HAL
open science

Neighborhood graphs and image processing

François Angot, Régis Clouard, Abderrahim Elmoataz, Marinette Revenu

To cite this version:

François Angot, Régis Clouard, Abderrahim Elmoataz, Marinette Revenu. Neighborhood graphs and image processing. SPIE European Symposium on Lasers, Optics, and Vision for Productivity in Manufacturing, 1996, Besançon, France. pp.12-23. hal-00980226

HAL Id: hal-00980226

https://hal.science/hal-00980226

Submitted on 17 Apr 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Neighborhood graphs and image processing

François Angot, Régis Clouard, Abderrahim Elmoataz, Marinette Revenu

GREYC - ISMRA

6 Boulevard Maréchal Juin - F 14050 CAEN cedex - FRANCE

tél: 31-45-27-21 - fax: 31-45-26-98 - e-mail: Francois.Angot@greyc.ismra.fr

ABSTRACT

Many image processing and image segmentation problems, in two or three dimensions, can be addressed and solved by methods and tools developed within the graph theory. Two types of graphs are studied: neighborhood graphs (with the duals Voronoï diagram and Delaunay graph) and adjacency graphs. In this paper, we propose an image representation based on graphs: the graph object, together with methods for attributing and weighting the graph, and methods to merge nodes, is defined within an object-oriented library of image processing operators. In order to demonstrate the interest of the approach, several applications dealing with 2D images are briefly described and discussed: we show that this change of representation can greatly simplify the tuning of image processing plans and how to replace complex sequences of image operators by one single basic operation on graphs. As results are promising, our library of graph operators is being extended to 3D images.

KEYWORDS

Image Processing, Neighborhood graphs, Adjacency graphs, Voronoï graphs

1. INTRODUCTION

Many image processing and image analysis problems, in two or three dimensions, can be addressed and solved by methods and tools developed within the neighborhood graph theory.

This is particularly the case when dealing with images showing populations of objects, or over-segmented images. In several situations, it is not enough to work with points (pixels or voxels); more global objects must be considered: visual primitives such as regions or boundary elements. Also when objects of the scene are not connected to each other, more general neighborhood relations than the digitalization grid of the image must be considered.

In some cases, rather than the neighborhood structure, it is the information about objects and neighborhood relations that must be manipulated:

- When the image contains several categories of objects, different characteristics have to be associated to the objects to distinguish them. Each object can be represented as a node of a graph, which stores these characteristics. This is the case when trying to quantify the distribution of an object population within another larger population.

- Bottom-up segmentation often results in over-segmented images. In order to improve the segmentation, neighboring regions can be merged according to some criteria and information must be associated to pairs of objects, i.e. to the neighborhood relations between objects.

Our objective is to design an image representation based on graphs. For that purpose, we have to define a data structure which allows to manipulate relationships between image primitives (i.e. proximity, connexity, similarity, etc.). The image concept can then be generalized in order to define specific methods and operators on graphs.

This approach has mainly been validated on 2D histological section images. This biological material shows cellular nuclei, organized in various architectures. Thanks to neighborhood graphs we could easily delimit clusters of nuclei, which classical methods have difficulties to deal with.

2. GRAPH THEORY

2.1. Definitions

A graph, G is a mathematical object defined by two inter-connected sets: a set of points, V, and a subset, E, of $V \times V$[1]. The elements of V are called the vertices or the nodes of the graph, the elements of E the edges:

$$G=(V,E) \text{ with } E \subset V^2, \tag{1}$$

$$\text{and } k \in E \Leftrightarrow \exists p,q \in V \,/\, k=(p,q). \tag{2}$$

This can also be written as:

$$V=\{p_i\} \text{ and } E=\{k_{i,j}\} \text{ with } k_{i,j}=(p_i,p_j). \tag{3}$$

Generally, edges are oriented. In the previous example, p is the origin of the arc k, q is its end. A graph can have loops if the origin and the end of an edge are the same point:

$$k \text{ is a loop} \Leftrightarrow \exists p \in V \,/\, k=(p,p). \tag{4}$$

In our case, an edge represents a neighborhood relation. Because this relation is symmetrical and anti-reflexive, only non-oriented graphs, without loop will be used. In that case, edges verify:

$$\forall k \in E, \exists p,q \in V \,/\, p \neq q \text{ and } k=(p,q), \tag{5}$$

$$\forall p,q \in V, (p,q) \in E \Leftrightarrow (q,p) \in E. \tag{6}$$

In a graph $G=(V,E)$, the neighborhood relation N is defined by:

$$\forall p,q \in V, q \in N(p) \Leftrightarrow (p,q) \in E. \tag{7}$$

For non-oriented graphs, this neighborhood relation is an equivalence relation, on which all graph processing related to connexity is based. It is also the source of the definition of trees, which are connected graphs, without cycle.

According to these definitions, a graph can be considered as an image with an irregular grid. An image is a set of points associated to a digitalization grid. The nodes of a graph are then analogous to the points and the edges take place of the grid. Following the same analogy, connected components of a graph can be associated to regions in an image.

It must be noticed that the dimension of an image is given by the digitalization grid: 2D, 3D. In a graph however, the nodes can be member of a space of any dimension. This dimension has no influence on the neighborhood structure of the graph. Thus, a graph considered as an image simplifies the generalization from two to three dimensions, which often causes problems[2].

2.2. Attributed and weighted graphs

Until now, the analogy between graphs and images only concerns the neighborhood notion. But it can also be extended to the information associated to points (gray level, labels, etc.). A numerical attribute can thus be associated to each node of V. The graph and vertices are then attributed by A:

$$A: \quad V \rightarrow \mathrm{R}$$

$$p \rightarrow a(p)$$

This function is a generalization of the function used by L. Vincent[3] in his study about mathematical morphology operators. We will see that most image processing operators can thus be extended to graphs.

In the case of 2D images, the grid introduces the 4-neighborhood, the hexagonal neighborhood, the 8-neighborhood, or even larger ones. In the case of graphs however, the neighborhood relations are not fixed in the 2D space. In addition to the attributing function, extra information about the neighborhood relation can be given by the means of a weighting function defined on the edges of E as:

$$W: \quad E \rightarrow \mathrm{R}$$

$$k \rightarrow w(k)$$

The weight of an edge is often defined as the difference between the attributes of its extremities. It is called canonical weighting:

$$\forall k_{i,j}=(p_i,p_j),\ w(k_{i,j})=|\ a(p_i)-a(p_j)\ |. \tag{8}$$

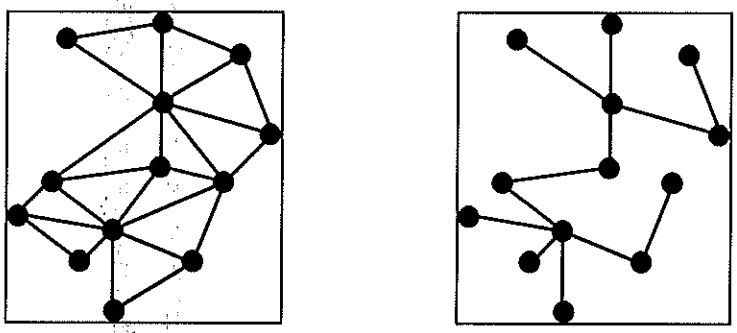
In some cases, the set V is included into a metric space. The distance defined on vertices then corresponds to the weight of edges:

$$\forall p,q,r \in V,\ d(p,p)=0,\ d(p,q)\geq 0,\ d(p,q)+d(q,r)\geq d(p,r), \tag{9}$$

$$\forall k=(p,q),\ \mathrm{w}(k)=\mathrm{d}(p,q). \tag{10}$$

So, attributes of a graph are a means to quantify nodes, whereas weights can correspond to a distance between the ends of edges. This distance can be a geometric distance defined in the space of nodes. But it can also correspond to a difference between objects (i.e. edge length).

Thanks to that weighting function, an important structure can be deduced from a graph: the minimum spanning tree (MST). The MST of a graph is the spanning tree where the sum of the edge weights is minimum. Two efficient algorithms build the MST: the Prim algorithm proceeds by making the node set grow, whereas the Kruskal algorithm proceeds by making the edge set grow.



1.a 1.b

Figure 1: example of a graph (a) and one of its spanning trees (b).

The attributing and weighting functions give the possibility to manipulate a graph as a numerical image with an irregular grid. We are now going to describe some image processing operators based on the graph theory.

3. IMPLEMENTATION

Our work on neighborhood graphs has been developed within the environment of the HORUS library. HORUS[4] is a library of image processing operators. Operators can manipulate different kinds of 2D objects: images of points (gray levels), images of labels, region maps, etc. In fact , any authorized data type can be considered as an image. It is particularly true for graphs, and we accordingly enriched the library with graph processing operators[5]. HORUS operators exhibit the following special characteristics:

- An operator can work on various kinds of images. As the library development has been made with an object-oriented language (C++), the operator chooses the function to apply, according to its entries. For instance, a thresholding operator can be applied either to a graph or to an image. This is convenient for users who do not need to have a complete knowledge of the structure, but only have to link simple operators to perform complex tasks.

- Each operator corresponds to an "atomic" operation and a complex "molecular" operation is made by the linking of several simple operators. It is thus rather easy to extend simple operators to graphs, whereas complex image processing programs are almost impossible to transform and extend them to graph structures. An operator takes the form of a run-time program with input and output parameters. These parameters can be

image files or numerical values so as to adjust the operator's behavior to the context. The UNIX operating system offers a programming language (SHELL) which allows to link programs, and redirect parameters thanks to pipes.

3.1. The graph structure

There are two classical ways to represent graphs, adjacency matrices and adjacency lists. In both cases, nodes must be numbered.

- Adjacency matrices offer direct access to an edge, given the numbers of two vertices. However, even if the number of arcs is very low, the matrix size is still v^2, where v is the number of vertices: v=Card(V). In the case of non-oriented graphs, the number of values can be reduced to $(v-1).v/2$.

- Adjacency lists are linked lists, associated to each vertex. They only represent existing edges, but imply a list search to access a particular edge. For non-oriented graphs, each edge is stored twice.

We have chosen the latter representation in order to minimize the size of data.

Figure 2 gives an idea of the data structure used to represent graphs. This data structure contains information for graph visualization: the position of each vertex, and the size of the corresponding image to localize nodes in the 2D space.

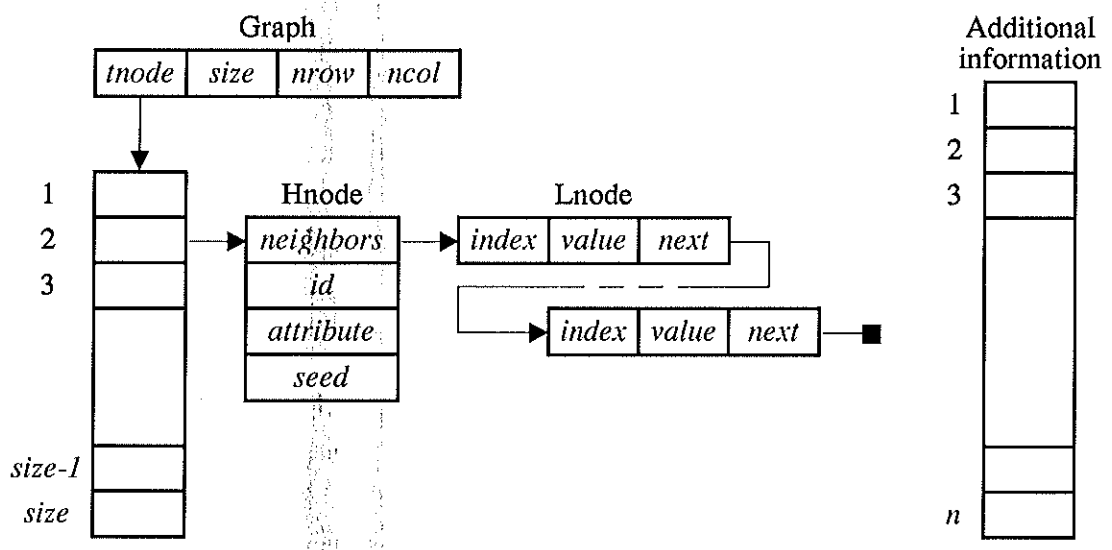


Figure 2: graph structure.

The fields of the structure have the following meaning:

- For the graph, *nrow* and *ncol* give the dimensions of the corresponding 2D image, *size* is the number of nodes.

- Each element of the array *tnode* is a pointer on a structure containing the list of neighbors (*neighbors*), the vertex identifier (*id*), the attribute of the vertex (*attribute*), and the position of the vertex in the image (*seed*), stored as a point. The only fields to change for manipulating a 3D graph are *nrow*, *ncol* and *seed*.

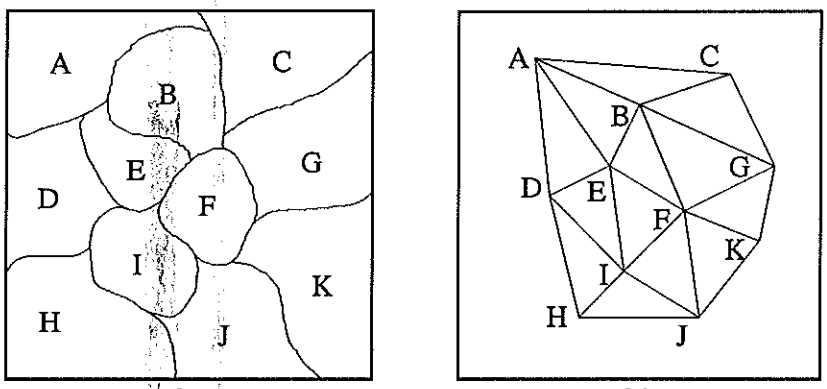
- In the adjacency list of a node, each cell contains the node number in the array *tnode* (*index*), the weight of the edge (*value*) and a pointer to the rest of the list (*next*).

The identifier of the nodes (*id*) is the only useful field to access additional information.

3.2. Constructing the graph

Depending on the application (hierarchical segmentation, characterization of an object population, etc.), and the related classes of images, either adjacency graphs or neighborhood graphs can be considered:

➢ In an image segmented into regions, boundaries correspond to the neighborhood relations between regions and the adjacency graph corresponds to the connexity between regions (figure 3). The values of nodes can be used to store a great range of characteristics of the regions such as the mean gray level or shape parameters (direction, elongation, area, etc.). The weights of edges represent properties of the boundaries such as the difference between attributes of the regions or the contrast on the boundary. Segmentation methods based on region merging according to various criteria, are greatly simplified when manipulating the image through its associated adjacency graph.



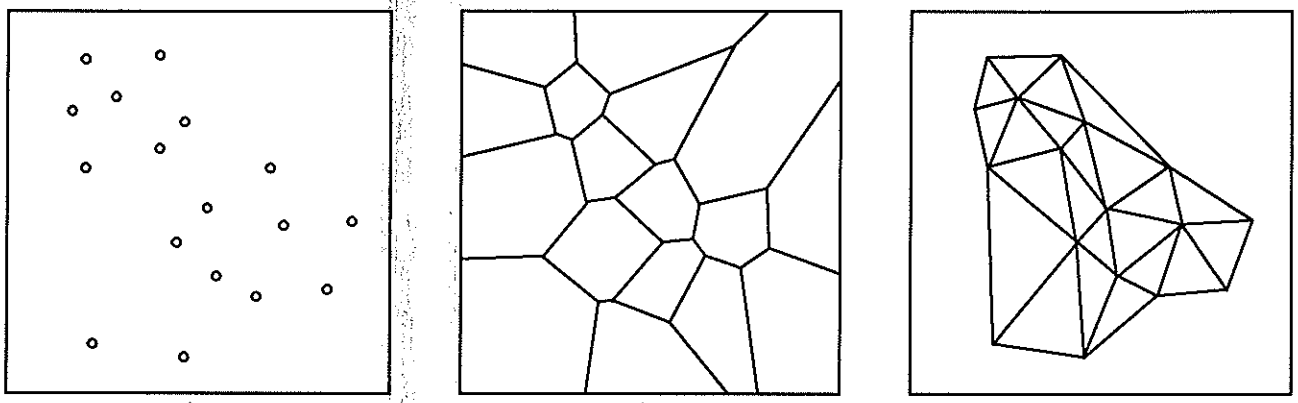
3.a 3.b

Figure 3: segmented image (a) and its corresponding adjacency graph (b).

➢ In the case of an object population, neighborhood relations come from the Voronoï diagram (figures 4a and 4b). This diagram is a tessellation of the space according to an object collection $\{o_i\}$, possibly reduced to points[6]. Each object o_i is associated to a region $vor(i)$ which contains the points closer to object o_i than to any other object:

$$p \in vor(i) \Leftrightarrow \forall j \neq i,\ d(p,o_i) \leq d(p,o_j). \tag{11}$$

The dual graph of the Voronoï diagram is called the Delaunay graph (DG). In this graph, each edge corresponds to a boundary of the Voronoï tessellation and represents a neighborhood relation between two objects; this is a neighborhood graph. It is also called the Delaunay triangulation, as edges only draw triangles in a 2D space (figure 4c). The attributes of the graph store information about the objects of the population, and weights characterize the neighborhood relations.



4.a 4.b 4.c

Figure 4: object population (a), Voronoï diagram (b) and Delaunay graph (c).

The construction of the adjacency graph or the DG from the Voronoï diagram can be done in one video scan of the image only. Each boundary between regions gives rise to an edge. For the construction of the Voronoï diagram in a discrete space, several methods exist and some are of great interest for generalization to higher dimensional spaces. With 2D masks,

two scans of the image will make the regions grow in the two opposite directions. However, the use of masks becomes heavy for higher dimensional spaces (3D).

A priority-based growing around the objects is then used to simultaneously build all Voronoï regions. This can be done using Euclidean distance or other metrics according to applications. This growing method can be used for other purposes: when encountering the first contact between two growing regions, the corresponding edge of the DG can be created. In addition, the edge can be weighted with the minimum distance between the two corresponding objects.

From the DG, several other interesting graphs can be deduced. When nodes are attributed by the nearest-neighbor distance, the influence graph can be considered. In the case of weighted graphs, families of more or less dense graphs can be constructed: α-graphs[7] which give information on the external distribution of edges, and β-graphs, describing the internal organization of nodes.

4. IMAGE PROCESSING BASED ON GRAPHS

Relying on the previously introduced analogy between images and graphs, we will show that most image processing operators can be extended to graphs. We will also see that some special additional operators must be developed.

In the case of an image, as a value is already associated to each point, image processing operators can be applied directly. In the case of a graph however, the only information associated to a point is the neighborhood structure. So, before applying image processing operators, a graph must be attributed and weighted by special operators.

The use of graphs implies two kinds of data examinations:

- A sequential examination considers each point in a pre-defined order. This can be the order of the array *tnode* of the structure. It is the case with operators such as thresholding, and it can be likened to applying masks to images. A point can be modified according to its value or according to its neighbors (in the mathematical morphology operators for instance).

- The examination of the neighbors of a point occurs in growing-based operators. It implies the use of a queue structure. The principle is the same as with an image: nodes are examined in an order that depends on their neighborhood and their values. This examination can be used to label groups of nodes, to calculate distance graphs, etc.

For graph visualization, several operators have been developed that must take into account both the neighborhood structure, the values of nodes, as well as the values of edges.

4.1. Attributing and weighting graphs

In order to put values on nodes and edges of a graph, a series of operators have been developed. One of their input parameters in a graph giving the neighborhood or adjacency structure. The output parameter is a graph with the same structure, where the elements (nodes and edges) are now attributed or weighted. The other input parameters are gray level images or region maps. Here are some examples of such operators:

- To compute the area of objects, the operator needs a region map where the numbers refer to the identifiers of the nodes (*id*). The number of points of each identifier is measured and stored into the corresponding node.

- To get the mean gray level of objects, the operator needs a region map (which is used as a mask and to measure the area) and a gray level image. The sum of the gray levels, divided by the area is stored into the node.

- To calculate the length of boundaries, a region map is required. The number of points of label u and neighbor to a point of label v is measured and stored into edge (u,v).

4.2. Mathematical morphology

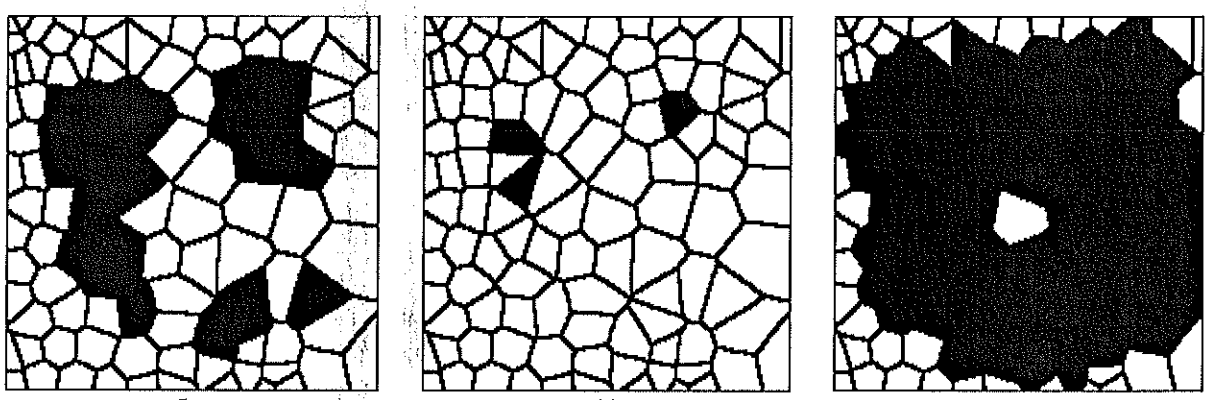
All classical image transforms by mathematical morphology (MM) can be extended to graphs, thanks to two basic operators: morphological erosion ε_n and dilation δ_n at the order n. The order correspond to the size of the neighborhood, i.e. the length of the path between vertices.

$$\varepsilon_n(G)(p)=\min\{\ v(q) \text{ for } q\in N_n(p)\ \}. \tag{12}$$

$$\delta_n(G)(p)=\max\{\ v(q) \text{ for } q\in N_n(p)\ \}. \tag{13}$$

These two operators can manipulate graphs where $v(p)$ can represent area, orientation of objects, mean distance to the neighbors of a node, etc.

Erosion and dilation, like image filtering, make local transformations on graphs. They can be applied either to numerical or binary graphs.



5.a 5.b 5.c

Figure 5: erosion (b) and dilation (c) of a binary graph (a) shown on a Voronoï diagram.

From these two basic operators, many other morphological operators can be deduced:

- morphological opening γ_n and closing ϕ_n

$$\gamma_n=\delta_n \circ \varepsilon_n \text{ and } \phi_n=\varepsilon_n \circ \delta_n. \tag{14}$$

- morphological gradient

$$g(G)=\delta_1(G)-\varepsilon_1(G) \tag{15}$$

- geodesic erosion and dilation with respect to another function G_2

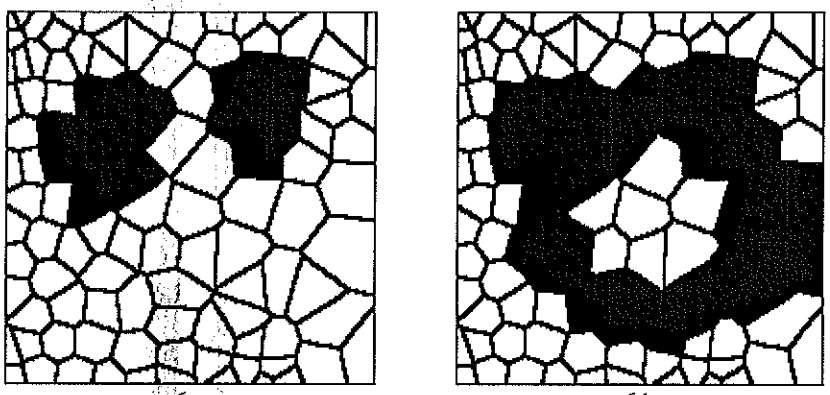
$$\varepsilon_n(G_1,G_2)=\max\ \{\ \varepsilon_n(G_1),G_2\ \}, \tag{16}$$

$$\delta_n(G_1,G_2)=\min\ \{\ \delta_n(G_1),G_2\ \}. \tag{17}$$

- geodesic reconstruction by erosion or dilation

$$\varepsilon_n \circ \ldots \circ \varepsilon_n \circ \varepsilon_n(G_1,G_2), \tag{18}$$

$$\delta_n \circ \ldots \circ \delta_n \circ \delta_n(G_1,G_2). \tag{19}$$



6.a 6.b

Figure 6: opening (a) and closing (b) applied to the graph of figure 5.

Thanks to the MM theory, a whole range of morphological operators can thus be used to work on graph attributes.

4.3. Characterizing a population of values

In certain situations, an image can be considered as a set of values. So, we sometimes have to characterize this population of values. It is the same with graphs: it is possible to describe the distribution of the attributes and weights of a graph.

To that purpose, one can consider the first-order histogram of the values of nodes or the weights of edges. Mean value, variance, cumulated histogram, can be computed.

However, in some cases, this first-order distribution is not sufficient to distinguish between different populations and a second-order histogram must then be considered. It is defined by the following matrix:

$$m_{i,j}=\text{Card}\{\ k=(p,q)\in E\ /\ a(p)=i \text{ and } a(q)=j\ \}. \tag{20}$$

The normalized matrices F or P, dividing $m_{i,j}$ by Card(E) or max$\{m_{i,j}\}$ are often used. Some classical features[8] can then be computed, to describe the second-order distribution:

- second-order moment $\Sigma\Sigma\, p_{i,j}^{\ 2}$
- contrast $\Sigma\Sigma\, (i-j)^2.p_{i,j}$
- entropy $-\Sigma\Sigma\, p_{i,j}.\log(p_{i,j})$

It is important to mention that these values, or other more complex features - providing that they are independent of the size of the population[9] - can be measured from the graph.

4.4. Merging vertices

In some image segmentation problems, using a bottom-up analysis strategy often results in an over-segmented image. On this kind of images, some primitives have then to be merged to improve the segmentation. With the use of a graph, this processing is made by merging vertices.

One first has to choose a convention when merging vertices u and v of a graph G. After the fusion, we can consider that the two nodes belong to an equivalence class. We then choose to keep the node of lower identifier to represent the class. For example, if $u<v$, the identifier of node number v in the array *tnode* will be set to u.

Then the neighborhood relations of the two nodes that have been merged must be modified. When merging u and v, edge (u,v) disappears and all the neighbors of v must become neighbors of u. This is done by the two methods *Link* and *Unlink* of the class *Graph* in the HORUS library.

A general function has been written to perform the merging of two vertices of a graph. It must be mentioned that this

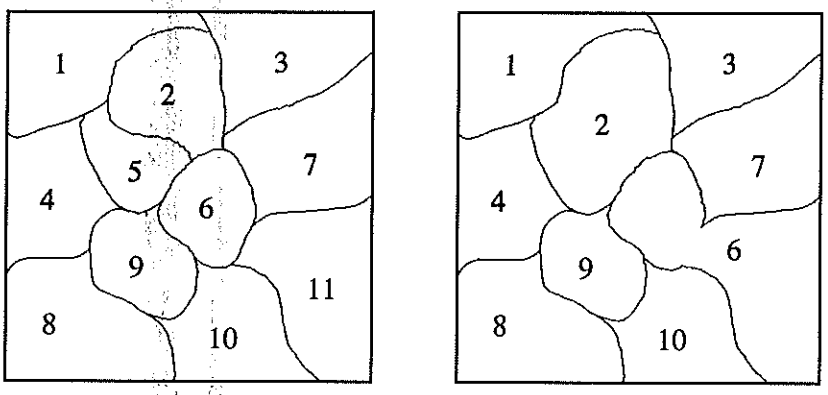
function can be called in various contexts, according to the application. In general, two vertices are merged if the weight of the edge is between two selected values. The fusion of the nodes of a graph can be done in different ways:

- During the examination of each node of the graph: every possible merging is stored. Then all possible fusions are done.

- Another strategy consists in merging the best nodes at some particular step of the processing. Then, one can compute new attributes and weights for the graph and start a new merging phase.

In both cases, the attributes and weights need to be changed. Various methods exist and the choice must be made according to each specific application.

An example of merging is given figure 7.



	id	*neighbors*
1	1	2, 3, 4, 5
2	2	1, 3, 5, 6, 7
3	3	1, 2, 7
4	4	1, 5, 8, 9
5	5	1, 2, 4, 6, 9
6	6	2, 5, 7, 9, 10, 11
7	7	2, 3, 6, 11
8	8	4, 9, 10
9	9	4, 5, 6, 8, 10
10	10	6, 8, 9, 11
11	11	6, 7, 10

	id	*neighbors*
1	1	2, 3, 4
2	2	1, 3, 4, 6, 7, 9
3	3	1, 2, 7
4	4	1, 2, 8, 9
5	2	
6	6	2, 7, 9, 10
7	7	2, 3, 6
8	8	4, 9, 10
9	9	2, 4, 6, 8, 10
10	10	6, 8, 9
11	6	

Figure 7: merging nodes 2-5 and 6-11.

5. APPLICATIONS

In this paragraph, we are going to give an idea of some applications in which operations on graphs can bring significant improvements. These applications are dealing with biomedical images (histology and cytology) and aerial images.

5.1. Cluster detection

In a study on immunohistochemical images (figure 8a), pathologists try to quantify the presence of an immunostaining on microscopic sections, in order to measure the proportion of labelled tumoral nuclei. On the corresponding color images, the labelled nuclei (tumoral cells, but also undesirable isolated cells) appear in brown and non-labelled counterstained nuclei are blue. The ratio of labelled tumoral nuclei and non-labelled tumoral ones must be measured on selected regions of interest (i.e. cancer cell nests). The problem is that the undesirable isolated cells intervene in this ratio whereas only nuclei of tumoral cells should be taken into account.

A characteristic of these histological images is that tumoral cells are gathered in clusters. Neighborhood graphs are a powerful tool to delimit these clusters and eliminate isolated cells.

During a first stage, nuclei are segmented following a classical image processing method; we are working on the green color component of the color image and the following image processing plan is applied:

- Elimination of the background by thresholding the image after a smoothing operation.
- Watershed morphological operation taking the minima of the image as germs, in order to get regions corresponding to nuclei.
- Elimination of small regions (area lower to 40 pixels) (figure 8b).

Then a second stage occurs to detect clusters of nuclei, and now, instead of images, graphs are manipulated, the nodes of which correspond to clusters:

- Construction of the DG through the Voronoï diagram. The graph is weighted by the minimal distance between clusters.
- Elimination of long edges in order to separate clusters (more than 10 pixels between nodes) (figure 8c).

As we assume that tumoral clusters have a minimal area, by attributing the nodes of the graph by the area of clusters, clusters of small size can finally be eliminated (figure 8d).

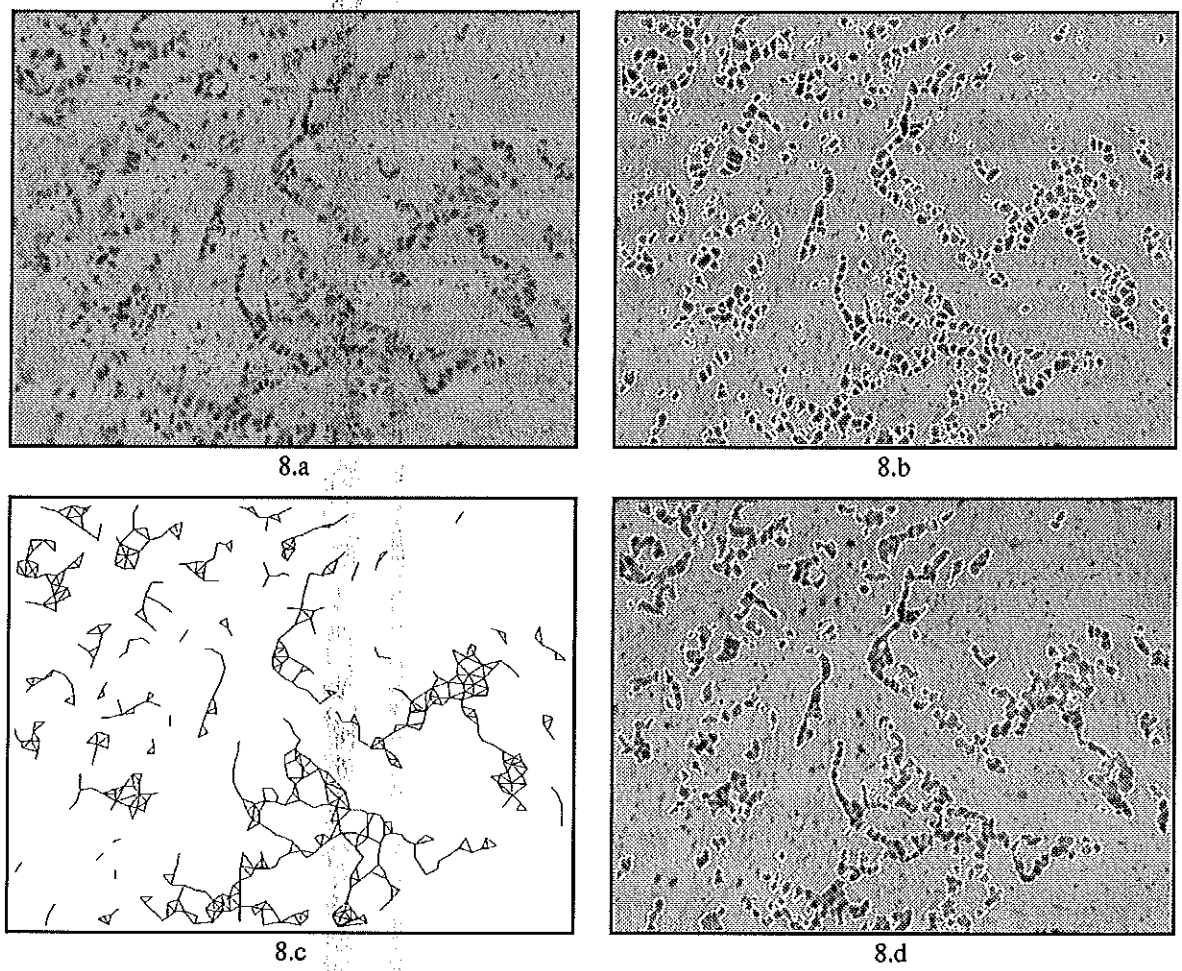


Figure 8: example of cluster detection.

Another interesting aspect of this kind of image is related to the distribution of the labeling over the image. A scattered distribution or a focalized one can result in the same final ratio. Graphs can bring some answer to that problem. Labelled and non-labelled nuclei are attributed with two different values and a distance on the graph is calculated. A distribution of distances is thus obtained and the first-order or second-order histograms can help distinguish between scattered and focalized distributions.

5.2. Hierarchical segmentation

For the segmentation of aerial images (figure 9a), any method based on the gray levels of the image results in an over-segmentation (figure 9b). To improve the segmentation in a significant way, a very simple method consists in working on the adjacency graph of the over-segmented image and weighting edges by the difference of region areas. A first merging step, on regions of similar areas (low weight) reduces segmentation errors (figure 9c). Then a second merging step, on regions of very different areas (high weight), removes small regions (figure 9d).

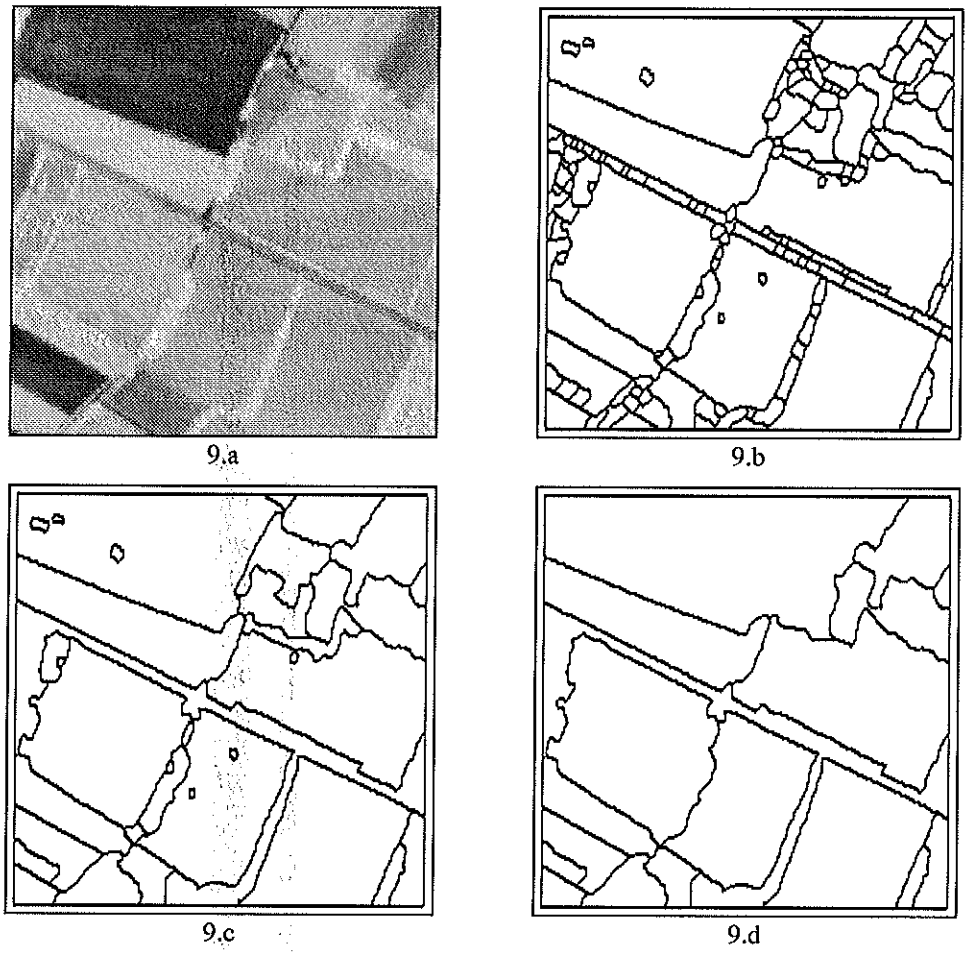


Figure 9: aerial image (a) and segmentation using gray levels (a).
Merging regions of similar areas (c) and of very different areas (d).

This problem of over-segmentation also occurs with another biomedical application dealing with cytological images. These images are characterized by nuclei that biologists must classify. Nuclei must thus be segmented in order to identify them. Here also, over-segmentation can be greatly improved by merging regions of similar size and mean gray level.

5.3. Segmentation based on texture

Graphs are also particularly useful when dealing with textured images. Whereas microscopic texture depends on the gray levels of neighboring points, macroscopic texture is linked to the proximity of broader primitives. The need of manipulating texels has been shown[10], and graphs are an excellent means to represent and manipulate relations between texels.

Once extracted, texels, which constitute the nodes of the graph, can be attributed by the mean gray level and classical characteristics of texture can be calculated. The major advantage of graphs stems here in the possible attribution of nodes by any other function (the texel area for instance), so as to calculate other additional characteristics. The segmentation of the textured image can then be based on the canonical weighting of edges related to any of those characteristics.

6. CONCLUSIONS

After studying graphs and their possible use in image processing, the following points must be emphasized:

➢ When dealing with pre-segmented images, two types of graphs play a prevailing part: neighborhood graphs (the duals Voronoï diagram and Delaunay graph) and adjacency graphs.

➢ Thanks to the atomicity of operators in our object-oriented library HORUS, we could easily introduce a new graph object, together with methods for attributing and weighting the graph, or methods to merge nodes. HORUS operators greatly simplify the tuning of image processing plans. This is the case in our application dealing with histological images.

➢ Sometimes complex sequences of image operators can be replaced by one single operation on graphs. When dealing with hierarchical segmentation, we demonstrated that a basic operator on the adjacency graph such as thresholding could advantageously replace long series of image operators.

As our results on 2D images are quite promising, and thanks to the fact that graph processing can be generalized to higher dimensional spaces, we are now developing a 3D library of operators including graphs (HORUS-3D). This library will enable us to tackle new applications. As new acquisition devices are available (confocal microscopy, positron emission tomography, Magnetic Resonance Imaging, etc.) there is a greater need for efficient graph-based tools and techniques for 3D image processing.

7. ACKNOWLEDGMENTS

Our research on graphs is carried out within the framework of the "Image Processing and Analysis" Centre of Caen. This centre covers a wide domain of applications, dealing with biomedical, as well as material or geographic images.

The authors are grateful to Mrs. P. Herlin of the pathology department of the cancer-research centre F. Baclesse of Caen for providing biomedical images. Special thanks to C. Porquet for her assistance during the translation of the paper.

8. REFERENCES

1. K. Voss, *Discrete Images, Objects and Functions in Z^n*. Algorithms and Combinatories 11, 1993.

2. Fernand Meyer, "Mathematical Morphology: From Two to Three Dimensions", *Journal of Microscopy*, Vol. 165, Pt. 1, pp. 5-28, January 1992.

3. Luc Vincent, "Graphes et morphologie mathématique", *Thèse de Doctorat*, Ecole des Mines de Paris, 1990.

4. Régis Clouard, "HORUS, une bibliothèque et un environnement de programmation d'opérateurs de traitement d'image", *Rapport de recherche du GREYC*, 1996.

5. François Angot, "Graphes de voisinage et traitement d'images", *Rapport du DEA Intelligence Artificielle et Algorithmique*, Université de Caen, 1995.

6. Jean-Marc Chassery and Annick Montanvert, *Géométrie discrète en analyse d'images*, Traité des Nouvelles Technologies, Séries Images, Hermes, 1991.

7. M. Worring and A. W. M. Smeulders, "Multi-Scale Analysis of Discrete point sets", *IEEE*. 0-8186-2915-0/92, 1992.

8. Abderrahim Elmoataz, "Mécanismes opératoires d'un segmenteur non dédié: définition d'une base d'opérateurs minimale et implantation", *Thèse de Doctorat*, Université de Caen, 1990.

9. Karsten Rodenacker, "Invariance of textural features in image cytometry under variation of size and pixel magnitude", *Analytical Cellular Pathology*, 8, 117-133, 1995.

10. Günter Wolf, Michael Beil and Hans Gusk, "Chromatin Structure Analysis Based on a Hierarchic Texture Model", *Analytical and Quantitative Cytology and Histology*, 17: 25-34, 1995.