Chapter 6

Graph Morphology in Image Analysis

Henk Heijmans

Centre for Mathematics and Computer Science
Amsterdam, The Netherlands

Luc Vincent

Xerox Imaging Systems
Peabody, Massachusetts

I. INTRODUCTION

Mathematical morphology can be considered as a set-based approach for the analysis of images [23,24,17]. One of its underlying ideas is to use so-called *structuring elements* to define neighborhoods of points. Recently it has been recognized that these ideas more generally apply to any space V that has the structure of a vector space, or at least a group [11]. In that case one can define a neighborhood of a point $x \in V$ as

$$N(x) = \{x + a \mid a \in A\}$$

where A is the structuring element. In this chapter we describe how many of the concepts of "classical morphology" (i.e., the case where $V = \mathbf{R}^d$ and where $+$ denotes vector addition) can be extended to spaces of images modeled by graphs.

A graph consists of a collection of points, called *vertices*, and a binary relation between them: two vertices either are related or they are not related. This relation is usually represented by a subset $E \subseteq V \times V$ called the *edges*; v and w are related if and only if $(v, w) \in E$. Graphs play an important role in many branches of mathematics and computer science. In the context of image analysis they are often used as a geometric representation of the scene under study. In this case the vertices correspond to the objects in the scene and the edges describe the (neighboring) relations between these objects. In this chapter, the images we consider consist of an underlying graph and a scalar function defined on the set of vertices. In Section II we recall some basic notation and terminology. In Section III we

explain what we mean by a graph representation of an image, and we introduce the notion of a graph operator.

Using the neighboring relations between vertices we are able to propose a large class of morphological operations on a graph. Beyond the basic operations (erosions and dilations, openings and closings), this class embraces almost all of the classical morphological transformations (distance function, skeletons, geodesic transformations, watersheds, etc.); see Section IV. In Section V, we present a number of examples of graph representation of images. There we introduce the Delaunay graph, the Gabriel graph, and the relative neighborhood graph. Furthermore, we illustrate some of the classical morphological transformations for graph-based images.

The previously mentioned class of transformations can be extended by introducing the notion of *structuring graph*. Exactly like structuring elements in classical morphology, structuring graphs act as probes to extract structural information from graphs. They have a simple structure and are relatively small compared to the graph that is to be transformed. The structuring graph is used to construct a neighborhood function on the vertices by relating individual vertices to each other whenever they belong to a local instantiation of the structuring graph. This is explained in Section VI. Then, in Section VII we use these neighborhood functions to define dilations and erosions. Subsequently, Section VIII deals with openings, closings, and other filters. Finally, Section IX is devoted to brief notes on implementation of morphological graph operations and to some concluding comments.

Let us close this introduction with the following important remark:

> Although the theory is presented in the framework of gray-level graphs, all the drawings are "binary" for the sake of clarity and simplicity.

II. MORPHOLOGY FOR FUNCTIONS: CONCEPTS AND BASIC RESULTS

In this section we give a brief overview of morphology for gray-level functions. For general results on mathematical morphology we refer to Serra's books [23,24]. A systematic exposition on gray-level morphology can be found in [23, Chapter 12] and [10].

Although we shall often mean by a gray-level some continuous or discrete numerical value, it may also represent a vector in color space. The only restriction we have to make on the set of gray levels T is that it possesses a complete lattice structure. Recall that T is called a complete lattice if T is a partially ordered set in which every subset S has a least bound $\bigvee S$ called the *supremum* of S, and a greatest lower bound $\bigwedge S$, called the *infimum* of S; see Birkhoff [5].

Let V be an arbitrary set and define $\mathrm{Fun}(V)$ to be the space of all functions $f: V \to T$, where the gray-level set T has a complete lattice structure. If we take

$T = \{0, 1\}$, then Fun(V) is the space of all binary images on V, also represented by $\mathcal{P}(V)$, the power set of V. Other choices for T, sometimes found in the literature, are $T = \{0, 1, 2, \ldots, m\}$, $T = \overline{\mathbf{Z}} = \mathbf{Z} \cup \{-\infty, \infty\}$, $T = \overline{\mathbf{R}} = \mathbf{R} \cup \{-\infty, \infty\}$, $T = [0, 1]$, $T = [0, \infty]$. With the pointwise ordering

$$f \leq g \qquad \text{if } f(v) \leq g(v) \text{ for } v \in V$$

the space Fun(V) becomes a complete lattice. In fact, Fun(V) inherits the complete lattice structure from T. In this chapter we will always assume that $T = \{0, 1, \ldots, m\}$, but we point out that most results carry over to the case where T is an arbitrary complete lattice.

In morphology we are interested in operators mapping the image space into itself.

Definition 2.1. Let ψ be an operator on Fun(V). We say that ψ is
(a) increasing if $f \leq g$ implies that $\psi(f) \leq \psi(g)$,
(b) an erosion if $\psi(\bigwedge_{i\in I} f_i) = \bigwedge_{i\in I} \psi(f_i)$ for an arbitrary family $\{f_i \mid i \in I\}$,
(c) a dilation if $\psi(\bigvee_{i\in I} f_i) = \bigvee_{i\in I} \psi(f_i)$ for an arbitrary family $\{f_i \mid i \in I\}$,
(d) extensive if $\psi(f) \geq f$ for every f,
(e) antiextensive if $\psi(f) \leq f$ for every f,
(f) idempotent if $\psi^2 = \psi$,
(g) a (morphological) filter if ψ is increasing and idempotent,
(h) an opening if ψ is increasing, antiextensive, and idempotent,
(i) a closing if ψ is increasing, extensive, and idempotent.

An important result in morphology says that dilations and erosions always occur in pairs. To any dilation δ there corresponds a unique erosion ε (and vice versa) such that

$$\delta(f) \leq g \Leftrightarrow f \leq \varepsilon(g) \qquad \text{for } f, g \in \text{Fun}(V) \tag{6.1}$$

If ε, δ are operators on Fun(V) such that (6.1) holds, then ε is an erosion, δ is a dilation, and the pair (ε,δ) is called an adjunction. We say that ε and δ are each other's adjoints. If ε and δ are adjoint, then

$$\varepsilon\delta\varepsilon = \varepsilon \qquad \text{and} \qquad \delta\varepsilon\delta = \delta \tag{6.2}$$

There exists yet another duality relation between dilations and erosions. We denote by $t^* = m - t$. Furthermore, we define the "negative" f^* of the function f by

$$f^*(x) = (f(x))^* = m - f(x) \tag{6.3}$$

If ψ is an operator on Fun(V), then the dual operator ψ^* is defined as

$$\psi^*(f) = (\psi(f^*))^* \tag{6.4}$$

Note that this method carries over to any gray-level space T for which there exists an order-reversing bijection. On $T = [0, \infty]$ one may, for example, define

$t^* = 1/t$. The mapping $f \to f^*$ is called a *dual automorphism*. For more details we refer to [10, Section III]. It is easy to see that ψ is increasing if and only if ψ^* is. If δ is a dilation, then δ^* is an erosion and vice versa. For openings and closings there exist similar duality relations.

An important class of function operators is formed by the so-called flat operators [10,25]. Here we shall not give a formal treatment of such operators, but give only a brief sketch of the underlying idea. By a *flat operator* we mean an operator on Fun(V) that is derived from an operator on the power set $\mathcal{P}(V)$ by thresholding. Starting with a function f, one obtains a family of sets X_t by thresholding the function at gray level t, that is, $X_t = \{v \in V \mid f(v) \geq t\}$. One then applies the set operator ψ to this family and uses the transformed family to construct $\psi(f)$. In this chapter we shall deal exclusively with flat operators. In the literature many different names have been proposed for these operators, such as FSP filters (FSP = fuction set processing) [15] or stack filters [35].

Let us conclude this section with some statements concerning flat dilations and erosions. In a sense that we shall not make precise here, the only way to define flat dilations and erosions is by considering neighborhood functions. A neighborhood function on V is a mapping $N : V \to \mathcal{P}(V)$. To any neighborhood function, there corresponds a reciprocal neighborhood function $\check{N}$ given by

$$\check{N}(v) = \{w \in V \mid v \in N(w)\} \tag{6.5}$$

Furthermore, with any neighborhood function one may associate an erosion and a dilation, adjoint to each other, given by

$$\begin{aligned} \delta(f)(v) &= \sup\{f(w) \mid w \in \check{N}(v)\} \\ \varepsilon(f)(v) &= \inf\{f(w) \mid w \in N(v)\} \end{aligned} \tag{6.6}$$

Let $\check{\varepsilon}$ and $\check{\delta}$ be the erosion and dilation corresponding to the reciprocal neighborhood $\check{N}$. One can show that

$$\varepsilon^* = \check{\delta}, \qquad \delta^* = \check{\varepsilon} \tag{6.7}$$

One can also show that every flat dilation and erosion is of the form (6.6).

III. BINARY AND GRAY-LEVEL GRAPHS

In the previous section we outlined the theory for morphological operators on the function lattice Fun(V) where V is an arbitrary set. In this section we are interested in the case that V is the vertex set of a graph. There exist many good textbooks on graphs; we refer in particular to the monograph of Berge [2].

In this section, by graph we always mean a nonoriented graph without loops and multiple edges. A graph G is a mathematical structure consisting of a set of vertices V and edges E. We denote this as $G = (V, E)$. Since edges are supposed

to be simple, they may be represented as a pair of vertices (v, w), denoting that v and w are neighbors. Our assumption that G is undirected can be made explicit by putting $(v, w) = (w, v)$. Let $G = (V, E)$ and $G' = (V', E')$ be two graphs. We say that G is a *subgraph* of G' if $V \subseteq V'$ and $E \subseteq E'$. In literature, the word subgraph is often used in a more restricted sense [2]. By a *homomorphism* from G to G' we mean a one-to-one mapping $\theta : V \to V'$ with the property that $(v, w) \in E$ implies that $(\theta(v), \theta(w)) \in E'$. In that case we say that G and G' are *homomorphic* and write $G \subset G'$. If the homomorphism θ is onto (and hence a bijection), it is called an *isomorphism*. The graphs G and G' are called *isomorphic* if they are related by an isomorphism. We denote this as $G \simeq G'$. An isomorphism from the graph G to itself is called a *symmetry* of G. We denote by $\mathrm{Sym}(G)$ the family of all symmetries of G. Obviously, this family forms a group called the *symmetry group* of G. The identity mapping id, defined by $\mathrm{id}(v) = v$, is contained in $\mathrm{Sym}(G)$ and is called the *trivial symmetry* of G (see Figure 1).

Let $G = (V, E)$ be a graph, and let $f \in \mathrm{Fun}(V)$. Then we call f a gray-level graph. If the gray-level set is $\{0, 1\}$, f is called a binary graph. Sometimes, if we want to emphasize the role of the underlying graph G, we write $(f \mid G)$ instead of f. If $(f \mid G)$ is a gray-level graph and $\tau \in \mathrm{Sym}(G)$, then we define $(\tau f \mid G)$ by $\tau f(v) = f(\tau^{-1}v)$, for $v \in V(G)$. Here $V(G)$ denotes the vertex set associated with the graph G.

Definition 3.1. A graph operator is a mapping which assigns to any graph $G = (V, E)$ an operator $\psi(\cdot \mid G)$ on the function space $\mathrm{Fun}(V)$. A graph operator is called flat if every $\psi(\cdot \mid G)$ is a flat operator. The graph operator ψ will be called G-increasing if ψ increases in G, that is, $\psi(X \mid G) \subseteq \psi(X \mid G')$ for $G \subseteq G'$ and $X \subseteq V(G)$. G-decreasingness of ψ is defined analogously. The graph operator ψ is called increasing if for any graph G the operator $\psi(\cdot \mid G)$ is increasing.

We say that ψ is a *graph erosion* if $\psi(\cdot \mid G)$ is an erosion on $\mathrm{Fun}(V)$ for every graph $G = (V, E)$. Analogously, we define graph dilations, openings, closings, filters, etc. A *graph neighborhood function* is a mapping N that, for every graph G, defines a neighborhood function on the vertex set of G. A graph operator is called *symmetry-preserving* if $\psi(\tau f \mid G) = \tau\psi(f \mid G)$, for $f \in \mathrm{Fun}(V)$, $\tau \in$

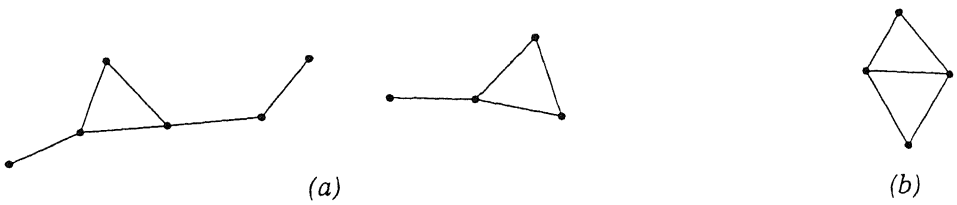


Figure 1. The left graph (a) has only trivial symmetry, whereas the symmetry group of the right graph (b) contains three elements (including the trivial symmetry).

Sym(G), and any graph $G = (V, E)$. Note that this last definition is the analogue of translation invariance in classical morphology. There is a one-to-one correspondence between graph neighborhood functions and flat graph adjunctions. If N is a graph neighborhood function that satisfies $N(\tau v \mid G) = \tau N(v \mid G)$ for $\tau \in$ Sym(G), then the resulting adjunction is symmetry preserving. To illustrate some of these abstract definitions we will present an example; this example is studied thoroughly in the following section.

Example 3.2. We define a graph neighborhood function N in the following way. If $G = (V, E)$ is some graph and $v \in V$, then we define $N(v \mid G)$ as the set containing v as well as all neighbors of v, that is, $N(v \mid G) = \{w \in V \mid (v, w) \in E\} \cup \{v\}$. It is obvious that N is symmetry preserving; that is, $N(\tau v \mid G) = \tau N(v \mid G)$ for every $\tau \in$ Sym(G). As a consequence, the erosion and dilation associated to this neighborhood function are symmetry preserving.

Throughout the remainder of this chapter we shall, whenever no confusion arises, suppress the argument G in the notation of both a gray-level graph and a graph operator.

IV. A SPECIAL CASE: NONSTRUCTURED GRAPH OPERATORS

Before the introduction of the notion of s-graph in [13], the only case of graph morphology that had ever been studied is what could be referred to as nonstructured graph morphology [30,29,32]. As we shall see (Section VI), these nonstructured graph operators turn out to be a special case of the structured ones.

In the present framework, as explained in detail in [30], the morphological operations are directly derived from the distance induced by the set of edges E on the set of vertices V. The graph distance d_G between two vertices v and w is given by the minimal number of edges to cross to go from v to w or, alternatively, by the length of the shortest paths connecting v to w in E. A collection of edges $\pi = (v_1, v_2, \ldots, v_k)$ is called a *path* between v and w if $v_1 = v$, $v_k = w$, and $(v_i, v_{i+1}) \in E$ for $i = 1, \ldots, k$. The *length* of the path is $l(\pi) = k - 1$. So we may write

$$d_G(v, w) = \inf\{l(\pi) \mid \pi \text{ is a path joining } v \text{ and } w \text{ in } E\} \tag{6.8}$$

Since the graph structure under study is not necessarily connected, it may well happen that *no* path connects v to w. In this case, we conventionally put the distance between these two vertices equal to ∞. So, strictly speaking, d_G is not a metric.

Given a vertex $v \in V$ and an integer $n \in \mathbf{Z}^+$, the ball $B_n(v)$ centered at v with radius n is given by

$$B_n(v) = \{v' \in V \mid d_G(v, v') \leq n\}$$

Following Serra [24, Chapters 1–2] and the example mentioned at the end of the previous section, we can now choose the $B_n(v)$'s as neighborhood functions (also called structural mappings, or sometimes structuring functions). These functions associate with each vertex the neighborhood it addresses in a dilation or erosion operation. Note that when $n = 1$, B_1 is exactly the neighborhood function N introduced in Example 3.2.

Dilations and erosions with respect to these neighborhood functions are then defined as follows:

Definition 4.1. Given a gray-level graph f on $G = (V, E)$, the dilation $\delta^{(n)}(f)$ and the erosion $\varepsilon^{(n)}(f)$ of size $n \geq 0$ of f are the gray-level graphs given by

$$\forall v \in V \quad \begin{cases} \delta^{(n)}(f) = \max\{f(v') \mid v' \in B_n(v)\} \\ \varepsilon^{(n)}(f) = \min\{f(v') \mid v' \in B_n(v)\} \end{cases} \tag{6.9}$$

In the sequel, these operations shall be called dilation and erosion "of size n." Intuitively, just as in classical morphology, dilations and erosions are defined as local maxima and minima, respectively. One of the main differences is that, in the present case, the number of vertices in a given neighborhood (ball) $B_n(v)$ is highly dependent on the vertex v. Indeed, such properties as translation invariance are meaningless in a graph.

When $n = 1$ the resulting operations are called elementary dilation and erosion and are simply denoted δ and ε. As in classical morphology, we have the following properties:

$$\delta^{(n)} = \underbrace{\delta \circ \delta \circ \cdots \circ \delta}_{n \text{ times}}, \tag{6.10}$$

$$\varepsilon^{(n)} = \underbrace{\varepsilon \circ \varepsilon \circ \cdots \circ \varepsilon}_{n \text{ times}} \tag{6.11}$$

Thus, operations involving large neighborhoods can be decomposed into a succession of elementary operators. This is taken into account for the actual implementation of nonstructured graph dilations and erosions (see Section IX).

To illustrate the effect of these operations on graphs, we shall use a binary example: suppose that the gray-level graph f under study takes its values in $\{0, 1\}$, and let $n \geq 0$ be an integer. In this case, performing a dilation of size n of f comes down to giving value 1 to each vertex v with value 0 (i.e., such that $f(v) = 0$) having a vertex with value 1 in its neighborhood $B_n(v)$:

$$f(v) = 0 \quad \text{and} \quad \exists v' \in B_n(v), \qquad f(v') = 1 \Rightarrow \delta^{(n)}(f)(v) = 1 \tag{6.12}$$

$$f(v) = 1 \Rightarrow \delta^{(n)}(f)(v) = 1 \tag{6.13}$$

By duality, eroding f amounts to giving value 0 to each 1-vertex v having a 0-vertex in its neighborhood $B_n(v)$. Figure 2 illustrates the effect of an elementary

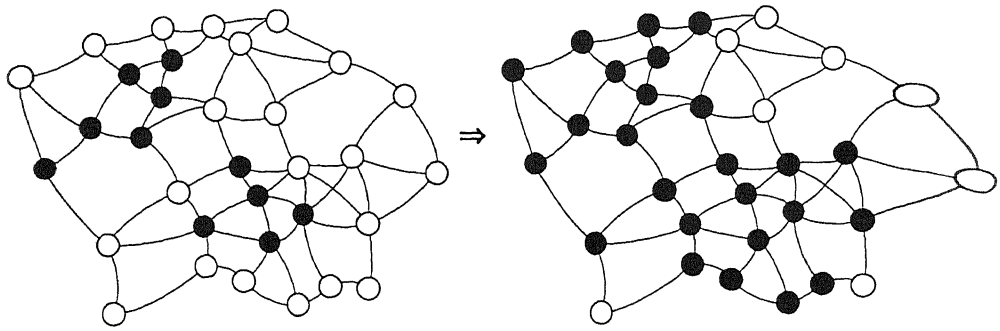


Figure 2. Nonstructured elementary dilation of a binary graph.

dilation of a binary graph. In this figure, the vertices with value one are represented in black.

In the present "nonstructured framework," beyond these basic operations and their associated openings and closings, the underlying distance d_G is particularly interesting because it allows us to define numerous more advanced transformations [28,30]. We cite among others,

- Distance functions
- Granulometries
- Skeletons, skeleton by influence zones (SKIZ)
- Catchment basins, watersheds
- Geodesic operators, reconstruction

These graph transformations, described and illustrated in [30], turn out to be particularly interesting in practice, as will be illustrated in the next section for distance functions, granulometries, and watersheds.

V. GRAPH REPRESENTATION OF IMAGES, EXAMPLES OF APPLICATION OF NONSTRUCTURED GRAPH TRANSFORMATIONS

A. Modeling of Neighborhood Relationships

At this point one may ask, why define all these morphological transformations? What concrete objects will they be applied to? What kind of problems will they help solve? It is now time to address these issues.

The initial motivation of graph morphology was the study of neighborhood relationships within populations of objects [30]. The idea is to model a set of objects V as the vertices of a graph and to process this graph via morphological

transformations to extract useful information. For example, in histology, assuming that the objects of V represent cells in a tissue, it seems reasonable to model such a population as the vertices of a neighborhood graph [30,32]. This graph provides plausible relationships between cells and can be chosen to be relatively independent of the deformations of the tissue itself. Using tools described in the previous section, one can then try to characterize quantitatively such notions as

- Average number of neighbors of a cell.
- "Isolated" cells of a given type (with respect to the underlying graphs).
- Size distribution of the clusters of cells of a given type, or cells sharing common characteristics. Such analyses can be done in the graph itself, which means that the actual distances between cells do not matter; only the graph distance is accounted for, through the use of granulometries on the graph.
- Average distance in the graph between two cells of a given type, closest distance between a cell of type A and a cell of type B, etc. (use of distance functions on graphs).

In fact, graph morphology operations have already been successfully used in histology for the study of germinal centers [22]. The same kind of approach can be used in various problems involving the quantitative description of spatial relationships between objects.

B. Definition of Appropriate Neighborhood Graphs

For the category of aforementioned neighborhood problems, the first step is to define and construct a neighborhood graph from a two-dimensional (or even n-dimensional) population of objects. This initial population is often available under the form of a discrete binary image whose connected components represent the objects under study. It usually results from a previous segmentation stage.

This modeling purpose is generally best served by the neighborhood graphs of the Delaunay triangulation family (see, for example, [20]), namely the *Delaunay triangulation* (DT) itself, the *Gabriel graph* (GG) [8], and the *relative neighborhood graph* (RNG) [27]. Indeed, these graphs do not depend on any parameter such as a maximal distance between objects or a minimal number of neighbors, a property that is very useful in practice [32]. This property implies in particular that these graphs are independent of any scaling and can therefore be used equally well for various kinds of populations. In addition, DTs, GGs and RNGs are connected graphs, are planar, and are included in one another, thus enabling a modeling of neighborhood relationships of increasing strength.

These graphs are defined from the well-known Voronoï diagram (see, e.g., [20, § 5.5]). Let us assume for simplicity that the objects to be modeled are points $p_1, p_1, \ldots, p_n$ in the continuous place $\mathbf{R}^2$. Recall that the *Voronoï polygon* associated with p_i, denoted $V(p_i)$, is given by

$$V(p_i) = \{p \in \mathbf{R}^2 \mid \forall j \neq i,\ d(p,p_i) < d(p,p_j)\} \tag{6.14}$$

The set of the boundaries of these Voronoï polygons is called the Voronoï diagram (see Figure 3).

This definition easily extends to the discrete framework and to the case where the objects to be modeled are no longer isolated points. One often speaks then of influence zones and skeleton by influence zones (SKIZ).

The definitions of DT, GG, and RNG follow straightforwardly. Let $V = \{p_i \mid i = 1, \ldots, n\}$ be the initial set of points.

Definition 5.1. The Delaunay triangulation of V is the graph $\mathcal{G}_{dt} = (V, E_{dt})$ such that E_{dt} is the set of the point pairs (p_i,p_j) whose associated Voronoï polygons are adjacent, i.e., share an edge.

When V does not contain any cocircular 4 points, one can show that DT is effectively a triangulation that is connected and planar.

To define the Gabriel graph and the relative neighborhood graph, it is convenient to start from two regions associated with a pair (p,q) of points in the plane: $D(p,q)$ denotes the *closed* disk having $[p,q]$ as a diameter and $Cr(p,q)$ is the intersection of the two *open* disks of radius $\overline{pq}$ respectively centered in p and q, sometimes called the *crescent*. These notions are illustrated in Figure 4. Note that $D(p,q)\backslash\{p,q\} \subseteq Cr(p,q)$.

Definition 5.2. The Gabriel graph $G_{gg} = (V, E_{gg})$ of V and the relative neighborhood graph $G_{rng} = (V, E_{rng})$ of V are such that

E_{gg} is the set of the point pairs (p_i,p_j), with $p_k \notin D(p_i,p_j)$, if $k \neq i, j$.
E_{rng} is the set of the point pairs (p_i,p_j), with $p_k \notin Cr(p_i,p_j)$, if $k \neq i, j$.

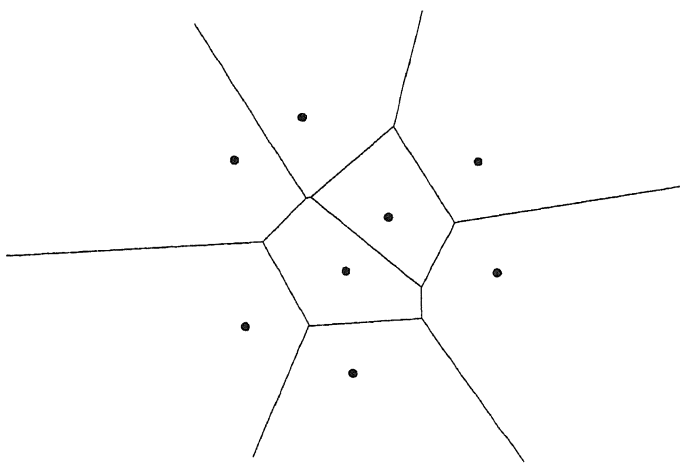


Figure 3. Example of Voronoï diagram.

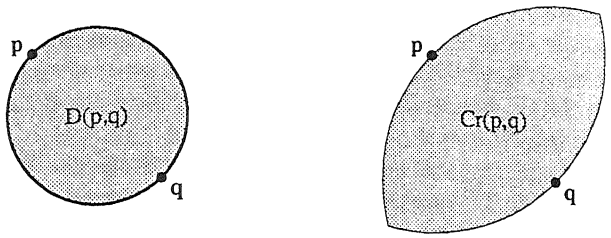


Figure 4. Regions on which the definitions of Gabriel graphs and relative neighborhood graphs are based.

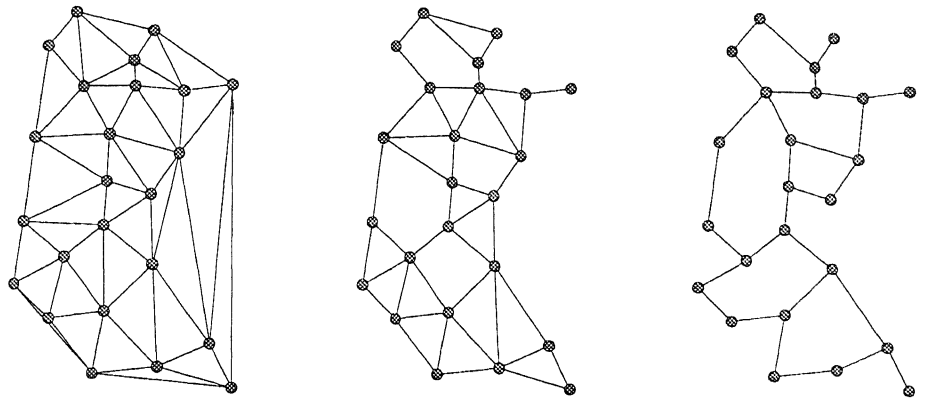


Figure 5. Three different neighborhood graphs stemming from the Voronoï diagram.

These three different graphs are illustrated by Figure 5. Obviously, the following inclusion relations hold:

$$E_{\text{rng}} \subseteq E_{\text{gg}} \subseteq E_{\text{dt}} \tag{6.15}$$

Definitions 5.1 and 5.2 can be extended to connected components of arbitrary size and shape in the plane [32, pp. 119–120].

C. Computation of DT, GG, and RNG

The typical algorithms for determining these neighborhood graphs in practice rely on computational geometry techniques [20]. When the connected components (objects) of the population are relatively circular or small in comparison to the distance between them, they can be assimilated to isolated points. One can then first apply some well-known Voronoï diagram algorithms, like the "divide-

and-conquer" approach described in [20, Chapter 5] or incremental techniques such as that proposed in [6]. These algorithms run in at most $O(n \log(n))$ time, n being the number of points. DT and GG can then be derived in a straightforward (and linear) manner. For RNG, the situation is much more complex, but it can anyhow be derived in $O(n \log(n))$ from the Voronoï diagram [26].

However, regardless of the computational efficiency of these methods, they are rather limited by the fact that they work only with isolated points as input data. Indeed, when the objects of the population are arbitrary connected components of a discrete binary image, they are to be modeled by polygons and the complexity of the associated algorithms becomes horrifying! Therefore, it seems much more appropriate to make use of digital techniques introduced in [30,28] and detailed in [32]. Let us now briefly describe and illustrate these algorithms.

For this purpose, we shall start from Figure 6a, whose connected components will be the vertices of our graphs. The successive steps of the algorithm are as follows:

1. Labeling of the connected components of the original image: each of them will be assigned a unique number. This labeling can be accomplished extremely efficiently by using, e.g., the first-in-first-out (FIFO)–based algorithms described in [34].
2. Computation of the labeled influence zones of these connected components. Roughly speaking, the labels associated with each component are propagated in the image until they completely fill up the remaining space, yielding

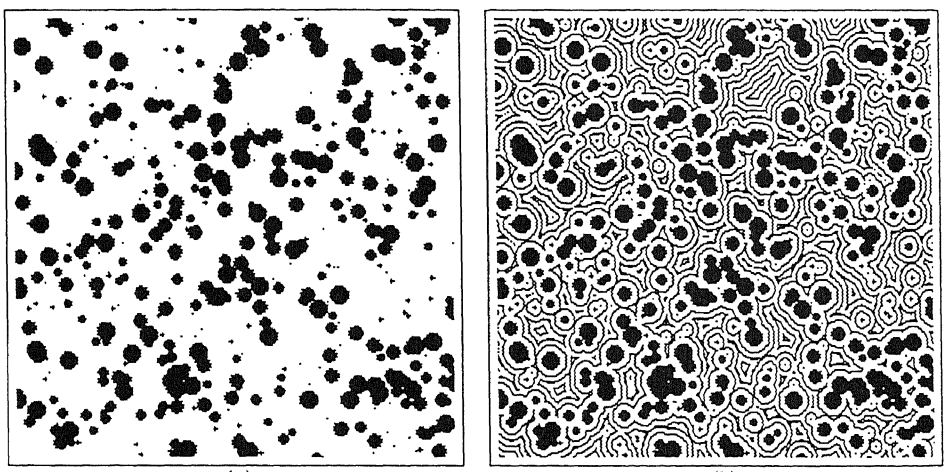


Figure 6. (a) Original binary image with several connected components and (b) level lines of the distance function of its background.

a digital equivalent of the Voronoï diagram. This computation has to be performed as accurately as possible (see [32, Chapter 5]), and discrete distances such as the city block, chamfers, octagonal, or hexagonal distances [7] are generally not good enough. Here we shall make use of an exact Euclidean distance function algorithm described in [33] (see Figure 6b). The associated labeled influence zones are displayed in Figure 7a. The boundaries of these zones constitute the actual SKIZ (see Figure 7b).

3. Contour tracking of the influence zones. By tracking the contours of zone with label i, one gets successively all the labels of the neighboring zones. This allows an easy computation of the "discrete" Delaunay triangulation of our initial binary image. By using some additional constraints detailed in [32], one gets the Gabriel graph in the same way. The Delaunay triangulation and Gabriel graph corresponding to Figure 6a are displayed in Figure 8. Similar techniques also enable us to derive discrete RNGs starting from arbitrary connected components.

D. Examples of Application

Besides the histology applications mentioned earlier, graph morphology has been used for very different problems. To give the reader some flavor we shall briefly mention some of them below.

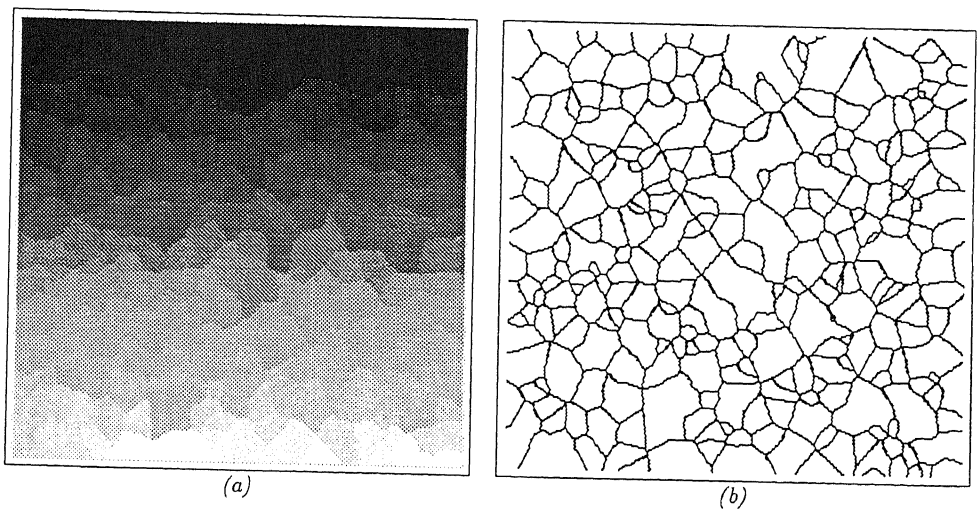
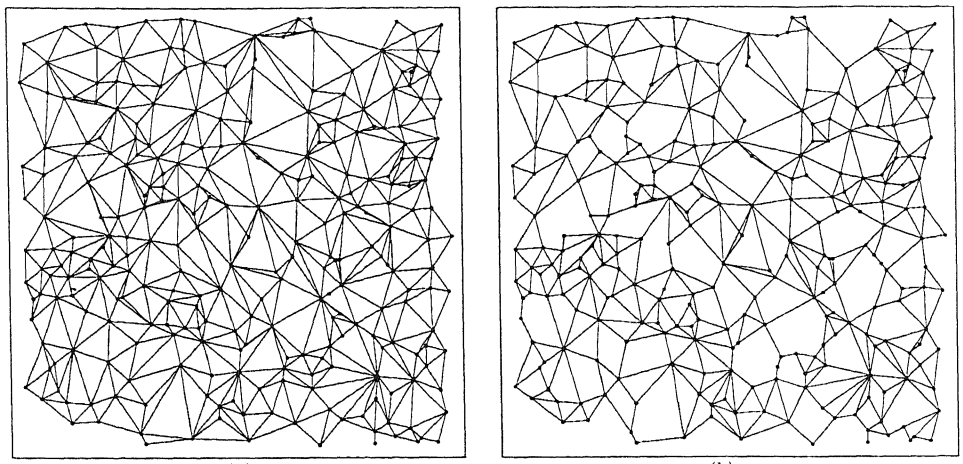


Figure 7. (a) Labeled Euclidean influence zones of the connected components of Figure 6a; different shades of gray represent different influence zones. (b) Corresponding Voronoï diagram: boundaries of these influence zones.



(a) (b)

Figure 8. (a) Delaunay triangulation and (b) Gabriel graph corresponding to Figure 6a.

1. Fracture Simulations in Porous Media

Graph morphology provides nice tools for the study of heterogeneous media at a macroscopic level, based on information on their microstructure [14]. Typically, one starts from a digitized picture of a medium and models its microstructure as a graph. This graph can be either of DT or GG type (see previous section) or based on the underlying discrete grid, depending on the kind of information that is to be extracted.

Here we are concerned with the study of crack propagation in propous media (see [31]). We assume that we have a binary picture of the medium (pores, value 1; matrix, i.e., medium itself, value 0; see Figure 9a) and that a traction is exerted on it. Under this force, cracks will appear in the material, and we are interested in finding out what the crack paths look like, what their lengths are, etc. The assumption underlying our approach is that crack paths tend to go preferentially through the pores of the material. Moreover, assuming a crack has reached a particular pore p, its next propagation step will most probably be one of the pores in the immediate neighborhood of p. Therefore, we first model the set of pores as the vertices of a neighborhood graph and choose a specific vertex or set of vertices V_I as crack initiations. Then we actually simulate the propagation of a crack in the graph starting at the vertices of V_I.

The graph that seems most relevant to this kind of problem is the Gabriel graph described in the previous section. For the actual simulation, one uses distance functions on this graph (see [14,31]): the graph distance function asso-

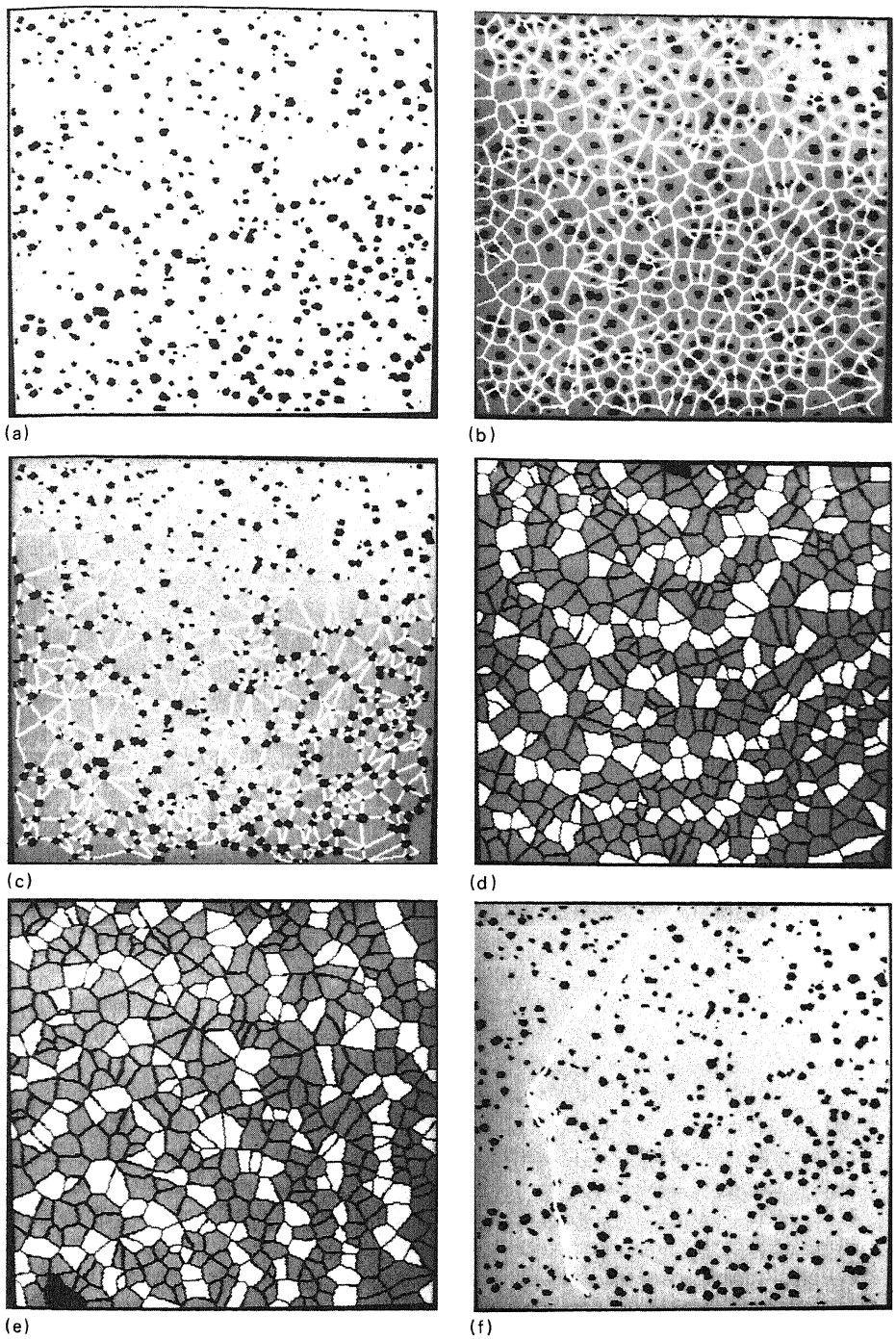


Figure 9. Using distance functions on a Gabriel graph to extract possible fracture lines in a porous material.

ciated with the set V_I is first determined; that is, to each vertex (pore), one assigns its distance to the closest crack initiation. This allows one to find the vertices V_E at one edge of the material that are first reached by a "wave" starting from V_I. Next, a backward distance function (back-propagation of a wave starting from V_E) allows one to determine the actual crack paths between V_I and V_E.

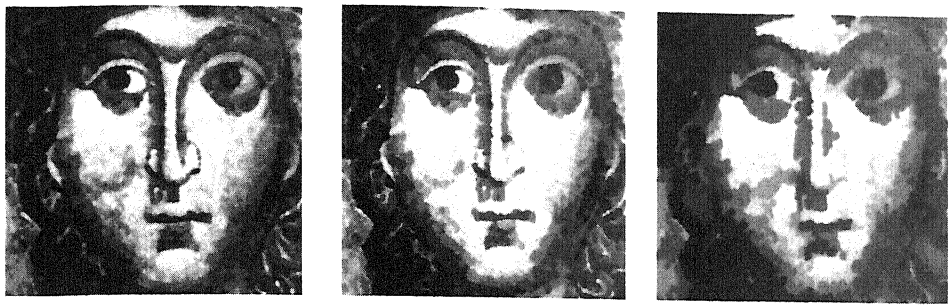
This technique has been tested on concrete examples as well as artificially designed random media and gives very promising results. It is illustrated in Figure 9 for the case of a medium made of graphite nodules (the pores) disconnected from a pig iron matrix. The Gabriel graph of the nodules (Figure 9c) is derived from the Voronoï diagram of Figure 9b. The forward and backward distance functions are then displayed modulo 2, each vertex being represented as its associated Voronoï zone, for the sake of clarity. Lastly, the extracted crack paths are shown in Figure 9f.

2. Hierarchical Representation and Segmentation of Images

Image segmentation is one of the most common problems in the field of image processing. A task that is often related to image segmentation is concerned with the "hierarchization" of an image, that is, the production of a series of images with decreasing level of detail: between two successive images of the series, details of least importance are suppressed while the important features are preserved. These two related issues can be approached in a common way by means of watersheds on images and graphs [29,32].

As explained in further detail in [29], the watershed transformation, whose use is more and more common in image analysis, associates with every minimum of a picture its *catchment basin*, that is, its influence zone. Computing the watersheds of the gradient of an image I allows one to decompose I into regions, each of which corresponds to a perceptually relevant feature. Unfortunately, due to noise, one often observes an oversegmentation; the regions into which the picture has to be decomposed are fragmented, sometimes very badly. For example, Figure 10b has been obtained by computing the catchment basins of the gradient of Figure 10a and assigning to each basin the mean gray level of the corresponding pixels in the original image. This image is often referred to as a mosaic image. The oversegmentation of Figure 10b can be clearly noted.

To get rid of this problem while producing a series of images with decreasing level of detail, the method originally proposed in [29] considers the adjacency graph of the catchment basins. A morphological gradient of this graph G is easily produced, for example, by computing the transform $\max(\delta(G) - G, G - \varepsilon(G))$. Then determining the watersheds of this graph allows one to merge catchment basins into catchment basins of second order. The resulting image, called a mosaic image of order 2 (see Figure 10c), has less detail than the previous one but the main features have been preserved. The procedure can be iterated to produce mosaic images of order 3, 4, etc. Unlike the classical Gaussian pyramids the present method has the advantage that it avoids blurring effects and preserves the most significant contours at best.



(a) (b) (c)

Figure 10. Successive image simplification by watersheds on graphs.

3. Study of Cornea Cell Populations

This is one of the medical applications in which graph morphology is currently under investigation. Images of the type shown in Figure 11* represent populations of cornea cells. The cells have roughly polygonal shapes, and the main problem is to determine the distribution of the number of edges of each cell. Beyond that, one would also like to answer such questions as: Do the small cells tend to have small cells as neighbors? What is the size of clusters of cells of a given type? The same questions can also be asked for the cells with few (or with many) edges.

To address these issues, we use the adjacency graph of the cells, which is obtained after watershed segmentation of the image in Figure 11 and contour tracking of each extracted region. As can be seen in Figure 12, this graph is a triangulation. Determination of the number of neighbors of a given vertex yields the number of edges of the associated cell in a straightforward manner. One can then assign to each cell its number of edges, thus producing a gray-level graph. Granulometries on this graph then provide useful information on the repartition of these cells in the tissue. The same analyses can be performed by assigning each cell its size or any other relevant parameter [30].

VI. STRUCTURING GRAPHS AND NEIGHBORHOOD FUNCTIONS

The basic idea underlying classical morphology is to extract information from an image by probing it at any position with some small geometric shape called a structuring element. Using operations related to the partial order of the underlying image space (e.g., supremum, infimum), one may construct a large class of image operators that are translation invariant. This approach easily carries over to gray-level graphs if one introduces the concept of a *structuring graph* or *s-graph*.

*Example provided by Dr. Barry Masters, USUHS, Bethesda, Maryland.

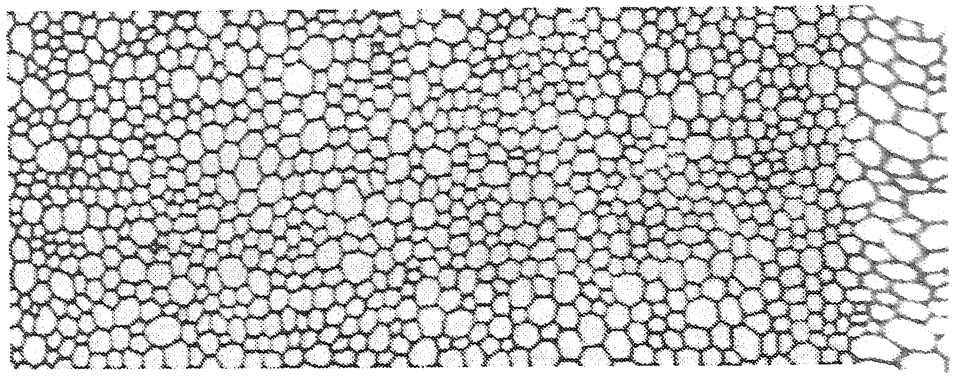


Figure 11. Population of cornea cells at high magnification.

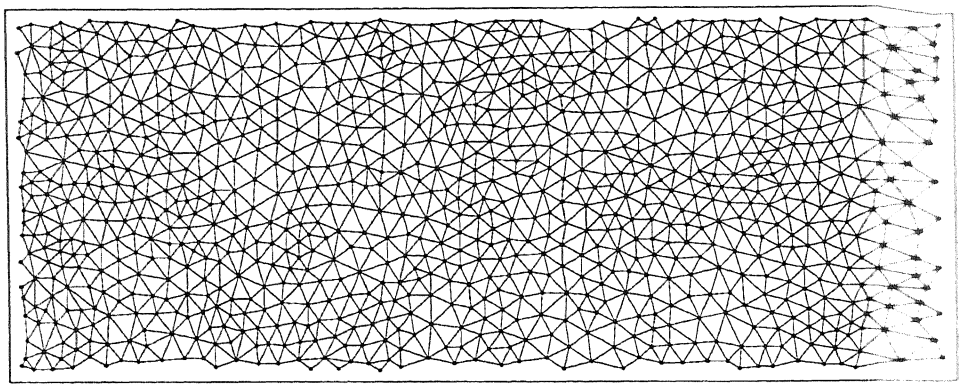


Figure 12. Adjacency graph corresponding to Figure 11.

Definition 6.1. An s-graph $\mathcal{A}$ consists of a graph $G_{\mathcal{A}} = (V_{\mathcal{A}}, E_{\mathcal{A}})$ and two nonempty subsets $B_{\mathcal{A}}, R_{\mathcal{A}} \subseteq V_{\mathcal{A}}$, respectively called the *buds* and the *roots*.

Matching an s-graph $\mathcal{A}$ to the graph G at vertex v amounts to finding a homomorphism θ mapping $G_{\mathcal{A}}$ into G such that $v \in \theta(R_{\mathcal{A}})$. Such a mapping θ is called an *embedding of* $\mathcal{A}$ *into* G *at* v. (*Note:* We point out that we use the word "matching" in a different meaning than is usual in the current literature on graphs.)

We can use an s-graph $\mathcal{A}$ to construct for any given graph $G = (V, E)$ a neighborhood function $N_{\mathcal{A}}$ on $\mathcal{P}(V)$ as follows:

$$N_{\mathcal{A}}(v \mid G) = \bigcup\{\theta(B_{\mathcal{A}}) \mid \theta \text{ is an embedding of } \mathcal{A} \text{ into } G \text{ at } v\} \qquad (6.16)$$

Here the second argument G indicates the dependence on the underlying graph G. It is obvious that

$$N_{\mathcal{A}}(\tau v \mid G) = \tau N_{\mathcal{A}}(v \mid G) \qquad \text{for every } \tau \in \mathrm{Sym}(G)$$

In Figure 13 we have illustrated the concept of an s-graph and the corresponding neighborhood function. In this figure and the following ones, roots of s-graphs are designated by arrows and buds are drawn in bold. Comparing this construction of a neighborhood function with classical translation-invariant morphology, where the neighborhoods are translates of a small set called the structuring element, the roots of the s-graph correspond to the origin of the structuring element (note that an s-graph may have more than one root) and the buds to the points of the structuring element. An important difference from classical morphology, however, is that for graphs the neighborhood structure may differ at each vertex so that the s-graph must prescribe the structure near a vertex.

Figure 13 shows that the neighborhood determined by an s-graph $\mathcal{A}$ depends on several factors. Adding points to the bud set (s-graph $\mathcal{B}$) or to the root set (s-graph $\mathcal{C}$) or decreasing the underlying graph (s-graph $\mathcal{D}$) has the effect that the neighborhood increases. This motivates us to define a partial order $\leq$ on the collection of all s-graphs which formalizes this observation. For two s-graphs $\mathcal{A}$,

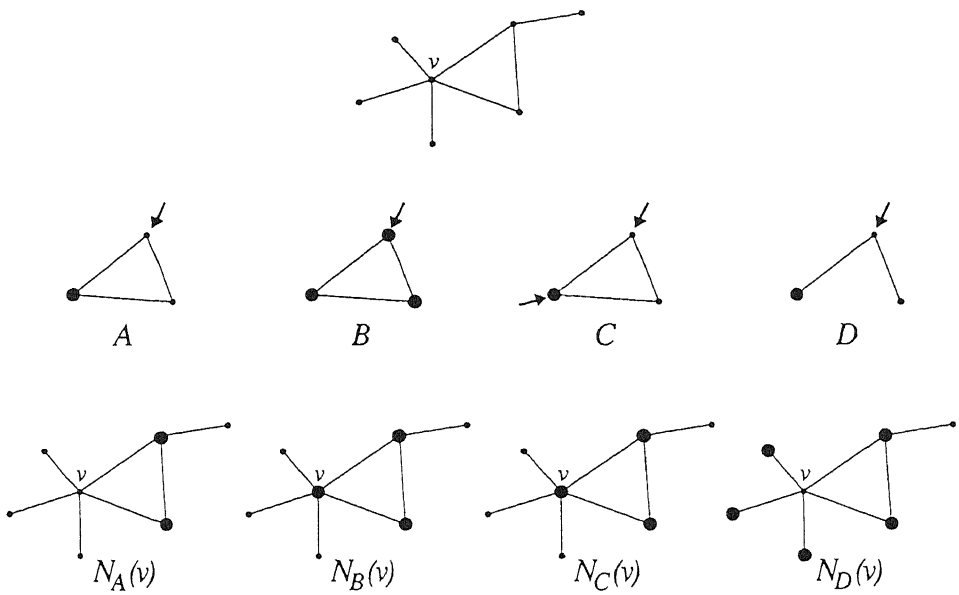


Figure 13. s-graphs A, B, C, D and their corresponding neighborhood functions at vertex v.

$\mathcal{B}$ we say that $\mathcal{A} \leq \mathcal{B}$ (stated as $\mathcal{A}$ is more selective than $\mathcal{B}$) if $N_{\mathcal{A}}(v \mid G) \subseteq N_{\mathcal{B}}(v \mid G)$ for any graph G and any vertex v on that graph. If $\mathcal{A} \leq \mathcal{B}$ and $\mathcal{B} \leq \mathcal{A}$, then $\mathcal{A}$ and $\mathcal{B}$ are equivalent and we write $\mathcal{A} \equiv \mathcal{B}$. In the example depicted in Figure 13 we have $\mathcal{A} \leq \mathcal{B}$, $\mathcal{B} \equiv \mathcal{C}$, $\mathcal{A} \leq \mathcal{D}$. In [13] we have shown the following result.

Proposition 6.2. Let $\mathcal{A}$, $\mathcal{B}$ be s-graphs. Then we have $\mathcal{A} \leq \mathcal{B}$ if and only if

(i) $G_{\mathcal{B}} \tilde{\subset} G_{\mathcal{A}}$

(ii) $N_{\mathcal{A}}(v \mid G_{\mathcal{A}}) \subseteq N_{\mathcal{B}}(v \mid G_{\mathcal{A}})$, for any $v \in R_{\mathcal{A}}$.

In particular, $\mathcal{A} \equiv \mathcal{B}$ if and only if $G_{\mathcal{A}} \simeq G_{\mathcal{B}}$ and $N_{\mathcal{A}}(v \mid G_{\mathcal{A}}) = N_{\mathcal{B}}(v \mid G_{\mathcal{A}})$ for any $v \in R_{\mathcal{A}}$.

In section II we have seen that to any neighborhood function N on the set V there corresponds a unique reciprocal neighborhood function $\check{N}$. So if $\mathcal{A}$ is an s-graph and $G = (V, E)$ a graph, then there exists a reciprocal neighborhood $\check{N}$ of the neighborhood function $N_{\mathcal{A}}(\cdot \mid G)$. One may wonder if there exists an s-graph $\mathcal{B}$ such that the reciprocal neighborhood function of $N_{\mathcal{A}}(\cdot \mid G)$ equals $N_{\mathcal{B}}(\cdot \mid G)$ for any graph G. In [13] we have shown that one can give an affirmative answer to this question by defining the so-called reciprocal s-graph $\check{\mathcal{A}}$, as follows:

$$G_{\check{\mathcal{A}}} = G_{\mathcal{A}}, \qquad B_{\check{\mathcal{A}}} = R_{\mathcal{A}}, \qquad R_{\check{\mathcal{A}}} = B_{\mathcal{A}}$$

see Figure 14a. Then we have the relation

$$\check{N}_{\mathcal{A}}(v \mid G) = N_{\check{\mathcal{A}}}(v \mid G)$$

If the s-graph $\mathcal{A}$ coincides with its reciprocal, or more precisely if $\mathcal{A} \equiv \check{\mathcal{A}}$, then we say that $\mathcal{A}$ is *symmetric*. Some examples can be found in Figure 14b.

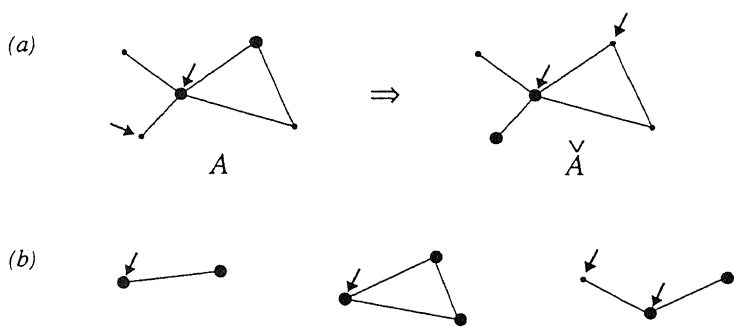


Figure 14. (a) An s-graph A and its reciprocal $\check{A}$; (b) symmetric s-graphs.

VII. DILATIONS AND EROSIONS

In Section II we indicated how to define gray-level dilations and erosions using neighborhood functions. In combination with the construction method for neighborhood functions from s-graphs described in the previous section, we have found a systematic way to build graph dilations and graph erosions from s-graphs.

Let $\mathcal{A}$ be an s-graph and let $N_{\mathcal{A}}$ be its corresponding neighborhood function. Consider an arbitrary graph $G = (V,E)$. Then $\delta_{\mathcal{A}}$ and $\varepsilon_{\mathcal{A}}$, given by

$$\begin{aligned} \delta_{\mathcal{A}}(f)(v) &= \sup\{f(w) \mid w \in N_{\mathcal{A}}^{\vee}(v \mid G)\} \\ \varepsilon_{\mathcal{A}}(f)(v) &= \inf\{f(w) \mid w \in N_{\mathcal{A}}(v \mid G)\} \end{aligned} \tag{6.17}$$

for $f \in \mathrm{Fun}(V)$, define a graph dilation and a graph erosion, respectively, and the pair $(\varepsilon_{\mathcal{A}},\delta_{\mathcal{A}})$ forms an adjunction. Furthermore, both operators are symmetry preserving. They are illustrated in Figure 15.

Recall from Section II that the dual of an operator ψ on $\mathrm{Fun}(V)$ is defined as $\psi^*(f) = (\psi(f^*))^*$, where $f^* = m - f$ is the negative of f. Since the mapping $f \to f^*$ turns suprema into infima and vice versa, one may conclude that the dual

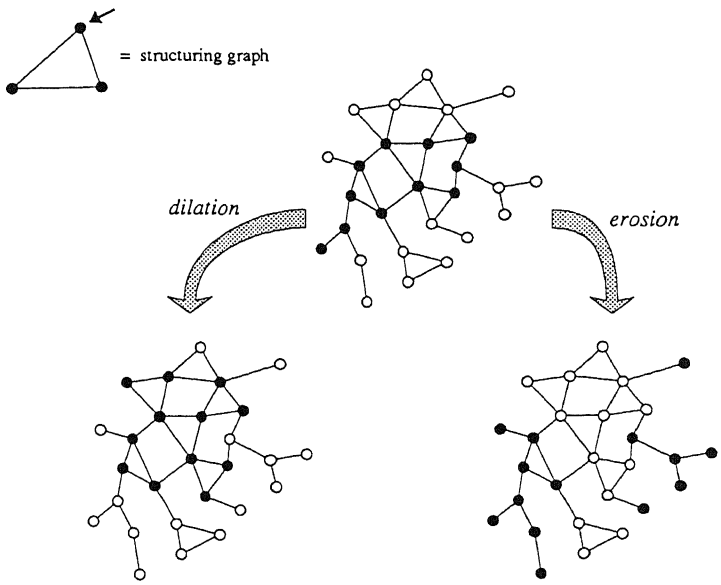


Figure 15. Dilation and erosion of a graph with respect to an s-graph.

of a dilation is an erosion and, conversely, that the dual of an erosion is dilation. In [13] we have established the following result.

Proposition 7.1. For any s-graph $\mathcal{A}$ we have

$$\delta^*_{\mathcal{A}} = \varepsilon_{\check{\mathcal{A}}} \qquad \text{and} \qquad \varepsilon^*_{\mathcal{A}} = \delta_{\check{\mathcal{A}}}$$

We point out that a similar property holds in classical morphology. Although the resemblances between graph morphological operators and the classical translation-invariant morphological operators are striking, there are also some important differences. In particular, it is well known that any translation-invariant dilation δ on the binary image space $\mathcal{P}(\mathbf{R}^d)$ is a Minkowski addition, that is, $\delta(X) = X \oplus \delta(\{0\})$. A similar result holds for gray-level dilations. Unfortunately, there exists no graph analogue of this fact. This means in particular that compositions or suprema of dilations using one structuring graph cannot be obtained using only one (larger) s-graph. This is due to the fact that the local graph structure near a vertex may be very diverse, and therefore the neighborhood determined by an s-graph depends not only on the number of buds and roots but also on the structure of the s-graph itself. This is quite different from classical morphology, where this local structure is independent of the position and therefore plays no role.

As a second distinction between graph and classical morphology, we note that Matheron's theorem, which, in the classical case, states that every increasing translation-invariant operator can be decomposed as an intersection of dilations or as a union of erosions, does not have an analogue in graph morphology. This follows immediately from the following considerations. The s-graph construction of a neighborhood function on the vertex set of a graph is not the most general method for obtaining neighborhood functions that are invariant under the symmetries of the graph. In fact, the s-graph approach requires a certain amount of local structure to be present near a vertex. One may construct more general neighborhood functions by requiring in addition that the local structure contents near a vertex does not exceed a certain amount. For example, we may define

$$N(v) = \{v\} \cup \{w \in V \mid (v, w) \in E\} \qquad \text{if } v \text{ has at most two neighbors}$$

and

$$N(v) = \{v\} \qquad \text{otherwise}$$

Such a construction gives rise to graph neighborhood functions (and hence graph dilations) that are not G-increasing in general; see Figure 16. In particular, the operators resulting from this construction are symmetry preserving but cannot be written as an infimum of graph dilations of the form $\delta_{\mathcal{A}}$.

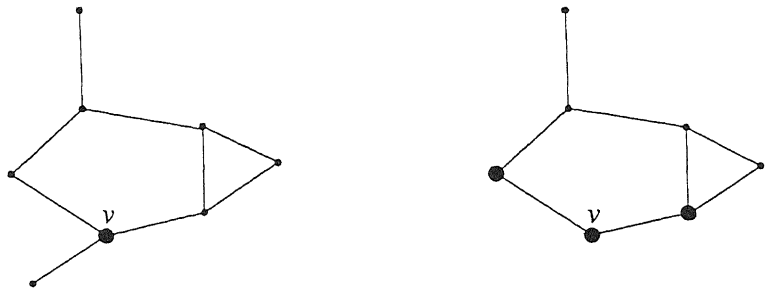


Figure 16. A neighborhood function that is not G-increasing; see text.

VIII. OPENINGS, CLOSINGS, AND OTHER FILTERS

An operator is called a *morphological filter* if it is increasing and idempotent. Idempotence is an important property for an operator because it means that repeated application of such an operator has no further effect on the outcome. In a sense, one could argue that any operator (morphological or otherwise) designed to clean noise from an image has to be applied repeatedly until the result remains constant. In practice, such an iterative procedure results in idempotent operators; see Heijmans and Serra [12]. A formal theory for morphological filters has been designed by Matheron [16]. In this section we shall apply some of his results to the framework of gray-level graphs. Although we shall mainly be concerned with openings and closings and construction methods for such operators, we will consider alternating sequential filters at the end.

Openings and closings lie at the heart of the theory of morphological filters. Here we shall only consider openings. The corresponding results for closings follow easily by duality in the following way: if ψ is an opening then ψ^* is a closing and vice versa. We refer to [13, Remark 6.5] for some difficulties concerning the definition of closings in the graph framework.

A well-known construction of openings is to compose an erosion and its adjoint dilation. For instance, $\delta_{\mathcal{A}}\varepsilon_{\mathcal{A}}$ is an opening for any s-graph $\mathcal{A}$. In Figure 17 we have depicted an example of an opening obtained in this way.

A second way to build openings, closely related to the previous one, also uses s-graphs. We use the following notation: if $G = (V, E)$ is a graph, $X \subseteq V$ and $t \in T$, then we define $\mathcal{F}_t(X)$ as the gray-level graph that equals t at the vertices in X and $\inf T$ elsewhere. Let $\mathcal{A}$ be an s-graph. We define the graph operator $\alpha_{\mathcal{A}}$ by

$$\alpha_{\mathcal{A}}(f) = \vee\{\mathcal{F}_t(\theta(B_{\mathcal{A}})) \mid G_{\mathcal{A}} \xrightarrow{\theta} G \text{ and } \mathcal{F}_t(\theta(B_{\mathcal{A}})) \leq f\}$$

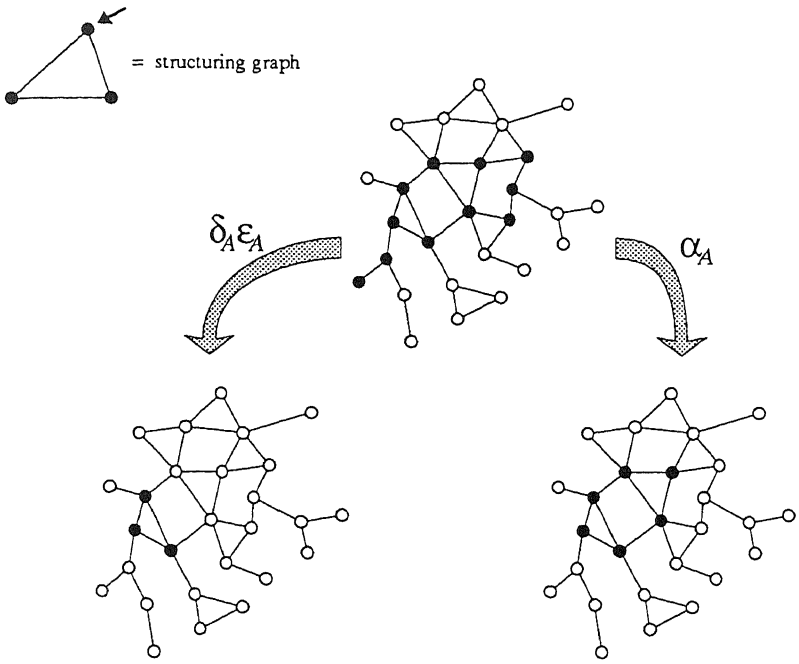


Figure 17. Two different openings of a binary graph with respect to a given s-graph.

This expression looks rather complex at first sight, but it becomes much simpler for binary graphs $(X \mid G)$ where $X \subseteq V$. In that case $\alpha_{\mathcal{A}}(X \mid G)$ is the union of all embedded bud sets $\theta(B_{\mathcal{A}})$ where θ is an embedding of $\mathcal{A}$ into G (at an arbitrary vertex) such that $\theta(B_{\mathcal{A}}) \subseteq X$.

It is obvious that $\alpha_{\mathcal{A}}$ is

1. an opening
2. a flat operator (i.e., it commutes with thresholding: see Section II)
3. symmetry preserving
4. $G-$ increasing

The opening $\alpha_{\mathcal{A}}$ is called a *structural graph opening*. This terminology stems from [21], where it has been shown that under rather mild assumptions (including translation invariance) structural openings form the basis for the collection of all openings. In [13] this result has been extended to the case of graphs.

Proposition 8.1. Let α be a graph opening that is flat, symmetry preserving and $G-$ increasing. Then α can be decomposed as a supremum of structural graph openings.

One can also show that (see [13]) that

$$\delta_{\mathcal{A}}\varepsilon_{\mathcal{A}} \leq \alpha_{\mathcal{A}}$$

An example where this inequality is strict is depicted in Figure 17b.

So far we have seen two ways to construct openings: the first is by composition of an erosion and its adjoint dilation, the second by definition of structural openings. Another powerful tool for building openings is provided by the so-called *inf-overfilters* (again, the corresponding results for closings follow by duality; here one must introduce the concept of a *sup-underfilter*). An increasing operator ψ is called an inf-overfilter if

$$\psi(\mathrm{id} \wedge \psi) = \psi$$

It is obvious that every extensive operator is an inf-overfilter. Furthermore, one can easily show that the class of inf-overfilters is closed under suprema and self-composition. Our interest in inf-overfilters stems from the fact that $\mathrm{id} \wedge \psi$ is an opening if ψ is an inf-overfilter. Now, if (ε, δ) is an adjunction, and if δ' is a dilation such that $\delta' \geq \delta$, then $\delta'\varepsilon$ is an inf-overfilter. Namely,

$$\delta'\varepsilon \geq \delta'\varepsilon(\mathrm{id} \wedge \delta'\varepsilon) = \delta'(\varepsilon \wedge \varepsilon\delta'\varepsilon) \geq \delta'(\varepsilon \wedge \varepsilon\delta\varepsilon) = \delta'\varepsilon$$

The considerations above are valid on arbitrary complete lattices. Here we shall apply them to our graph framework. Let $\mathcal{A}$, $\mathcal{B}$ be two s-graphs such that $\mathcal{A} \leq \mathcal{B}$. Then $\delta_{\mathcal{A}} \leq \delta_{\mathcal{B}}$. Now the abstract theory gives us that $\delta_{\mathcal{B}}\varepsilon_{\mathcal{A}}$ is an inf-overfilter and hence that $\mathrm{id} \wedge \delta_{\mathcal{B}}\varepsilon_{\mathcal{A}}$ is an opening; see Figure 18 for an example.

In classical morphology there exists yet another way to define openings. Take A to be a symmetric structuring element (i.e., $x \in A$ iff $-x \in A$) that does not contain the origin, and define

$$\alpha(X) = (X \oplus A) \cap X$$

One can easily show that α defines an opening. If A is a ring-shaped set (annulus) centered about the origin, then α removes isolated particles; see Figure 5.2 of [24]. For this reason α is called an *annular opening*. We can generalize this notion to graph morphology in the following way. Recall from Section VI that an s-graph is called symmetric if $\check{\mathcal{A}} = \mathcal{A}$

Proposition 8.2. Let $\mathcal{A}$ be a symmetric s-graph; then $\mathrm{id} \wedge \delta_{\mathcal{A}}$ is an opening.

Proof. Though it is not difficult to give a direct proof for gray-level graphs, the demonstration for binary graphs is much more transparent. Since we are dealing exclusively with flat operators, a restriction to binary graphs means no loss of generality. Furthermore, we will suppress the argument G in the notation.

For a binary graph $X \subseteq V$ the dilation can be written as

$$\delta_{\mathcal{A}}(X) = \bigcup_{x \in X} N_{\mathcal{A}}(x)$$

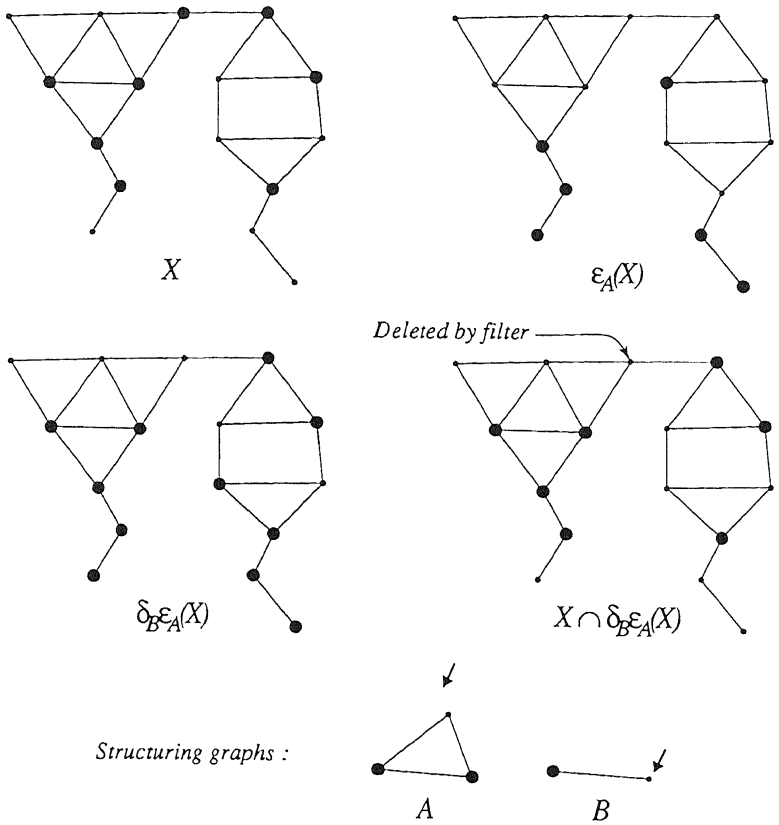


Figure 18. Example of an inf-overfilter.

It is obvious that id $\wedge\ \delta_{\mathcal{A}}$ is antiextensive. To get idempotence it suffices to show that

$$(\mathrm{id} \wedge \delta_{\mathcal{A}})(\mathrm{id} \wedge \delta_{\mathcal{A}}) \geq \mathrm{id} \wedge \delta_{\mathcal{A}},$$

or equivalently that

$$\delta_{\mathcal{A}}(\mathrm{id} \wedge \delta_{\mathcal{A}}) \geq \mathrm{id} \wedge \delta_{\mathcal{A}}$$

Let $X \subseteq V$ and $Y = X \cap \delta_{\mathcal{A}}(X)$. We show that

$$\delta_{\mathcal{A}}(Y) \supseteq Y$$

Let $y \in Y$. Then $y \in \delta_{\mathcal{A}}(X)$ and so $y \in N_{\mathcal{A}}(x)$ for some $x \in X$. Since $\mathcal{A}$ is symmetric we get $x \in N_{\mathcal{A}}(y) \subseteq \delta_{\mathcal{A}}(X)$, since $y \in X$. Therefore, $x \in X \cap \delta_{\mathcal{A}}(X) = Y$, and with $y \in N_{\mathcal{A}}(x)$ this yields $y \in \delta_{\mathcal{A}}(Y)$, which was to be proved.

An example of an annular opening is depicted in Figure 19.

A class of morphological filters which turned out to be quite successful for the cleaning of noisy images are the so-called alternating sequential filters. For a full account of the underlying theory we refer to [24,25]. Here we only sketch the underlying idea. Let $\alpha_1, \alpha_2, \alpha_3, \ldots$ be a sequence of openings, and $\phi_1, \phi_2, \phi_3, \ldots$ a sequence of closings such that

$$\alpha_i\alpha_j = \alpha_j\alpha_i = \alpha_j \quad \text{and} \quad \phi_i\phi_j = \phi_j\phi_i = \phi_j \quad \text{if } j \geq i \tag{6.18}$$

The families α_i and ϕ_i may be chosen independently; often, however, they are taken to be each other's dual. The operators

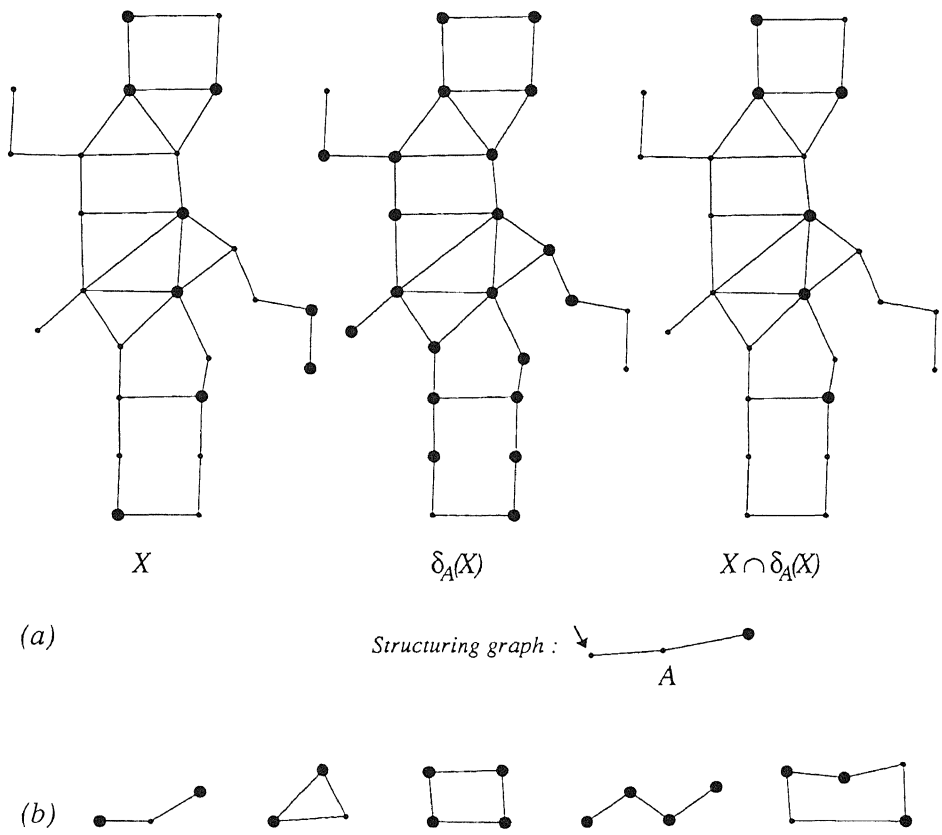


Figure 19. (a) Example of an annular opening. (b) Some typical invariants of the annular opening; note that every invariant is a "union of translates" of the invariant at the left.

$$M_i = \alpha_i\phi_i\alpha_{i-1}\phi_{i-1}\cdots\alpha_1\phi_1, \qquad N_i = \phi_i\alpha_i\phi_{i-1}\alpha_{i-1}\cdots\phi_1\alpha_1$$

are called alternating sequential filters or AS filters. In [25] two other AS filters have been introduced. A systematic way to obtain α_i and ϕ_i is to put

$$\alpha_i = \delta^i\varepsilon^i \quad \text{and} \quad \phi_i = \varepsilon^i\delta^i$$

where (ε, δ) is an adjunction and ε^i denotes the ith power of ε.

The effect of an AS filter is that it successively removes larger and larger noise particles. Furthermore, it treats fore- and background in a more or less similar way. In graph morphology we can construct AS filters by choosing structural openings $\alpha_{\mathcal{A}_i}$ and their dual closings $\phi_{\mathcal{A}_i}$ that satisfy the semigroup property (6.18). At this point it is important to recall the following results from [13]. Let $\mathcal{A}$, $\mathcal{B}$ be s-graphs; we say that $\mathcal{B}$ is *$\mathcal{A}$-open* if for every $v \in B_{\mathcal{B}}$ there is an embedding of $\mathcal{A}$ into $G_{\mathcal{B}}$ at v.

Proposition 8.3. Let $\mathcal{A}$ $\mathcal{B}$ be s-graphs. The equalities

$$\alpha_{\mathcal{A}}\alpha_{\mathcal{B}} = \alpha_{\mathcal{B}}\alpha_{\mathcal{A}} = \alpha_{\mathcal{B}}$$

hold if and only if $\mathcal{B}$ is $\mathcal{A}$-open.

Now take a sequence of s-graphs $\mathcal{A}_i$ such that $\mathcal{A}_{i+1}$ is $\mathcal{A}_i$-open. See Figure 20 for a number of such sequences. Then $\mathcal{A}_j$ is $\mathcal{A}_i$-open if $j \geq i$. Let $\alpha_i = \alpha_{\mathcal{A}_i}$ and $\phi_i = \phi_{\mathcal{A}_i}$ be the corresponding structural openings (respectively, closings). Then the operators M_i and N_i defined above are AS filters.

Proposition 8.3 also lays the foundation for the definition of granulometries that in turn yield size distributions. We can think of a granulometry as a collection of openings α_i $(i \geq 1)$ that satisfies the first condition in Eq. (6.18).

IX. CONCLUDING REMARKS

Before we conclude this chapter, let us give a few hints on the implementation of morphological transformations on graphs. We already dealt briefly with the obtention of neighborhood graphs from binary images (see Section V). Transforming such objects morphologically first involves encoding them in an appropriate way. This is achieved via a data structure derived from the adjacency matrix of the graph [2] and is detailed in [28,30]. Given a vertex, this structure allows direct access to its neighbors.

For the algorithms themselves, two cases have to be considered: structuring and nonstructuring graphs. The first case is by far the more difficult one; what one has to do basically is to find all the different ways of matching the structuring graphs within the graph to be transformed. This is achieved by scanning the search tree as efficiently as possible. The implementation of nonstructured graph operators is much easier. Indeed, as mentioned in Section IV, all the nonstruc-

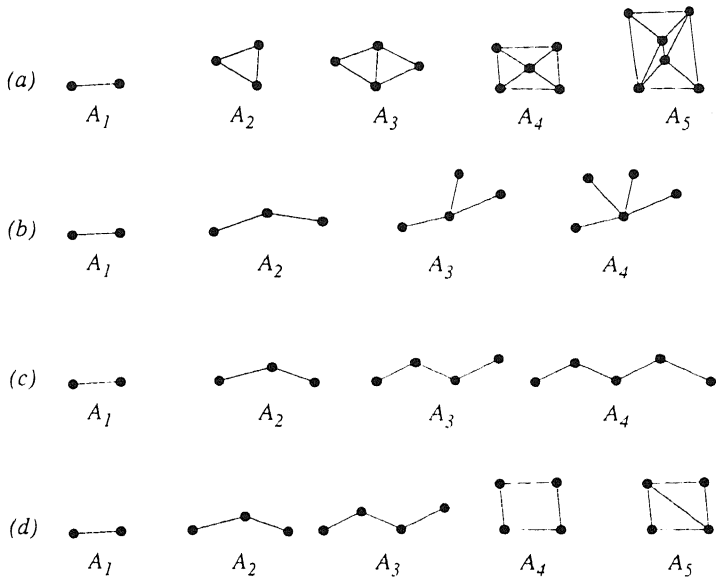


Figure 20. Families of s-graphs to be used as AS filters and granulometries on graphs. Note that A_{i+1} does not necessarily have more buds than A_i.

tured graph morphology operations are based on the *distance* induced by the set of edges E on the vertex set V. Therefore, *breadth-first scannings* will be at the basis of most algorithms. They are implemented via the use of a *queue* of vertices, that is, FIFO structure (see Figure 21). This is explained in further detail in [28,32,34].

For example, to determine the distance function of a graph—that is, to assign to a vertex its distance to a particular set W of vertices—one starts from the vertices of W and does a breadth-first scanning. In this procedure, the vertices at distance 1 are first met, then those at distance 2, etc., until stability is reached. This algorithm—as well as many others described in [32]—is particularly efficient because each vertex is considered a minimal number of times. It was used for the first example presented in Section V. Another example of a distance function, on a Delaunay triangulation this time, is presented in Figure 22.

Graph morphology provides a collection of morphological tools for the investigation of populations of objects for which neighborhood relations are of interest. Here objects may be physical or biological objects, such as the nuclei in a microscopic image of some cell tissue, but they may also refer to a symbolic description of a scene. As to the latter case, one may think of the situation in which the objects represent the intensity extrema of a scene (see [18]). One can

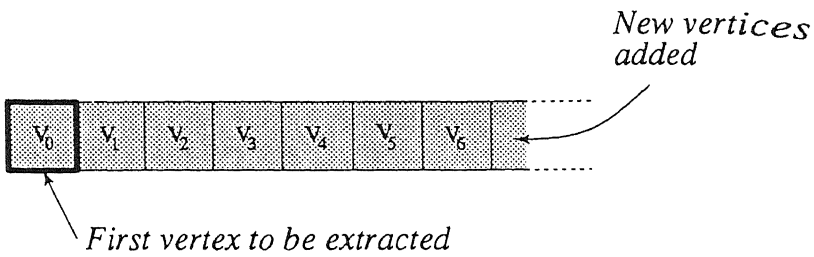


Figure 21. How a queue (FIFO structure) of vertices works.

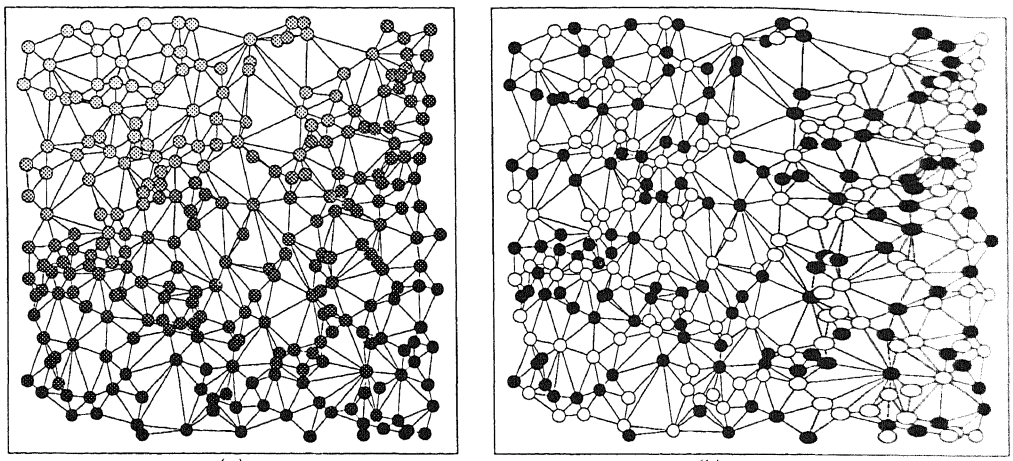


Figure 22. Example of a distance function on the Delaunay triangulation of Figure 8a. In (a), darker vertex values represent larger distances; in (b), the distance function is displayed modulo 2.

model spatial relationships between objects by different types of graphs. In this chapter we have discussed three such graphs, namely the Delaunay triangulation, the Gabriel graph, and the relative neighborhood graph.

Although the intrinsic structure of a gray-level graph is much poorer than that of the Euclidean space $\mathbf{R}^d$ or the discrete space $\mathbf{Z}^d$, it is still rich enough to define a large class of morphological operators for such images that inherit many of the nice properties of their classical analogues. Thereto we have introduced the concept of a structuring graph that generalizes the structuring element in classical morphology. If the structuring graph consists of only two vertices connected by an edge, then the resulting graph operators are of the so-called nonstructured

type. They are easy to implement but still allow the construction of a large collection of morphological algorithms. Structured graph operations, which use more information about the local graph structure near a vertex, allow a much larger collection of morphological operations (see, e.g., the sequence of structuring graphs depicted in Figure 20, which can be used to define granulometries and AS filters), but they are definitely harder to implement. As often in science, the future will tell us to which extent these structured and nonstructured graph operations turn out to be useful in practical applications.

Acknowledgments

The work of L. Vincent was supported in part by Dassault Electronique and the National Science Foundation under grant MIPS-86-58150, with matching funds from DEC and Xerox.

REFERENCES

1. van Antwerpen, G., and van Munster, R. J., *TCL-IMAGE User's Manual*, Institute of Applied Physics, TNO, Delft, The Netherlands, 1989.
2. Berge, C., *Graphs*, 2nd rev. ed., North-Holland, Amsterdam, 1985.
3. Beucher, S., and Lantuéjoul, Ch., Use of watersheds in contour detection, in *Proceedings of the International Workshop on Image Processing, Real-Time Edge and Motion Detection/Estimation*, Rennes, France, 1979.
4. Beucher, S., Segmentation d'images et morphologie mathématique, Ph.D. dissertation, Ecole Nationale Supérieure des Mines de Paris, France, 1990.
5. Birkhoff, G., *Lattice Theory*, 3rd ed., Colloquium Publications, vol. 25, American Mathematical Society, Providence, Rhode Island, 1984.
6. Boissonat, J. D., Shape reconstructions from planar cross-sections, Technical Report INRIA, Le Chesnay, France, 1986.
7. Borgefors, G., Distance transformations in digital images, *Comput. Vision Graphics Image Process.*, *34*, 334–371 (1986).
8. Gabriel, K. R., and Sokal, R. R., A new statistical approach to geographic variations analysis, *Syst. Zool.*, *18*, 259–278 (1969).
9. Gondran, M., and Minoux, M., *Graphs and Algorithms*, Wiley, Chichester, 1984.
10. Heijmans, H. J. A. M., Theoretical aspects of gray-level morphology, *IEEE Trans. Pattern Anal. Machine Intell.*, *13*, 568–582 (1991).
11. Heijmans, H. J. A. M., and Ronse, C., The algebraic basis of mathematical morphology. Part I. Dilations and erosions, *Comput. Vision Graphics Image Process.*, *50*, 245–295 (1990).
12. Heijmans, H. J. A. M., and Serra, J., Convergence, continuity and iteration in mathematical morphology, *J. Visual Commun. Image Represent.*, in press.
13. Heijmans, H. J. A. M., Nacken, P., Toet, A., and Vincent, L., Graph morphology, *J. Visual Commun. Image Representation*, in press.
14. Jeulin, D., Vincent, L., and Serpe, G., Propagation algorithms on graphs for phys-

ical applications, Internal report CMM, Ecole Nationale Supérieure des Mines de Paris, France, 1990.
15. Maragos, P., and Schafer, R. W., Morphological filters. Part I. Their set-theoretic analysis and relations to linear shift-invariant filters, *IEEE Trans. Acoust. Speech Signal Process.*, *35*, 1153–1169 (1987).
16. Matheron, G., Filters and lattices, Internal report CGMM, Ecole Nationale Supérieure des Mines de Paris, France, 1989.
17. Matheron, G., *Random Sets and Integral Geometry*, Wiley, New York, 1975.
18. Nacken, P., Hierarchical image structure description based on intensity extrema, in *Geometrical Problems of Image Processing*, Georgenthal (DDR) (U. Eckhardt, A. Hübler, W. Nagel, and G. Werner, eds.), Akademie Verlag, Berlin, 1991, pp. 99–106.
19. O, Y.-L., and Toet, A., Mathematical morphology in hierarchical image representation, in *Proceedings, NATO ASI on the Formation, Handling and Evaluation of Medical Images*, A. E. Todd-Pokropek and M. A. Viergevere (eds), NATO ASI Series F, Springer, Heidelberg, 447–462, 1991.
20. Preparata, F. P., and Shamos, M. I., *Computational Geometry: An Introduction*, Springer-Verlag, New York, 1985.
21. Ronse, C., and Heijmans, H. J. A. M., The algebraic basis of mathematical morphology. Part II. Openings and closings, *Comput. Vision Graphics Image Process. Image Understand.*, *54*, 384–400, 1991.
22. Raphaël, M., Vincent, L., Raymond, E., Grimaud, M., and Meyer, F., Germinal center (GC) analysis with graphs and mathematical morphology, in *Proceedings, Congress of the International Society for Analytical Cytology*, Bergen, Norway, in press.
23. Serra, J., *Image Analysis and Mathematical Morphology*, Academic Press, London, 1982.
24. Serra, J., (ed.), *Image Analysis and Mathematical Morphology*, Part II, *Theoretical Advances*, Academic Press, London, 1988.
25. Serra, J., and Vincent, L., An overview of morphological filtering, *IEEE Trans. Circuits Syst. Signal Process.*, *11*, 47–100, 1992.
26. Supowit, K. J., The relative neighbourhood graph with an application to minimum spanning trees, *J. ACM*, *30*(3), 428–447 (1983).
27. Toussaint, G. T., The relative neighborhood graph of a finite planar set, *Pattern Recogn.*, *12*, 1324–1347 (1980).
28. Vincent, L., Mathematical morphology on graphs, in *Proceedings SPIE, Visual Communications and Image Processing 88*, Cambridge, Massachusetts, 1988, pp. 95–105.
29. Vincent, L., Mathematical morphology for graphs applied to image description and segmentation, in *Proceedings Electronic Imaging West 89*, Pasadena, 1989, vol. 1, pp. 313–318.
30. Vincent, L., Graphs and mathematical morphology, *Signal Process.*, *16*, 365–388 (1989).
31. Vincent, L., and Jeulin, D., Minimal paths and crack propagation simulations, in *Acta Stereologica 8, Proceedings 5th European Congress for Stereology*, 1989, pp. 487–494.

32. Vincent, L., Algorithmes morphologiques à base de files d'attente et de lacets. Extension aux graphes, Ph.D. dissertation, Ecole Nationale Supérieure des Mines de Paris, France, 1990.
33. Vincent, L., Exact Euclidean distance function by chain propagations, Technical Report 91-4, Harvard Robotics Laboratory, Cambridge, February 1991; *Proc. IEEE Comput. Vision Pattern Recogn.*, in press.
34. Vincent, L., Morphological algorithms, in *Mathematical Morphology in Image Processing* (E. Dougherty, ed.), Marcel Dekker, New York, 1991.
35. Wendt, P. D., Coyle, E. J., and Gallagher, N. C., Stack filters, *IEEE Trans. Acoust. Speech Signal Process.*, *34*, 898–911 (1986).