Morphological segmentation produces a Voronoï tesselation of the markers

Fernand Meyer (email : meyer@cmm.ensmp.fr)
Centre de Morphologie Mathématique
Ecole des Mines de Paris
77305 Fontainebleau / France

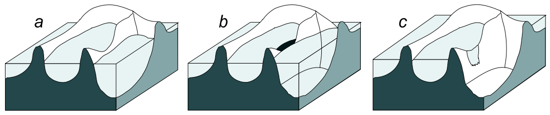

Figure 1: a) flooding in progress, starting from all minima
b) construction of a dam for separating the lakes within distinct catchment basins
c) different scenario : a catchment basin without source is flooded through a neighboring basin

Abstract

The construction of a minimum spanning forest on a graph, where the trees are rooted in a predefined set of nodes, is shown to be equivalent to constructing the Voronoï tesselation of the nodes for a lexicographic distance function. The domain of application here is morphological segmentation with markers.

1 Introduction

For segmenting an image f, first its edges are enhanced by computing its gradient magnitude $||\nabla f||$. After edge enhancement, the segmentation process starts with creating flooding waves that emanate from a set of markers (feature points inside desired regions) and flood the topographic surface $||\nabla f||$. The simplest markers are the regional minima of the gradient image. The points where these flooding waves meet each other form the segmentation boundaries (see fig.1a and b). Very often, the minima are extremely numerous, leading to an oversegmentation. For this reason, in many practical cases, the watershed will take as sources of the flooding a smaller set of markers, which have been identified by a preliminary analysis step as inside germs of the desired segmentation [?, ?]. In such cases, catchment basins without sources are flooded by the overflood of neighboring basins, as shown in fig.1c. Such a flooding algorithm, using hierarchical queues has been described in [?]. A competition between the various floodings occurs and the contour lines which are selected are highest portions of watershed lines one meets when traveling from one marker to another. The result is a tessellation or partition of the space, where each tile possesses one seed and is the union of all catchment basins from the original relief flooded by this particular seed. We will show that the resulting segmentation can be interpreted as the Voronoï tesselation of the markers, provided an appropriate lexicographic distance is chosen. Strategies for defining markers are diverse and problem dependent [?]. The algorithm belongs to the large family of region growing algorithms [?].

2 On trees and forests

2.1 The region adjacency graph

Any partition $\mathcal{A}$ for which a dissimilarity between adjacent regions has been defined can be represented as a region adjacency graph $G = (X, E)$, where X is the set of nodes and E is the set of edges. The nodes represent regions of the partition. Adjacent regions i and j are linked by an edge $u = (i, j)$ with a weight s_{ij} expressing the dissimilarity between them. In case of a topographic surface the partition is the set of its catchment basins, the dissimilarity between two adjacent basins being the altitude of the pass separating them. A path $\mu = (i_1, i_2, ...i_k)$ is a sequence of neighboring nodes. The adjacency matrix $A = (\alpha_{ij})$ of the graph is defined by:

$$\alpha_{ij} = \left\{ \begin{array}{ccc} s_{ij} & \text{if} & (i,j) \in E \\ & \infty & \text{if not} \end{array} \right\}$$

For any $\lambda \geq 0$, one defines a derived graph $G_\lambda = [X, E_\lambda]$ with the same nodes but only a subset of edges : $E_\alpha = \{(i,j) \mid \alpha_{ij} \leq \lambda\}$. The connected components of this graph create a partition of the objects X into classes. They constitute the classes of the classification at level λ. If $L = (i, i_2, ...i_p, j)$ is this path, the maximal dissimilarity along L verifies $\max(\alpha_{ii_1}, \alpha_{i_1 i_2}, ...\alpha_{i_p j}) \leq \lambda$. Two nodes belong to the same class at level λ if and only if there exists a path in G linking these two nodes along which all dissimilarity indices are below λ : $\alpha^*_{ij} = \min_{L \in C_{ij}} (\max(\alpha_{ii_1}, \alpha_{i_1 i_2}, ...\alpha_{i_p j})) \leq \lambda$ where C_{ij} is the set of all paths between i and j. Going back to morphological flooding : two catchment basins belong to the same class at level λ if and only if the corresponding minima belong to the same lake at flooding level λ. It is easy to verify that α^*_{ij} is an ultrametric distance, called max-distance, verifying for any i, j, k : $\alpha^*_{ik} \leq \max\left(\alpha^*_{ij}, \alpha^*_{jk}\right)$. The closed balls $\text{Ball}(i, \rho) = \left\{j \in X \mid \alpha^*_{ij} \leq \rho\right\}$ form the classes of the partition at level ρ. For increasing levels of ρ one obtains coarser classes.

2.2 Lexicographic distances and paths algebra

Let us consider the tree of fig.2, where the nodes a, b, c and d are markers. If a flood starts from the markers, which flood will attain the node x first ? In our case it is obviously the flood coming from the source b ; hence x will be assigned to the marker b. Is there a distance function for which the distance between x and b is shorter than the distance from x to any othe marker ? The max-distances from x to the markers are ranked as follow : $(\alpha^*_{xd} = 7) > (\alpha^*_{xa} = \alpha^*_{xb} = \alpha^*_{xc} = 6)$. From the first inequality we can discard the flood from d, as it has an edge of altitude 7 to pass, whereas the highest edge for the other floods is only 6. The ultrametric distance is myopic and is unable to discriminate between the markers a, b and c. The floods from a, b and c have all to pass through the edge (y, x) of altitude 6 ; the first reaching the node y will be the first crossing the edge (y, x), hence the first reaching x. Applying the same reasoning to the node y, we can discard a as $\left(\alpha^*_{ya} = 5\right) > \left(\alpha^*_{yb} = \alpha^*_{yc} = 4\right)$. Among the remaining candidates b and c, the flood coming from b reaches

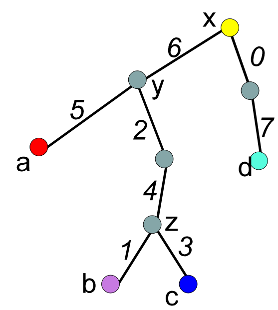

Figure 2: Following the path from x towards the sources, one sees that :
$(\alpha^*_{xd} = 7) > (\alpha^*_{xa} = \alpha^*_{xb} = \alpha^*_{xc} = 6)$
$\left(\alpha^*_{ya} = 5\right) > \left(\alpha^*_{yb} = \alpha^*_{yc} = 4\right)$
$(\alpha^*_{zb} = 1) < (\alpha^*_{zc} = 3)$

z and crosses the edge of altitude 1 before the flood coming from c, as $(\alpha^*_{zb} = 1) < (\alpha^*_{zc} = 3)$. Finally the flood coming from b ultimately wins because it arrives first all along the path between b and x. The distance which is minimal along the winning path of flooding is a lexicographic distance. The lexicographic length of a path between a node x and a marker m_1 is defined as a sequence $(\lambda_1, \lambda_2, ..., \lambda_k)$ of weights : the largest weight λ_1 on the path between x and m_1, then the largest weight λ_2 on the remaining part of the path and so on until one reaches the marker m_1. The sequence of values obtained is never increasing $\lambda_1 \geq \lambda_2 \geq .... \geq \lambda_k$. In our case we see that the lexicographic distance correctly ranks the floodings reaching x.

$\lambda(x, d) = 7 > \lambda(x, a) = (6, 5) > \lambda(x, c) = (6, 4, 3) > \lambda(x, b) = (6, 4, 1)$

The lexicographic length of the path $A = a_1 a_2 \ldots a_k \ldots a_n$ may be formally defined by $\Lambda(a_1 a_2 \ldots a_k \ldots a_n) = val(a_{k-1} a_k) \triangleright \Lambda(a_k \ldots a_n)$

where $a_{k-1} a_k$ is the largest edge of path A and $\triangleright$ represents the concatenation operator. Note that the paths are directed and that $\Lambda(a_1 a_2 \ldots a_k \ldots a_n) \neq \Lambda(a_n a_{n-1} \ldots a_k \ldots a_1)$. We have defined the lexicographic length of a path $\Lambda(a_1 a_2 \ldots a_k \ldots a_n)$. The lexicographic distance between a node x and a node y will be defined as the lexicographic length of the shortest path from x to y, which will be noted $\widehat{\Lambda}(x, y)$. If x is lexicographically closer to m_k and if $(x \rightarrow m_k)$ is the corresponding shortest path, we have seen that the flooding starting from m_k reaches all the nodes of this path before any other flooding. All the nodes of this path thus belong to the influence zone of m_k.

Therefore the influence zone is connected. It is possible to show that the segmentation associated to a set of markers can now be defined as the Voronoi partition generated by this set of markers using the lexicographic distance. The next section shows how to efficiently construct the matrix of shortest distances for this new lexicographic distance.

3 Dioids and Algorithms

3.1 A dioid of lexicographic distances

Shortest-path algorithms on graphs can be expressed as linear algebra operations on dioids [**?**].

Let S be the set of all lexicographic distances, that is decreasing n-tuples $(\lambda_1, \lambda_2, \ldots, \lambda_k)$ with $k \leq N-1$, $\lambda_i \in \mathcal{R}^+$ such that $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_k$. We define on S the usual lexicographic order relation, which we will note $\prec$, such that: $(\lambda_1, \lambda_2, \ldots, \lambda_k) \prec (\mu_1, \mu_2, \ldots, \mu_k)$ if either $\lambda_1 < \mu_1$ or $\lambda_i = \mu_i$ until rank s where $\lambda_{s+1} < \mu_{s+1}$. We define $a \preceq b$ as $a \prec b$ or $a = b$.

We now define two operator, $\boxplus$ and $*$ on S:

- an operator $\boxplus$, which will be called "addition" associates to two lexicographic lengths the shortest:

  $$a \boxplus b = \begin{array}{l} a \text{ if } a \preceq b \\ b \text{ if } b \preceq a \end{array} \qquad \forall a, b \in S.$$

  The $\boxplus$ operation, called "addition"is associative, commutative and has a neutral element ∞ called the zero element: $a \boxplus \infty = a$

- an operator $*$, which will be called "multiplication" associates to the lexicographic lengths of two paths A and B the lexicographic length of the concatenation of boths. Hence if $A = a_1 a_2 \ldots a_k \ldots a_n$ is a path made of two subpaths $B = a_1 a_2 \ldots a_k$ and $C = a_k a_{k+1} \ldots a_n$, then $\Lambda(A) = \Lambda(B) * \Lambda(C)$. Let $a = (\lambda_1, \lambda_2, \ldots, \lambda_k)$ and $b = (\mu_1, \mu_2, \ldots, \mu_l)$; we will define $a * b$ as:

  - if $\mu_1 > \lambda_1$ then $a * b = b$
  - if $\lambda_k \geq \mu_1$ then $a * b = a \rhd b$ where $\rhd$ is the concatenation
  - else let j be the smallest index verifying $\lambda_{j-1} \geq \mu_1 \geq \lambda_j$ then $a * b$=$(\lambda_1, \lambda_2, \ldots, \lambda_j, \mu_1, \mu_2, \ldots, \mu_l)$

  This algorithm guarantees that $a * \infty = \infty$. We also define $\infty * a = \infty$, so that the zero element is an absorbing element for $*$. The operator $*$ is associative, has a neutral element 0 called the unit element: $a * 0 = a$

The multiplication is distributive with respect to the addition both to the left and to the right, and the null element is an absorbing element for the multiplication : $a * \infty = \infty$

3.2 Path algebra

Hence the set S of lexicographic distances with the $\boxplus$ and $*$ operators is a dioïd. It follows that the set $\mathcal{M}_N(S)$ of the matrices of lexicographic distances is also a dioid for the $\boxplus$ and $*$ operators, which are defined as:

- $C = A \boxplus B = (c_{ij}) \Leftrightarrow c_{ij} = a_{ij} \boxplus b_{ij} \qquad \forall i, j$
- $C = A * B = (c_{ij}) \Leftrightarrow$

  $c_{ij} = \sum\limits_{1 \leq k \leq n}^{\boxplus} a_{ik} * b_{kj} \qquad \forall i, j$

A large number of shortest paths problems are solved by computing A^k or $A^{(k)} = E \boxplus A^1 \boxplus A^2 \boxplus ....A^k$. Obviously A^k_{ij} is the length of the shortest path with $k+1$ nodes between i and j and $A^{(k)}_{ij}$the length of the shortest path with at most $k+1$ nodes between i and j

3.2.1 Algebraic solutions of a linear system

A^* verifies $A^* = E \boxplus A * A^*$.

Multiplying by a matrix B yields $A^*B = B \boxplus A * A^*B$. Defining $Y = A^*B$, one sees that A^*B is solution of the equation $Y = B \boxplus A * Y$. Moreover it is the smallest solution.

With a right choice of B, various problems may be solved:

- For finding A^* one may solve $Y = E \boxplus A * Y$ yielding $A^*E = A^*$ as solution
- with $B = \begin{bmatrix} \infty \\ \vdots \\ 0 \\ \vdots \\ \infty \end{bmatrix}$, one obtains A^*B, the i-th column of the matrix A^*, that is the distance of all nodes to node i.

For finding the solution of such linear systems, most linear algebra algorithms are still valid :

- The Jacobi algorithm : one iterates $Y^{(k)} = A * Y^{(k-1)} \boxplus B$ until convergence $Y^{(N)} = A^* * B$
- The Gauss-Seidel algorithm : one decomposes the matrix as the sum of a lower triangular matrix L, and an upper triangular matrix U: $A = L \boxplus U$. The solution of $Y = A * Y \boxplus B$ is obtained by constructing : $Y^{(k)} = LY^{(k-1)} \boxplus UY^{(k)} \boxplus B$. It converges faster, as in the product $UY^{(k)}$ one may already use the freshly computed values of $Y^{(k)}$.
- The Jordan algorithm permits to invert a matrix by successive pivoting :
  For k from 1 to N
  For all i and j from 1 to N : $a_{ij} = a_{ij} \boxplus a_{ik} * a_{kj}$
  Illustration :

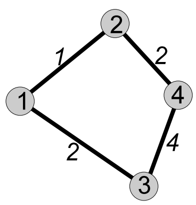

Figure 3: A weighted graph

Consider the weighted graph of fig.3. For its incidence matrix is $A = \begin{bmatrix} 0 & 1 & 2 & \infty \\ 1 & 0 & \infty & 2 \\ 2 & \infty & 0 & 4 \\ \infty & 2 & 4 & 0 \end{bmatrix}$ the algorithm goes as follows :
k =1 :
$a_{23} = a_{23} \boxplus a_{21} * a_{13} = \infty \boxplus 1 * 2 = 2$
$a_{32} = a_{32} \boxplus a_{31} * a_{12} = \infty \boxplus 2 * 1 = 2, 1$
k=2 : $a_{14} = 2$; $a_{41} = 2, 1$; $a_{13} = 2$; $a_{31} = 2$; $a_{34} = 2, 2$; $a_{43} = 2, 2$
At the end of this step, the correct result is already obtained and the next step does not change it. If now, for instance the nodes 1 and 2 are markers, one has to compare the columns 1 and 2 of the matrix A^*, and affect each node to the column where its value is smallest. Here the nodes 1 and 3 are affected to the marker 1 and the nodes 2 and 4 to the marker 2.

- The greedy algorithm (Dijkstra)

If $\overline{Y} = A^* B$ is the solution of $Y = AY \boxplus B$, for a column vector B then there exists and index i_0 such that $\overline{y_{i_0}} = \sum^{\boxplus} b_i$.Hence the smallest b is solution : $\overline{y_{i_0}} = b_{i_0}$. Each element of $Y = AY \boxplus B$ is obtained by $y_k = \sum_{j \neq k}^{\boxplus} a_{kj} * y_j \boxplus b_k = \sum_{j \neq k, i_0}^{\boxplus} a_{kj} * y_j \boxplus a_{ki_0} y_{i_0} \boxplus b_k$. Suppressing the line and the column of rank i_0 and taking for B the vector $b_k^{(1)} = a_{ki_0} y_{i_0} \boxplus b_k$, one obtains a new system of size $N - 1$ to solve.

4 Conclusion

We have shown that the popular segmentation method with markers is equivalent to constructing the skeleton of influence of the marker for a particular lexicographic distance. The path algebra introduced by Gondran and Minoux allows to use many existing and well-studied algorithms to process the segmentation. One may then chose the algorithm which best adapted to the particular situation. While not being as fast as a straightforward segmentation on the first run, the interest appears when multiple segmentations are performed on the same image with only the marker set being modified, as can be expected in an interactive segmentation process. The interest is again increased for images of floating point numbers because the priority queues used to implement the watershed can't take advantage of a fixed predefined number of classes.

Aknowledgements: We thank Romain Lerallut for having programmed and tested the algorithms presented here.

References