Topological Operators on the Topological Graph of Frontiers

Ehoud Ahronovitz, Christophe Fiorio, Sylvain Glaize

Topological Operators on the Topological Graph of Frontiers

Ehoud Ahronovitz, Christophe Fiorio, and Sylvain Glaize

LIRMM, 161 rue Ada, F-34392 Montpellier Cedex 5

Abstract. The Topological Graph of Frontiers is in our opinion a good graph structure representing the topology of segmented images. In this paper we deal with topological operators which achieve directly on the graph current operations performed on segmented images.
Well known graph structures such as the Region Adjacency Graph [Pav77] [Ros74] do not (and cannot) keep track of the topology and so cannot maintain it. We claim that the structures and operators described here, on the contrary, allow and do this maintenance. One of the most important informations in such images is the inclusion of nested regions and one of the most important operators is the union of regions. We deal essentially with these in this paper. They are described in detail herein and we show how the topological coherence is maintained. This is why we entitle them topological operators. Other operators that we have already developed are briefly described.
Keywords: topological operator, enclosed region, topological graph of frontiers, topological representation, segmented image manipulation.

1   Introduction

In image analysis we need tools to structure and manipulate images and their regions. In previous works [Fio95,AAF95] we developed a topology (called *star-topology*) aiming at a good representation of images and a graph structure called topological graph of frontiers (*TGF* for short) [Fio96]. They lead to represent regions and their topological relations in 2D images. This was necessary because there were no representations well adapted neither to the very close study of the adjacency relations, nor to the inclusion of regions.

Now we aim at topological operations and manipulations on images through the graph structure of the *TGF*. We entitle them *topological* because obviously such operations should maintain the topological coherence of the resulting image and of course of the associated data structures.

Here we present essential operations on regions: we focus on the topological union, then we show how operations like image or region(s) cutting or subgraph manipulations can be achieved. In order to do such operations, we have to add to the *TGF* some attributes like the depth level of regions. We also have to classify and mark edges according to this depth level. So we have to study thoroughly the inclusion of regions. Indeed, we are led to define the *enclosing* of regions,

which is at a first glance the inclusion of a region not only in another region, but in a set of regions.

This paper is structured as follows: first we make a short review of the *TGF*. Then the depth-weighting of the *TGF* and the darts (sort of half-edges, described in the previous part) marking are detailed in section 3. They lead to the topological union operation described in the next section.

The cutting and the subgraph operations are rather too long to be detailed here. We only give their main features, the complete technical details are available in [Gla98].

All the main topics come with algorithms for each step and we study the complexity when necessary.

2   Outlines of the *TGF*

The Topological Graph of Frontiers is a topological consistent representation of images. It is based on the star-topology [AAF95,KKM90,Kov89] and is derivated from combinatorial maps [Lie89,Edm60]. We give hereafter the outlines of this structure.

Firstly, we have to recall that the RAG (Region Adjacency Graph) does not preserve or maintain all the topological information about the regions in the image [KM95,Kov89,Fio95]. So, if we still want to process an image through a graph representation, we need a more complete structure. This is why we developed the *TGF*.

2.1   The *TGF* structure

The reader not accustomed to graph structures, may follow figures 1, 2 and 3 along with this paragraph.

The *TGF* is a planar multi-graph where:

- each vertex represents one region,
- each edge represents one frontier between two regions and not only the adjacency (e.g. two adjacent regions having n common frontiers will yield n edges between the corresponding vertices.),
- each edge is composed of two "half-edges" called hereafter *darts*, so that each dart represents the frontier as a part of the contour of one of the two adjacent regions,
- the inclusion of regions is represented too.

Note: We can see that each dart represents a part of the boundary as seen by the interior of the region. Thence, two adjacent regions have as many edges as they have different (not connected) boundaries between them, and each edge is split into two darts, each representing the boundary as seen from the corresponding region. This is important if we want to maintain topological coherence [Lie89].
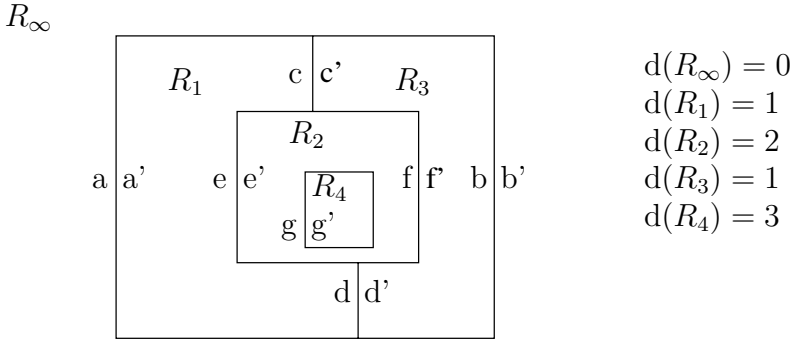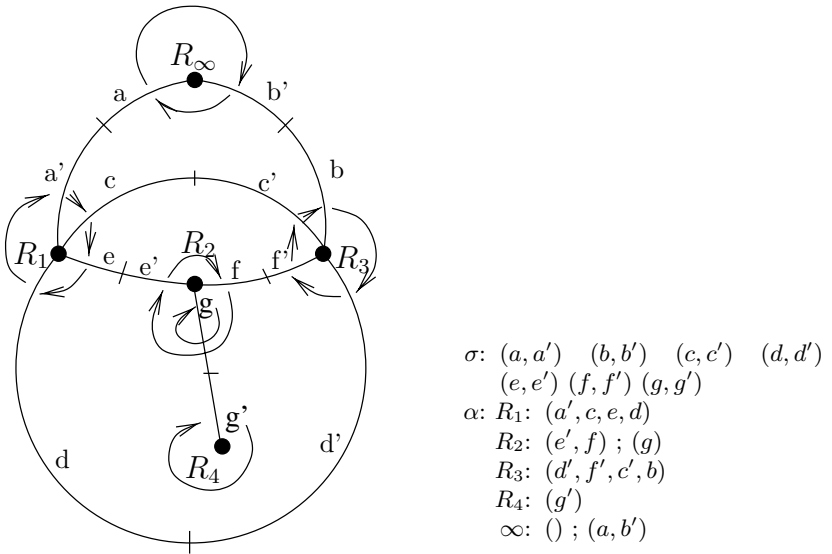
R_∞

$$d(R_\infty) = 0$$
$$d(R_1) = 1$$
$$d(R_2) = 2$$
$$d(R_3) = 1$$
$$d(R_4) = 3$$

Fig. 1. Depth weighted regions

σ: (a, a') (b, b') (c, c') (d, d')
   (e, e') (f, f') (g, g')
α: R_1: (a', c, e, d)
   R_2: (e', f) ; (g)
   R_3: (d', f', c', b)
   R_4: (g')
   ∞: $()$; (a, b')

Fig. 2. *TGF* of segmented image of Figure 1

Fig. 3. *TGF* algebraic representation

We recall that [Fio96] presented a linear algorithm scanning the image with a 2×2 window and constructing the *TGF* along the scan.

Let $G(V, D, \alpha, \sigma)$ be a *TGF*, V is the set of vertices, D the set of darts, α a permutation on D and σ an involution on D. The set of edges will be denoted E (see hereafter). The features of such a graph are:

1. Each region is represented by one and only one vertex.
2. Each dart e is always incident to a vertex R in V. We say that e is incident to region R and denote e^R the vertex incident to e.

3. Each edge is an unordered couple $(e, \sigma(e))$ where $e \in D$. Let E be the set of edges; then $(e_1^R, e_2^{R'}) \in E$ iff there exists a frontier between R and R'.
4. α is a permutation on D whose cycles are circular permutations on a set of darts incident to the same vertex. Each cycle represents one contour of a region and follows exactly the order induced by the set of frontiers composing the contour. Thus following the darts linked in one α cycle, consists also in following one entire closed contour of the region.
5. $R_\infty \in V$ is a particular vertex which represents the exterior of the image. It is somewhat a region containing all the other regions of the image and having no exterior contour.
6. We associate to each vertex the list of α cycles corresponding to the region represented. The outer contour (exterior) will be the list head except for R_∞ (no exterior contour).

Figure 2 and 3 show the graphic and algebraic versions for the TGF associated to the segmented image of Figure 1. Note the list of α cycles and the void cycle in R_∞ which has no exterior contour.

2.2   A first point

Advantages of the TGF lie essentially on the multi-graph modelizing the frontiers and the inclusion of regions. However, it lacks the structure of enclosed regions (deeper than the only one level of inclusion). In the next section we show how we added this feature.

Now, we can define the inclusion of regions by the following:

– A region A is said to be *strictly included* in a region B iff there exists an edge between A and B such that the incident dart (of its edge) to A is in an interior contour of B. We note $A \prec B$.
– A region A is said to be *included* in a region B iff there exist a sequence $(R_1, \ldots, R_n)$ of regions such that $A \prec R_1 \prec \ldots \prec R_n \prec B$. We note $A \subset B$. Note that this last inclusion relation defines a complete order relation.

The TGF structure is quite sufficient to check the inclusion of regions. However, if we want to manipulate set of regions, or if we need to know wether a region is *enclosed* (strictly defined later) in a set of regions, we have to complete the information joined to the TGF.

The next section deals with that problem and labels regions with a depth information and darts with an inclusion mark.

3   Depth-weighting in a TGF

In order to distinguish exterior and interior regions in a TGF, we associate to each vertex a depth parameter relative to region R_∞. We denote $d(R)$ the depth of region R, and $N(R)$ the set of the neighbouring regions of R. A neighbouring region is a region having a common frontier with R.

The depth parameter can be defined by $d(R) = min_{R_i \in N(R)}(d(R_i)) + 1$

A *TGF* is said to be *depth weighted* if every region owes a depth parameter. In order to depth weight the *TGF*, we apply simple breadth search algorithm to the *TGF* starting with R_∞. This is rather easier than trying to weight during the construction of the graph. Figure 1 gives an example of regions depth weighed.

3.1   Dart marking

The dart marking operation will allow us to make a classification of the darts relatively to a set of regions. Then the classification will lead to determine regions enclosed in a set of regions. As this notion is important for operations like the topological union or subgraph extraction, let us first give a definition for the union of regions.

Simple union definition: The first type of union is called *simple union*. It is valid on any set of linked regions, that is to say that for any two regions A and B of the set, there exists a path from a dart incident to A to a dart incident to B. The path may be any combination of the α and σ involutions (*orbite* in combinatorial maps).

Let V' denote the set of vertices of the linked regions $R_1, \ldots, R_n$. Then the simple union $\bigcup R$ of $R_1, \ldots, R_n$ is $G(V', D', \alpha', \sigma')$ where D' is the union of all incident darts to the vertices in V' and α' and σ' the restrictions to D' of α and σ.

Note that this is only a mean to denote a set of regions in a *TGF*. It may not denote a region, as it may contain a dart d such that $\sigma(d) \in D'$, or even have two darts towards a same region for only one frontier.

The classification of darts studied in the next section will lead us from the above simple union to the topological union of regions which in turn results in a valid region and maintains the coherence of the graph representation.

Classification of darts The classification that we make separates the darts into three classes: exiting, enclosed or entering[1]:

- *exiting darts* are defined and marked by Algorithm 1.   Intuitively such darts lead to regions of lower depth than the depth of the regions simply united, that is to the outside.
  The principle of the algorithm is to find a dart of depth $n - 1$, n being the minimal depth of the simply united regions, then to examine the darts without going into the set of simply united regions.
  Darts are examined in the order induced by α as long as no enclosed dart is found (see Algorithm 1). In this case we proceed to the next region according to σ. Exiting darts always exist except if R_∞ is in the simple union. In this case, exiting darts make no sense.

---

[1] **Important remark:** Note that this classification is made relatively to a given set of simply united regions. If this set is changed, then the classification changes too.

Algorithm 1: Marking of exiting darts

Data: a weighted TGF $G(V, D, \alpha, \sigma)$, the set U of vertices of the simply united regions,

Pre-condition: $R_\infty \notin U$

Result: a weighted and marked TGF $G(V, D', \alpha, \sigma)$ where the elements of D' are those of D with a mark on the exiting darts relatively to U.

$p_{union} \leftarrow \min_U p(R)$;
/* find the first exiting dart */
$b \leftarrow$ a dart such that $p(\sigma(b)^R) \geq p_{union}$ and $\sigma(b)^R \notin S$;
$b_{init} \leftarrow b$;
$marque(b) \leftarrow$ "exiting";
repeat
  $\quad b \leftarrow \alpha(b)$;
  $\quad$ **if** $\sigma(b) \in S$ **then**
  $\quad\quad$ /* move to next region */
  $\quad\quad b \leftarrow \sigma(b)$;
  $\quad$ **else**
  $\quad\quad marque(b) \leftarrow$ "exiting";
until $b == b_{init}$;

- *enclosed darts*: a dart b is enclosed if $\sigma(b) \in \bigcup R$. Such darts exist if the set $\bigcup R$ of simply united regions contains at leat two regions. Intuitively, they belong to edges joining two vertices of the simple union.
  Such darts should not exist in a TGF. So an algorithm making a *topological union* (see Section 4 for the definition) should discard them.
  A simple algorithm can look for non yet marked darts and mark as enclosed those that verify the condition. It is also possible to include this marking in the exiting darts algorithm (see Algorithm 1): when moving to the next region ($b \leftarrow \sigma(b)$), mark b and $\sigma(b)$ as enclosed.
- *entering darts* can be defined as the incident darts to $\bigcup R$ (incident to any of the simply united regions) that are neither exiting nor enclosed. Intuitively, they lead to a deeper region (depth parameter higher) than the set of simply united regions.
  If such darts exist, it means that there are regions "included" in the set $\bigcup R$. This will be a good way to define and recognize "enclosed" regions (see below).

Marking algorithm for a TGF: As seen before, all the incident darts to a set of simply united regions can be marked according to one of the three classes. What about the unmarked darts ? They are not adjacent to any region concerned by the union, so that we mark them "unconcerned".

Note that the algorithm does not modify neither te vertices nor the involutions. It merely adds a marking information to all the darts.

Marking example: Figure 2 represents the *TGF* for the segmented image of Figure 1. Suppose that R_1 and R_3 are simply united. The depth of the union $\bigcup\{R_1, R_3\}$ is 1 as $d(R_1) = d(R_3) = 1$. The initial dart is then a' or b' as $\sigma(a')^R = R_\infty$ and $d(R_\infty) = 0$ (lower depth level) and likewise for $\sigma(b')^R$.

Let a' be the initial dart. Then a' is marked exiting and we proceed to $\alpha(a') = c$. As $\sigma(c) = c'$ and $c'^R \in \bigcup\{R_1, R_3\}$, c and c' are enclosed. Then as $\alpha(c') = b'$, b' is exiting; $\alpha(b') = d'$ and $\sigma(d')^R \in \bigcup\{R_1, R_3\}$. Hence d and d' are enclosed. Finally, $\alpha(d) = d'$; the exiting darts are all marked; there are no more enclosed darts, therefore e and f' are marked entering and all the other darts are unconcerned.

We are very close now to be able to define the *topological union* and the *enclosure* of regions. Thus, in the previous example, it will lead us to claim that R_2 is enclosed in the topological union of R_1 and R_3.

4   Topological union of regions

The *topological union* of regions consists in replacing a set of linked regions by one region, while keeping coherent the *TGF* structure.

When the classification (marking) process of the darts relative to a given simple union ends, we can easily deduce the region achieving the topological union. It is sufficient to:

- delete all the enclosed darts;
- keep only one edge per frontier so that:
    - for a frontier with an exterior region the edge is relative to an exiting dart,
    - for a frontier with an interior region the edge is relative to an entering dart;
- change all remaining entering darts to interior contours.

Note that during all this process, we have to maintain the α and σ coherence.

Figure (4) shows the result of the topological union of regions R_1 and R_3 in figure 2, where:

- enclosed darts c, c', d and d' are discarded,
- exiting darts a and a' (or b and b') are deleted in order to keep one edge to R_∞,
- entering darts e and e' (or f and f') are deleted in order to keep one edge to R_2,
- f and f' (or e and e') are changed to an interior contour.

Before the review of the topological union algorithm (Algorithm 2), we can define an enclosed region.

A region A is said to be *strictly enclosed* in a set S of regions iff there exists an edge between A and the topological union of S such that the corresponding incident dart to A is in an interior contour of the topological union. We denote it by $A \blacktriangleleft S$.
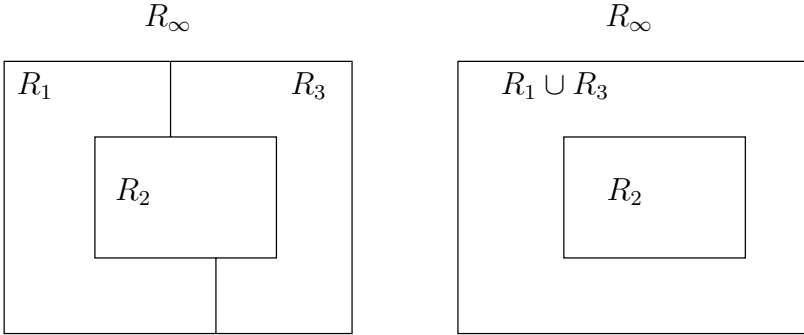
$$R_\infty \qquad\qquad\qquad\qquad R_\infty$$

| R_1 | | R_3 |
| | R_2 | |

| $R_1 \cup R_3$ |
| R_2 |

Fig. 4. left: before topological union; right: after

A region A is said to be *enclosed* in a set S of regions iff there exists a sequence $(S_1, S_2, ..., S_n)$ of sets of regions such that $A \blacktriangleleft S_1$ and $\forall S_{1_i} \in S_1, S_{1_i} \blacktriangleleft S_2$ ... $\forall S_{n-1_i} \in S_{n-1}, S_{n-1_i} \blacktriangleleft S_n$ ... $\forall S_{n_i} \in S_n, S_{n_i} \blacktriangleleft S$. We denote it by $A \text{ C } S$.

The algorithm making the topological union lies heavily on the darts marking one. It does not modify directly the darts of the TGF as it needs the TGF structure during the whole marking process. Hence, for example, we do not discard enclosed darts as soon as they are found.

Algorithm 2 summarizes the topological union process. A complete description can be found in [Gla98]. Note that at the end of the process, a new depth-weighting of the TGF is necessary. Indeed, after a topological union, the depth level is no longer true:

Let R_1, R_2, R_3 be three regions such that $d(R_1) = d(R_2) - 1 = n$, R_1 and R_2 being neighbours, but R_3 being a neighbour of R_2 only. Thus, $n \leq d(R_3) \leq n+2$. The depth of R_U, the topological union is n. Then R_2 and R_U will have a discrepency of 2 in their depths, which is wrong.

In order to avoid a complete computation of the depth, we have to check the depth level coherence between the exterior and all interior contours of the topological union R_U. If a region R_f with a common frontier with R_U has a wrong depth level, then the depth of all the linked regions to R_f should be recomputed (R_U being excluded from this action).

4.1   Complexity

The algorithm scans all the darts of the simple union. Each dart is checked once, except the first one for each processing of algorithm 1. Hence, the complexity is $O(n.\phi)$, n being the number of darts in the simple union and ϕ the complexity to access one element in the dart list.

Algorithm 2: Topological union of regions

Data: a weighted *TGF* $G(V_1, D_1, \alpha_1, \sigma_1)$, a set of vertices U ($R_\infty \notin U$).
Result: a weighted *TGF* $G(V_2, D_2, \alpha_2, \sigma_2)$ in which the regions in U are topo-
        logically united into one region.
Initialization (building R_U);
Select the first (exiting) dart;
repeat
    check next exiting dart;
    if *dart is redundant* **then**
        mark redundant
    else
        if *dart is enclosed* **then**
            mark enclosed
        else
            mark exiting;
            add to R_U;

until *current dart == starting dart*;
while *there exists a non marked dart* **do**
    select this dart;
    if *dart is enclosed* **then**
        mark enclosed
    else
        /* entering */
        check all associated entering darts;
        add them to R_U;
/* All concerned darts are checked */
discard the simply united regions;
update α and σ;
recheck depth level;

---

5   Other operators

We defined also a cutting operator as well as extracting parts of images along with their subgraph structure.

For example the cutting operator allows to cut a region into two different disctinct regions (see Figures 5 and 6). The problem is that we cannot cut a region according to any given path (see Figure 7).
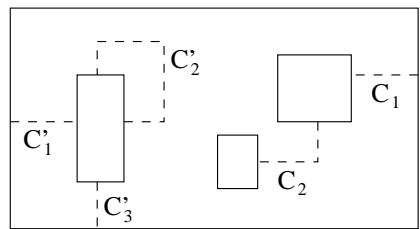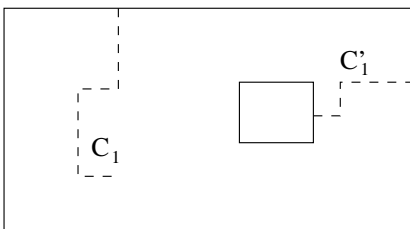
[Gla98] contains a detailed study of valid interpixel paths (see Figure 8) for the cutting operator. The algorithm principles are rather long. Figures 5 to 8 give a slight idea of how the algorithm works and how it checks the validity of a cutting path.
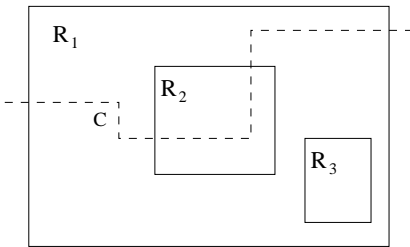
Fig. 5. Cutting path
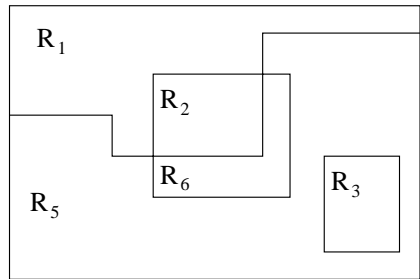

Fig. 6. Regions after cutting

Fig. 7. Non valid cutting paths

6   Conclusion

We have defined new topological operators on the TGF. They enlarges the use of such a graph structure in image analysis. We can now merge regions or cut them, extract parts of images or include such elements, while the topological coherence is maintained. This last fact is very important and makes the difference with many classical operators that do not take it into account.

Essentially, we showed in this paper that the TGF is a well-suited structure for topological operations. The depth-weighting and the dart marking are basic processes that lead to a deep view of the inclusion of regions and allow sophisticated search paths in an image.
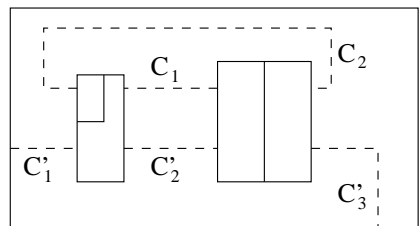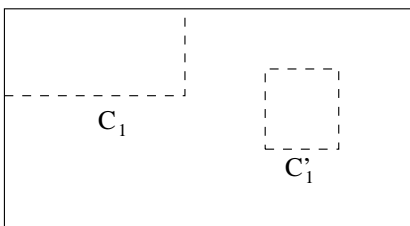
Fig. 8. Valid cutting paths

We can decide to go into a set of regions by an entering dart, scan it completely or partially by α or σ, then skip to another region or set of regions. In fact, we can repeat such operations as needed, due to the topological coherence.

We have to acknowledge that we have not yet thought about similar 3D operations since defining such a structure for 3D image is still under research. We focus currently on new operators, for example the minimal set of regions containing a region or linking non adjacent regions.

References

AAF95.   E. Ahronovitz, J.-P. Aubert, and C. Fiorio. The star-topology: a topology for image analysis. In 5th *Discrete Geometry for Computer Imagery, Proceedings*, pages 107–116. Groupe GDR PRC/AMI du CNRS, september 1995.

Edm60.   J. Edmonds. A combinatorial representation for polyhedral surfaces. *Notices of the American Mathematical Society*, 7, 1960.

Fio95.   C. Fiorio. *Approche interpixel en analyse d'images : une topologie et des algorithmes de segmentation.* Thése de doctorat, Université Montpellier II, 24 novembre 1995.

Fio96.   C. Fiorio. A topologically consistent representation for image analysis: the frontiers topological graph. In S. Ubeda S. Miguet, A. Montanvert, editor, *6th International Workshop, DGCI'96*, number 1176 in Lecture Notes in Computer Sciences, pages 151–162, Lyon, France, November 1996.

Gla98.   S. Glaize. Manipulations topologiques et opérations sur les graphes topologiques des frontières. Technical Report 98122, Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier, 161, rue Ada - F-34392 Montpellier Cedex 5, november 1998.

KKM90.   E. Khalimsky, R. Kopperman, and P.R. Meyer. Computer graphics and connected topologies on finite ordered sets. *Topology and its Applications*, 36:1–17, 1990.

KM95.   W. Kropatsch and H. Macho. Finding the structure of connected components using dual irregular pyramids. In 5th *Discrete Geometry for Computer Imagery, Proceedings*, pages 147–158. Groupe GDR PRC/AMI du CNRS, september 1995.

Kov89.   V.A. Kovalevsky. Finite topology as applied to image analysis. 46:141–161, 1989.

Lie89.   P. Lienhardt. Subdivision of n-dimensional spaces and n-dimensional generalized maps. In 5th *Annual ACM Symposium on Computational Geometry*, pages 228–236, Saarbrücken, Germany, 1989.

Pav77.   T. Pavlidis. *Structural pattern recognition.* Springer-Verlag, New-York, 1977.

Ros74.   A. Rosenfeld. Adjacency in digital pictures. *Inform. and control*, 26:24–33, 1974.