theorem which states that a simple closed digital Jordan curve surrounds a digital disk. The solution to this problem is well-known in Planar Euclidean Topology. Thus, our multilevel methodology can be applied (choosing suitable models) to obtain the corresponding digital version.

Moreover, this framework also works in higher dimensions. As an example, the proof given for the digital Jordan curve theorem can easily be adapted to solve the corresponding 3-dimensional problem (compare this solution with [6], where Kopperman et al. rewrite a new proof of this result).

V. FINAL REMARKS

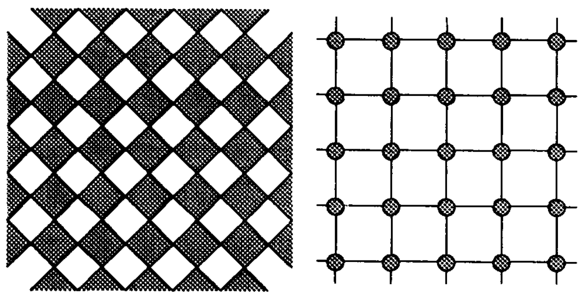
In this paper, we have developed a general framework that allows to use very powerful tools and results (those of Polyhedral Topology) in Digital Topology. The models used in the architecture depend on the Screen Model. For example, if our Screen Model is represented by the Fig 9(a), the graph used as Logical Model is the one in the Fig 9(b) (called hexagonal graph). In this case, each pair of cells has the same connectivity degree and the graph is planar, so we choose the same graph to represent the Conceptual Level. Observe that, in this case, this model verifies the Jordan Curve Theorem. The proof is the same as in the case of the graph of the 8-adjacencies.

There are models which do not verify the Jordan Curve Theorem. An example of this is the graph E_4, represented by the Fig 10(b), which is the logical model of the screen model represented by the Fig 10(a). This graph is planar, so we must consider the same graph



(a) (b)

Figure 9: (a) Screen model; (b) The graph E_6



(a) (b)

Figure 10: (a) Screen model; (b) The graph E_4

in the Conceptual Level. Thus the top-connection is equivalent to the 4-connection. Now, the vertices of a minimal cycle on this graph define a closed simple digital curve but its complement is connected.

Very frequently, some authors have shown a proof of the Digital Jordan Curve Theorem in order to prove the consistency of their theories. For this reason we also have chosen it for our framework. In the literature there are several proofs of this theorem using different techniques. The first author who gave a proof was Rosenfeld who presented two versions in a series of papers ([9,10,12]). One is taking an 8-curve (i.e., a curve in the graph E_8) and proving that its complement has two 4-connected components (i.e., connected by arcs in the graph E_4 of the 4-adjacencies). The other is taking a 4-curve (i.e., a curve in the graph E_4) and proving that its complement has two 8-connected components (i.e., connected by arcs in E_8).

It is easy to observe that the theorem presented in this paper includes both versions. This is a direct consequence of the following property. Given a closed digital curve C in E_8, then:

1. If C is an 8-curve, the top-components of $E_8 \setminus C$ coincide with the 4-connected components.
2. If C is an 4-curve, the top-components of $E_8 \setminus C$ coincide with the 8-connected components.

ACKNOWLEDGEMENTS

We would like to thank Julio Rubio for his very useful comments on an earlier draft of this paper.

REFERENCES

[1] L.A. Ankeney and G.H. Ritter, Cellular Topology and its Applications in Image Processing, *Int. Journ. Compt. Inf. Sciences* 12, 1983, 433-455.

[2] B. Bollobás, *Graph Theory: an introductory course*, Graduate Texts in Math., vol. 63, Springer-Verlag, 1979.

[3] E. Khalimsky, R. Kopperman, P.R. Meyer, Computer graphics and Connected Topologies on finite ordered sets, *Topology and Appl.* 36, 1990, 1-17.

[4] T.Y. Kong, A.W. Roscoe, A Theory of Binary Digital Pictures, *Comput. Vision Graphics Image Process.* 32, 1985, 221-243.

[5] T.Y. Kong and A. Rosenfeld, Digital Topology: Introduction and Survey, *Computer Vision, Graphics and Image Processing* 48, 1989, 357-393.

[6] R. Kopperman, P.R. Meyer, R.G. Wilson, A Jordan Surface Theorem for three-dimensional Digital Spaces, *Discrete and Computational Geometry* 6, 1991, 155-161.

[7] E. Kovalevsky, The topology of cellular complexes as applied to image processing, in *Computer Analysis of Images and Patterns*, Proc. II Int. Conference CAIP'87 on Automatic Image Processing, 1987, 162-173.

[8] F.P. Preparata, M.I. Shamos, *Computational Geometry: an introduction*, Texts and Monographs in Computer Science, Springer-Verlag, 1985.

[9] A. Rosenfeld, Connectivity in Digital Pictures, *Journ. Assoc. Compt. Mach.* 17, 1970, 146-160.

[10] A. Rosenfeld, Arcs and Curves in Digital Pictures, *Journ. Assoc. Compt. Mach.* 20, 1973, 81-87.

[11] A. Rosenfeld, Adjacency in Digital Pictures, *Information and Control* 26, 1974, 24-33.

[12] A. Rosenfeld, Digital Topology, *Amer. Math. Monthly* 86, 1979, 621-630.