Segmentation by discrete watersheds
Part 2: Algorithms and Seeded watershed cuts

Jean Cousty

FOUR-DAY COURSE
on
Mathematical Morphology in image analysis
Bangalore 19-22 October 2010

Thinnings: a new paradigm to compute watershed cuts

Thinnings: a new paradigm to compute watershed cuts

- Intuitively, a thinning consists of iteratively lowering the values of the edges which satisfy a certain property

Thinnings: watershed algorithm

Thinnings: a new paradigm to compute watershed cuts

- Intuitively, a thinning consists of iteratively lowering the values of the edges which satisfy a certain property
  - $\mathscr{B}$-thinnings
  - $\mathscr{M}$-thinnings
  - $\mathscr{I}$-thinnings

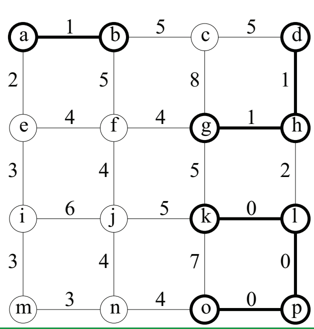
Local edge classification

- The *altitude* of a vertex x, denoted $F^{\ominus}(x)$, is the minimal altitude of an edge which contains x:
  - $F^{\ominus}(x) = \min\{F(u) \mid u \in E \text{ and } x \in u\}$

Thinnings: watershed algorithm

Local edge classification

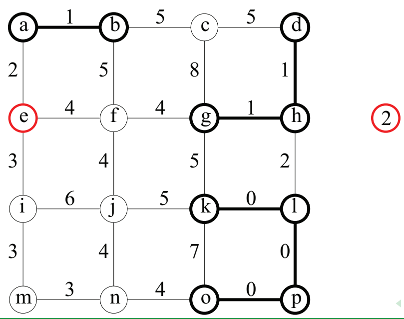
The *altitude* of a vertex x, denoted $F^{\ominus}(x)$, is the minimal altitude of an edge which contains x:
$F^{\ominus}(x) = \min\{F(u) \mid u \in E \text{ and } x \in u\}$



J. Serra, J. Cousty, B.S. Daya Sagar : Course on Math. Morphology 4/34

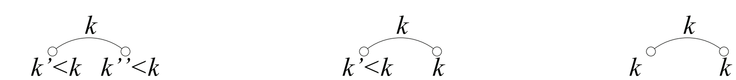
Local edge classification

- The *altitude* of a vertex x, denoted $F^{\ominus}(x)$, is the minimal altitude of an edge which contains x:
  - $F^{\ominus}(x) = \min\{F(u) \mid u \in E \text{ and } x \in u\}$

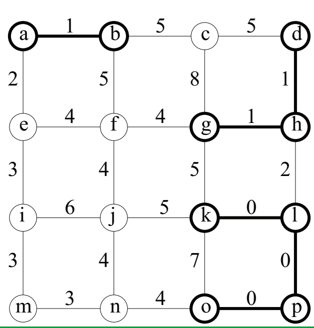


Thinnings: watershed algorithm

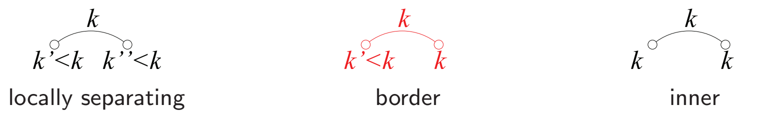
Local edge classification



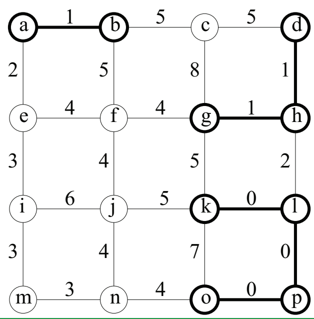
locally separating — border — inner



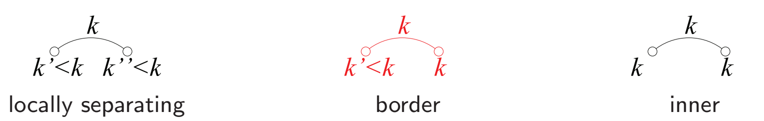
Local edge classification



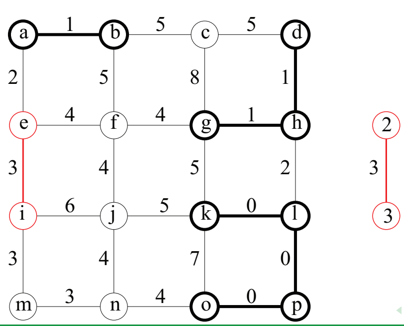
- We say that $u = \{x, y\}$ is a *border edge for (for F)* if:
  - $F(u) = \max(F(x), F(y))$; and $F(u) > \min(F(x), F(y))$



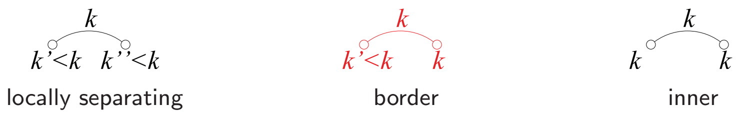
Local edge classification



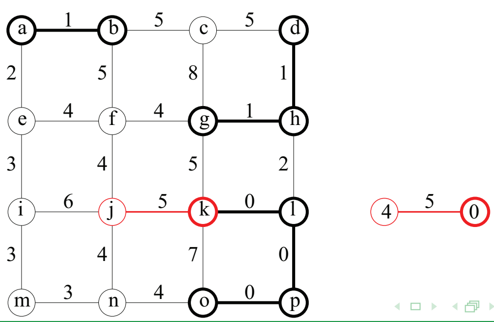
- We say that $u = \{x, y\}$ is a *border edge for (for F)* if:
  - $F(u) = \max(F(x), F(y))$; and $F(u) > \min(F(x), F(y))$



Local edge classification



- We say that $u = \{x, y\}$ is a *border edge for (for F)* if:
  - $F(u) = \max(F(x), F(y))$; and $F(u) > \min(F(x), F(y))$



$\mathscr{B}$-thinnings & $\mathscr{B}$-cuts

The *lowering of F at u* is the map F' such that:

- $F'(u) = \min_{x \in u}\{F^{\ominus}(x)\}$; and
- $F'(v) = F(v)$ for any edge $v \in E \setminus \{u\}$.

$\mathscr{B}$-thinnings & $\mathscr{B}$-cuts

Definition

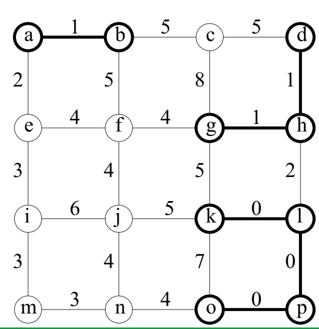
- *A map H is a $\mathscr{B}$-thinning of F if H may be derived from F by iterative lowerings at border edges*

Thinnings: watershed algorithm

$\mathscr{B}$-thinnings & $\mathscr{B}$-cuts

Definition

- *A map H is a $\mathscr{B}$-thinning of F if H may be derived from F by iterative lowerings at border edges*

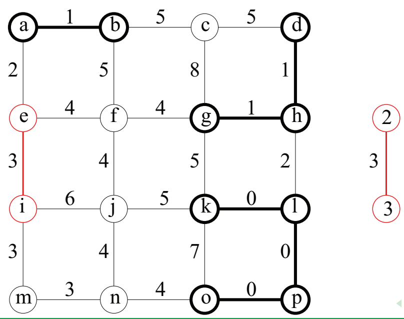


J. Serra, J. Cousty, B.S. Daya Sagar : Course on Math. Morphology 5/34

$\mathscr{B}$-thinnings & $\mathscr{B}$-cuts

Definition

- *A map H is a $\mathscr{B}$-thinning of F if H may be derived from F by iterative lowerings at border edges*

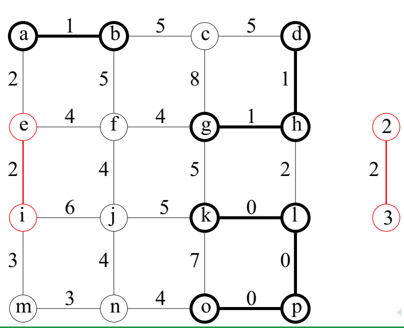


Thinnings: watershed algorithm

$\mathscr{B}$-thinnings & $\mathscr{B}$-cuts

Definition

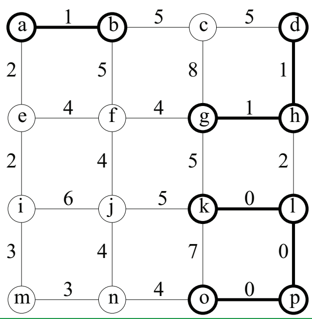
- *A map H is a $\mathscr{B}$-thinning of F if H may be derived from F by iterative lowerings at border edges*



$\mathscr{B}$-thinnings & $\mathscr{B}$-cuts

Definition

- *A map H is a* $\mathscr{B}$-thinning of F *if H may be derived from F by iterative lowerings at border edges*

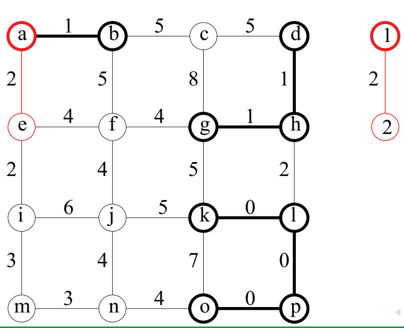


Thinnings: watershed algorithm

$\mathscr{B}$-thinnings & $\mathscr{B}$-cuts

Definition

- *A map H is a* $\mathscr{B}$-thinning of F *if H may be derived from F by iterative lowerings at border edges*

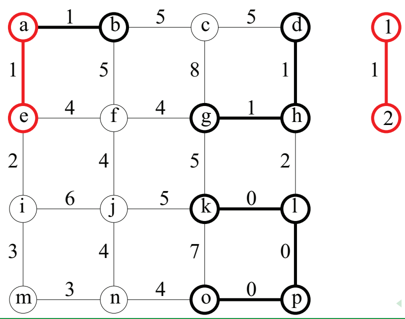


J. Serra, J. Cousty, B.S. Daya Sagar : Course on Math. Morphology 5/34

$\mathscr{B}$-thinnings & $\mathscr{B}$-cuts

Definition

- *A map H is a* $\mathscr{B}$-thinning of F *if H may be derived from F by iterative lowerings at border edges*

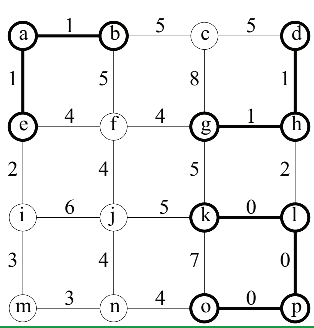


Thinnings: watershed algorithm

$\mathscr{B}$-thinnings & $\mathscr{B}$-cuts

Definition

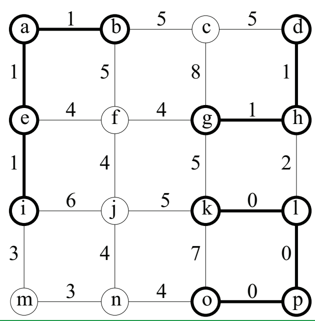
- *A map H is a* $\mathscr{B}$-thinning of F *if H may be derived from F by iterative lowerings at border edges*



$\mathscr{B}$-thinnings & $\mathscr{B}$-cuts

Definition

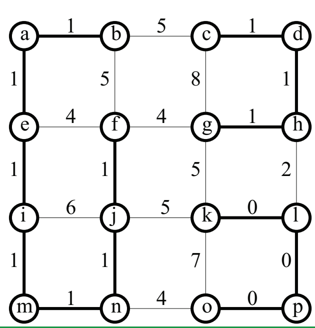
- *A map H is a* $\mathscr{B}$-thinning of F *if H may be derived from F by iterative lowerings at border edges*



$\mathscr{B}$-thinnings & $\mathscr{B}$-cuts

Definition

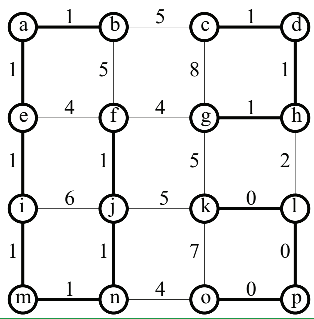
- *A map H is a* $\mathscr{B}$-thinning of F *if H may be derived from F by iterative lowerings at border edges*



$\mathscr{B}$-thinnings & $\mathscr{B}$-cuts

Definition

- *A map H is a $\mathscr{B}$-thinning of F if H may be derived from F by iterative lowerings at border edges*
- *The map H is a $\mathscr{B}$-kernel of F if H is a $\mathscr{B}$-thinning of F and if there is no border edge for H.*

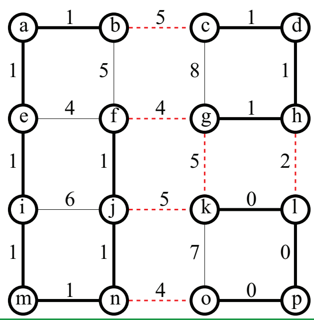


Thinnings: watershed algorithm

$\mathscr{B}$-thinnings & $\mathscr{B}$-cuts

Definition

- *We say that $S \subseteq E$ is a $\mathscr{B}$-cut of F if there exists a $\mathscr{B}$-kernel H of F such that S is the set of all edges linking two distinct minima of H.*



$\mathscr{B}$-kernels, $\mathscr{B}$-cuts & watersheds

Theorem

- *A graph X is an MSF relative to the minima of F* if and only if *X is the graph of the minima of a $\mathscr{B}$-kernel of F*
- *An edge set $S \subseteq E$ is a $\mathscr{B}$-cut of F* if and only if *S is a watershed of F*

Border thinning algorithm ?

- $\mathscr{B}$-thinnings:
  - Rely on a local condition

Border thinning algorithm ?

- $\mathscr{B}$-thinnings:
  - Rely on a local condition
  - Adapted to parallel strategies

Border thinning algorithm ?

- $\mathscr{B}$-thinnings:
  - Rely on a local condition
  - Adapted to parallel strategies

Problem

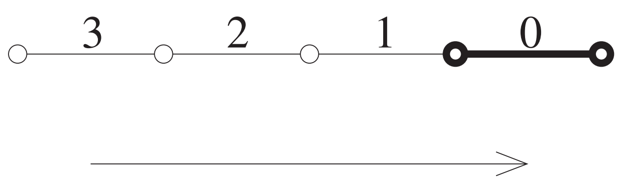
- *But a same edge can be lowered several times*

Border thinning algorithm ?

- $\mathscr{B}$-thinnings:
  - Rely on a local condition
  - Adapted to parallel strategies

Problem

- *But a same edge can be lowered several times*

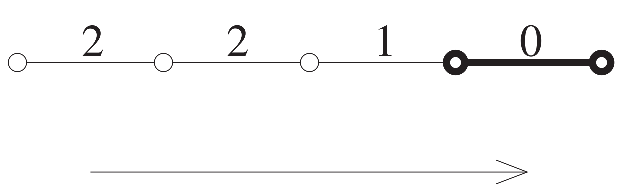


Border thinning algorithm ?

- $\mathscr{B}$-thinnings:
  - Rely on a local condition
  - Adapted to parallel strategies

Problem

- *But a same edge can be lowered several times*

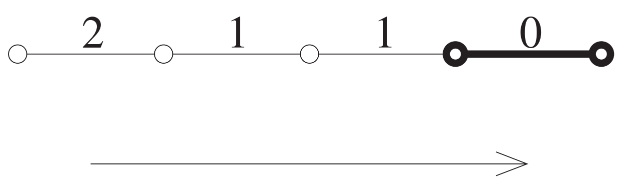


Border thinning algorithm ?

- $\mathscr{B}$-thinnings:
  - Rely on a local condition
  - Adapted to parallel strategies

Problem

- *But a same edge can be lowered several times*



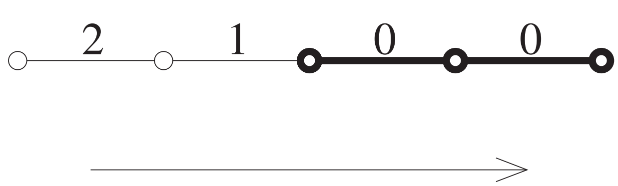
Thinnings: watershed algorithm

Border thinning algorithm ?

- $\mathscr{B}$-thinnings:
  - Rely on a local condition
  - Adapted to parallel strategies

Problem

- *But a same edge can be lowered several times*

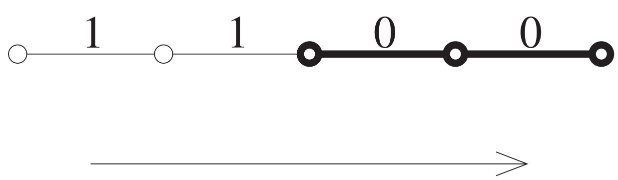


Border thinning algorithm ?

- $\mathscr{B}$-thinnings:
  - Rely on a local condition
  - Adapted to parallel strategies

Problem

- *But a same edge can be lowered several times*



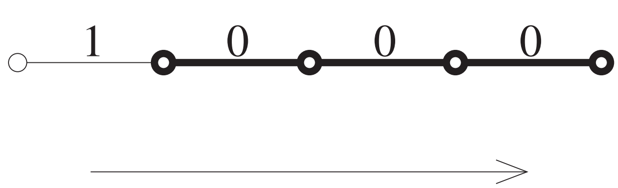
Thinnings: watershed algorithm

Border thinning algorithm ?

- $\mathscr{B}$-thinnings:
  - Rely on a local condition
  - Adapted to parallel strategies

Problem

- *But a same edge can be lowered several times*



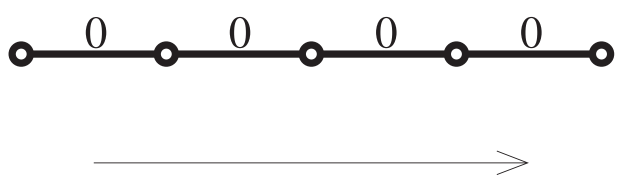
J. Serra, J. Cousty, B.S. Daya Sagar : Course on Math. Morphology 7/34

Border thinning algorithm ?

- $\mathscr{B}$-thinnings:
  - Rely on a local condition
  - Adapted to parallel strategies

Problem

- *But a same edge can be lowered several times*



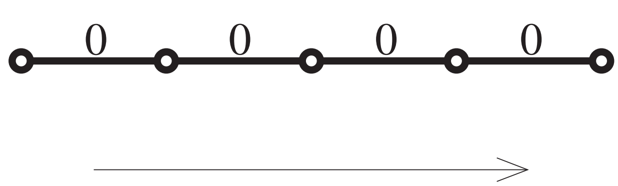
Thinnings: watershed algorithm

Border thinning algorithm ?

- $\mathscr{B}$-thinnings:
  - Rely on a local condition
  - Adapted to parallel strategies

Problem

- *But a same edge can be lowered several times*
- *Naive sequential algorithm runs in $O(n^2)$*



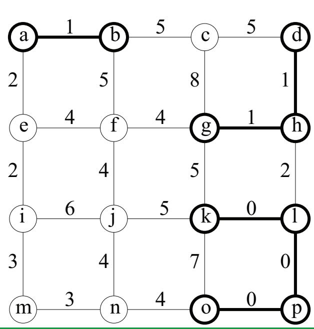
J. Serra, J. Cousty, B.S. Daya Sagar : Course on Math. Morphology 7/34

Towards a sequential linear-time algorithm . . .

Towards a sequential linear-time algorithm . . .

Definition

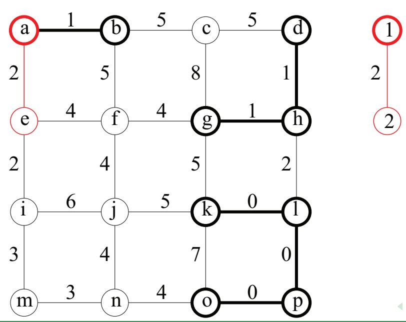
- *An edge u is* M-border (for *F*) *if u is a border edge for F and if one of its vertices belongs to a minimum of F*



Towards a sequential linear-time algorithm . . .

Definition

- *An edge u is* M-border (for F) *if u is a border edge for F and if one of its vertices belongs to a minimum of F*

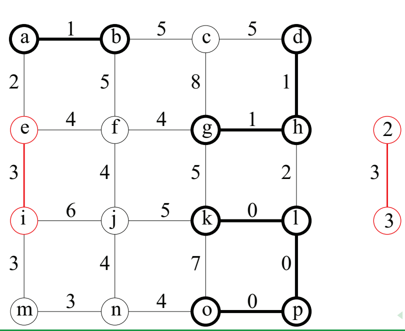


Thinnings: watershed algorithm

Towards a sequential linear-time algorithm . . .

Definition

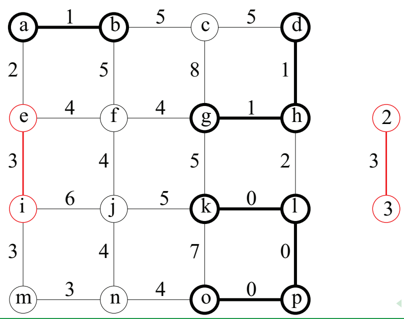
An edge u is M-border (for F) if u is a border edge for F and if one of its vertices belongs to a minimum of F



Towards a sequential linear-time algorithm . . .

Definition

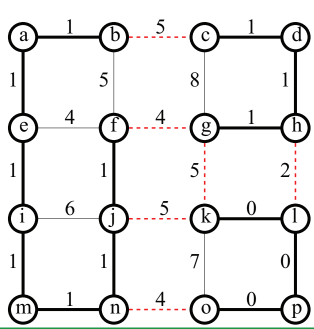
- *An edge u is* M-border (for *F*) *if u is a border edge for F and if one of its vertices belongs to a minimum of F*
- *We can then define* $\mathscr{M}$-thinnings, $\mathscr{M}$-kernels and $\mathscr{M}$-cuts of *F*



Towards a sequential linear-time algorithm . . .

Definition

- *An edge u is* M-border (for *F*) *if u is a border edge for F and if one of its vertices belongs to a minimum of F*
- *We can then define* $\mathscr{M}$-thinnings, $\mathscr{M}$-kernels and $\mathscr{M}$-cuts of *F*



$\mathscr{M}$-kernels, $\mathscr{M}$-cuts & watersheds

Theorem

- *A graph X is an MSF relative to the minima of F* if and only if *X is the graph of the minima of a $\mathscr{M}$-kernel of F*
- *An edge set $S \subseteq E$ is an $\mathscr{M}$-cut of F* if and only if *S is a watershed cut of F*

$\mathscr{M}$-kernel Algorithm

Data: (V, E, F): an edge-weighted graph
Result: F: an $\mathscr{M}$-kernel of the input map, and its minima (V_M, E_M)

$L \leftarrow \emptyset$;
Compute $M(F) = (V_M, E_M)$ and $F^{\ominus}(x)$ for each $x \in V$;
foreach *$u \in E$ outgoing from (V_M, E_M)* **do** $L \leftarrow L \cup \{u\}$;
while *there exists $u \in L$* **do**
    $L \leftarrow L \setminus \{u\}$;
    if *u is border for F* **then**
        $x \leftarrow$ the vertex in u such that $F^{\ominus}(x) < F(u)$;
        $y \leftarrow$ the vertex in u such that $F^{\ominus}(y) = F(u)$;
        $F(u) \leftarrow F^{\ominus}(x)$; $F^{\ominus}(y) \leftarrow F(u)$;
        $V_M \leftarrow V_M \cup \{y\}$; $E_M \leftarrow E_M \cup \{u\}$;
        foreach *$v = \{y', y\} \in E$ with $y' \notin V_M$* **do** $L \leftarrow L \cup \{v\}$;

$\mathscr{M}$-kernel algorithm: analysis

Results

- *Any edge is lowered at most once*

Thinnings: watershed algorithm

$\mathscr{M}$-kernel algorithm: analysis

Results

- *Any edge is lowered at most once*
- Linear-time $(O(|V| + |E|))$ whatever the range of F
  - *No need to sort*
  - *No need to use a hierarchical/priority queue*
  - *No need to use union-find structure*

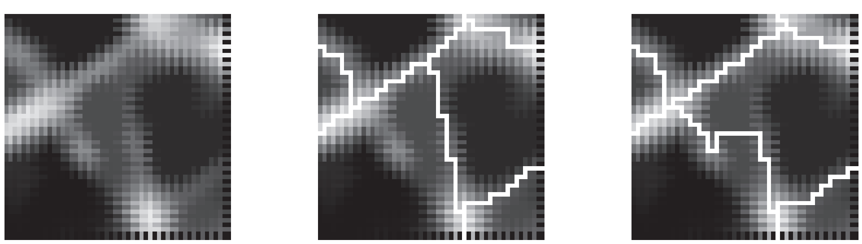
J. Serra, J. Cousty, B.S. Daya Sagar : Course on Math. Morphology 11/34

$\mathscr{M}$-kernel algorithm: analysis

Results

- *Any edge is lowered at most once*
- Linear-time *($O(|V| + |E|)$)* whatever the range of F
  - *No need to sort*
  - *No need to use a hierarchical/priority queue*
  - *No need to use union-find structure*
- *The only required data structure is a list for the set L*

Watershed on plateaus?



(a) (b) (c)

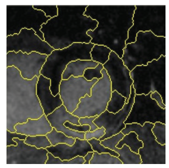
(a) Representation of an edge weighted graph (4-adjacency)

- Watersheds computed by $\mathscr{B}$-kernel algorithms implementing set L
  - (b) as a LIFO list
  - (c) as a priority queue with a FIFO breaking ties policy

Watershed: pracical problem #2

Problem

In practice: over-segmentation



Over-segmentation

Solution 2

- **Seeded watershed** (or marker based watershed)

Over-segmentation

Solution 2

- **Seeded watershed** (or marker based watershed)
- Methodology proposed by Beucher and Meyer (1993)
  1. **Recognition**
  2. **Delineation** (generally done by watershed)
  3. **Smoothing**

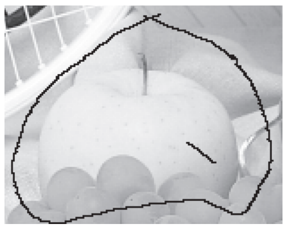
Over-segmentation

Solution 2

- **Seeded watershed** (or marker based watershed)
- Methodology proposed by Beucher and Meyer (1993)
  1. **Recognition**
  2. **Delineation** (generally done by watershed)
  3. **Smoothing**
- **Semantic information** taken into account at steps 1 and 3

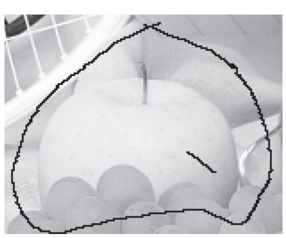
Seeded watershed

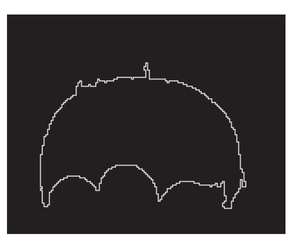
- Seeded segmentation is very popular
  - A user “marks by seeds” the object that are to be segmented

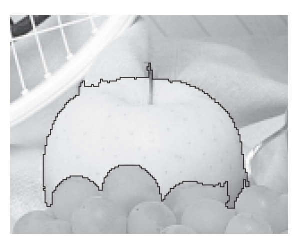


Seeded watershed

- Seeded segmentation is very popular
  - A user “marks by seeds” the object that are to be segmented
- MSF cuts fall into this category







Seeded watershed

- Seeded segmentation is very popular
  - A **automated procedure** "marks by seeds" the object that are to be segmented
- MSF cuts fall into this category

Seeded watershed

- Seeded segmentation is very popular
  - A **automated procedure** "marks by seeds" the object that are to be segmented
- MSF cuts fall into this category

A morphological solution

- *Mathematical morphology is adapted to the design of such automated recognition procedure*

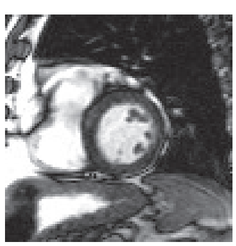
Seeded watershed: application

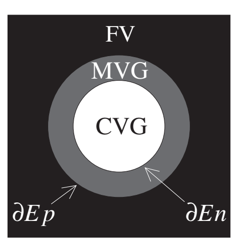
- *Myocardium segmentation in 3D+t ciné MRI*
  - J. Cousty *et al.*, 2007

Seeded watershed: application

- *Myocardium segmentation in 3D+t ciné MRI*
  - J. Cousty *et al.*, 2007

- Cross-section by cross-section acquisition





Seeded watershed: application

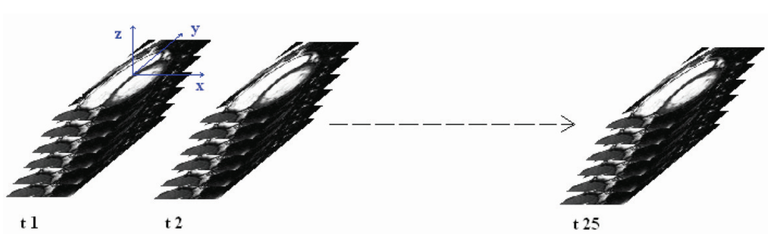
- *Myocardium segmentation in 3D+t ciné MRI*
  - J. Cousty *et al.*, 2007
- Cross-section by cross-section acquisition
- First, along time (ECG gated)

im

Thinnings: watershed algorithm

Seeded watershed: application

- *Myocardium segmentation in 3D+t ciné MRI*
  - J. Cousty *et al.*, 2007
- Cross-section by cross-section acquisition
- First, along time (ECG gated)
- Then, in space



Seeded watershed: application

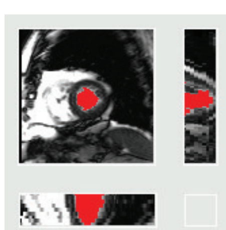
- *Endocardial segmentation:*

Thinnings: watershed algorithm

Seeded watershed: application

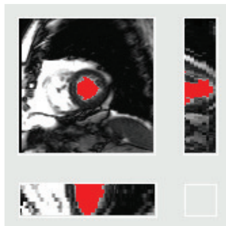
- *Endocardial segmentation:*

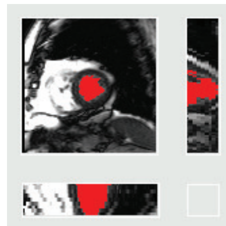
- Upper threshold (recognition)



Seeded watershed: application

- *Endocardial segmentation:*
- Upper threshold (recognition)
- Geodesic dilation in a lower threshold (delineation)



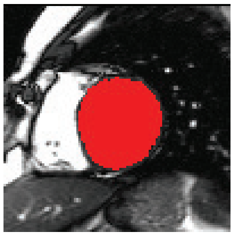


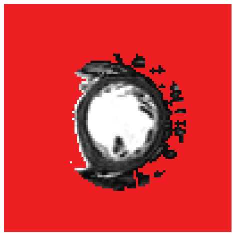
Seeded watershed: application

- *Epicardial segmentation:*

Seeded watershed: application

- *Epicardial segmentation:*
- Internal and external markers (recognition):
  - Repulsed dilation
  - Homotopic dilation



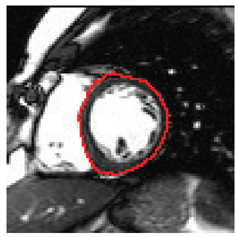


Seeded watershed: application

- *Epicardial segmentation:*
- Internal and external markers (recognition):
  - Repulsed dilation
  - Homotopic dilation
- Watershed in 4D space

Seeded watershed: application

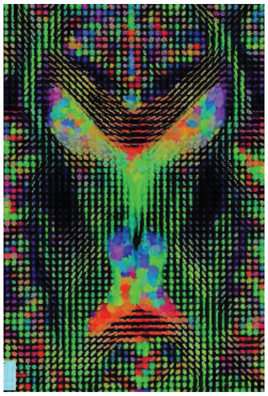
- *Epicardial segmentation:*
- Internal and external markers (recognition):
  - Repulsed dilation
  - Homotopic dilation
- Watershed in 4D space
- Smoothing (alternated sequential filters)



Seeded watershed: application

res

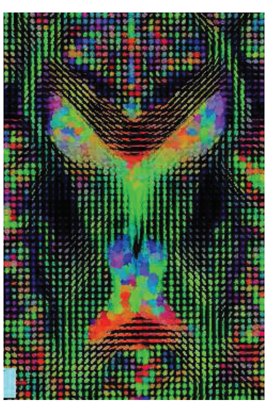
Seeded watershed for Diffusion Tensor Images (DTIs)



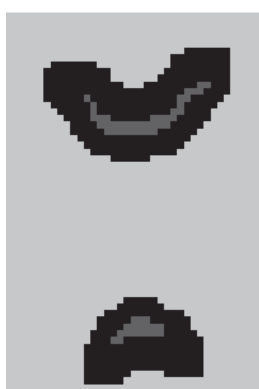
DTI

- 3D Diffusion Tensor Image equipped with the direct adjacency
- Edges weighted by the Log-Euclidean distance between tensors

Seeded watershed for Diffusion Tensor Images (DTIs)



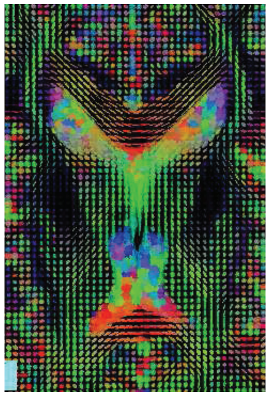
DTI



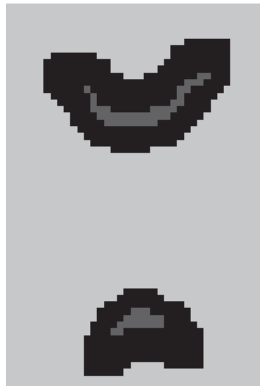
seeds

- 3D Diffusion Tensor Image equipped with the direct adjacency
- Edges weighted by the Log-Euclidean distance between tensors

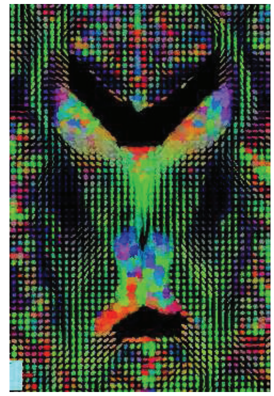
Seeded watershed for Diffusion Tensor Images (DTIs)



DTI



seeds



segmentation by MSF cuts

- 3D Diffusion Tensor Image equipped with the direct adjacency
- Edges weighted by the Log-Euclidean distance between tensors
- Seeds automatically obtained from a statistical atlas

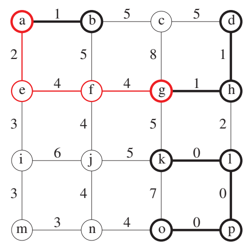
Discrete optimization for seeded segmentation

- Minimum spanning forests
- **Shortest paths spanning forests**
- **Min-cuts**
- **Random Walkers**

Connection value

Definition

- *Let* $\pi = \langle x_0, \ldots, x_\ell \rangle$ *be a path in* G.
  - $\Upsilon_F(\pi) = \max\{F(\{x_{i-1}, x_i\}) \mid i \in [1, \ell]\}$

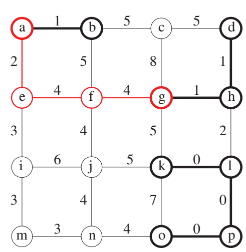


- $\Upsilon_F(\langle a, e, f, g \rangle) = 4$

Connection value

Definition

- *Let* $\pi = \langle x_0, \ldots, x_\ell \rangle$ *be a path in* G.
  - $\Upsilon_F(\pi) = \max\{F(\{x_{i-1}, x_i\}) \mid i \in [1, \ell]\}$
- *The connection value between two points* x *and* y *is*
  - $\Upsilon_F(x, y) = \min\{\Upsilon_F(\pi) \mid \pi \text{ path from } x \text{ to } y\}$

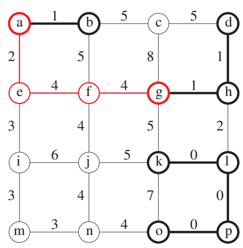


- $\Upsilon_F(\langle a, e, f, g \rangle) = 4$
- $\Upsilon_F(a, g) = 4$

Connection value

Definition

- *Let* $\pi = \langle x_0, \ldots, x_\ell \rangle$ *be a path in* G.
  - $\Upsilon_F(\pi) = \max\{F(\{x_{i-1}, x_i\}) \mid i \in [1, \ell]\}$
- *The* connection value between two points x and y *is*
  - $\Upsilon_F(x, y) = \min\{\Upsilon_F(\pi) \mid \pi$ *path from* x *to* $y\}$
- *The* connection value between two subgraphs X and Y *is*
  - $\Upsilon_F(X, Y) = \min\{\Upsilon_F(x, y) \mid x \in V(X), y \in V(Y)\}$



- $\Upsilon_F(\langle a, e, f, g \rangle) = 4$
- $\Upsilon_F(a, g) = 4$
- $\Upsilon_F(\{a, b\}, \{g, h, d\})$

Subdominant ultrametric

Remark

- *The connection value is a (ultrametric) distance in a graph*

MSFs preserve connection values

Theorem

- **If** Y *is an MSF relative to* X,
- **Then**, *for any two distinct components* A *et* B *of* X :
  - $\Upsilon_F(A, B) = \Upsilon_F(A', B')$
- *where* A' *et* B' *are the two components of* Y *that contains* A *et* B

Shortest paths spanning forests

Remark

- *The connection value is a (ultrametric) distance in a graph*

Shortest paths spanning forests

Definition

- *Let X be a graph (the seeds)*
- *We say that Y is a* shortest path forest relative to X *if*
  - *Y is a forest relative to$X$ and*
  - *for any $x \in V(Y)$, there exists, from x to X, a path π in Y such that $F(\pi) = F(\{x\}, X)$*

MSFs and shortest paths forests

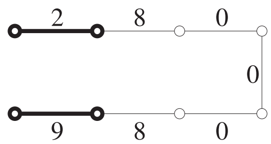
Property

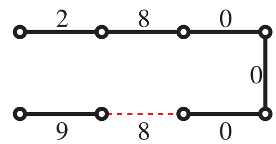
- *If Y is a MSF relative to X, then Y is a shortest path spanning forest relative to X*

MSFs and shortest paths forests

Property

- *If Y is a MSF relative to X, then Y is a shortest path spanning forest relative to X*



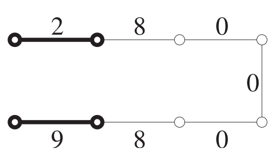


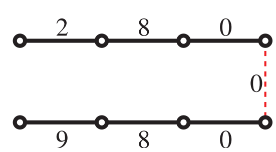
Thinnings: watershed algorithm

MSFs and shortest paths forests

Property

- *If Y is a MSF relative to X, then Y is a shortest path spanning forest relative to X*



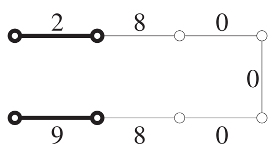


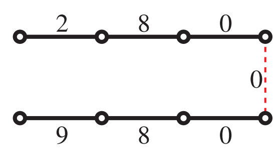
J. Serra, J. Cousty, B.S. Daya Sagar : Course on Math. Morphology 27/34

MSFs and shortest paths forests

Property

- *If Y is a MSF relative to X, then Y is a shortest path spanning forest relative to X*





Remark

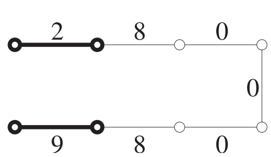
- *The converse is, in general, not true*

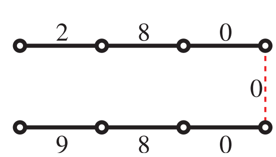
Thinnings: watershed algorithm

MSFs and shortest paths forests

Property

- *If Y is a MSF relative to X, then Y is a shortest path spanning forest relative to X*





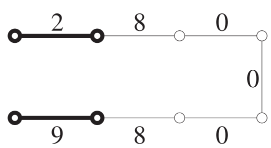
Remark

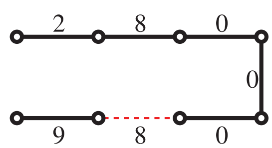
- *The converse is, in general, not true*
- *No connection value preservation*

MSFs and shortest paths forests

Property

- *If Y is a MSF relative to X, then Y is a shortest path spanning forest relative to X*

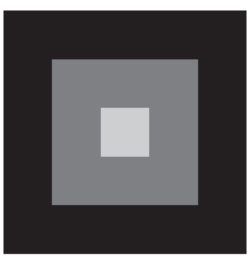




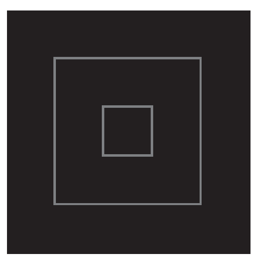
Remark

- *The converse is, in general, not true*
- *No connection value preservation*

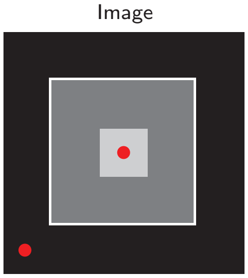
Synthetic image example



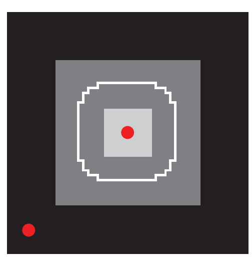
Image



Dissimilarities



MSF cut (white) - seeds (red)



SPF cut (white) - seeds (red)

Shortest path forests and watersheds

Property

- *The graph X is a shortest path spanning forest relative to the minima of F if and only if X is an MSF relative to the minima of F*

Shortest path forests and watersheds

Property

- *The graph X is a shortest path spanning forest relative to the minima of F if and only if X is an MSF relative to the minima of F*

Property

- *Let X be a graph (the seeds)*
- *A subset S of E is a watershed of the flooding of F by X if and only if S is a cut induced by a shortest path spanning forest relative to X*

Min-cuts

Definition

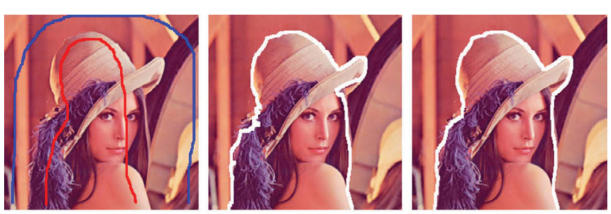
- *Let X be a graph (the seeds)*
- *Let $C \subseteq E$ be a cut relative to X*
- *The cut C is called a minimum cut (min-cut) relative to X if, for any cut C' relative to X we have $F(C) \leq F(C')$*

Thinnings: watershed algorithm

Min-cuts

Definition

- *Let X be a graph (the seeds)*
- *Let $C \subseteq E$ be a cut relative to X*
- *The cut C is called a minimum cut (min-cut) relative to X if, for any cut C' relative to X we have $F(C) \leq F(C')$*



(a) (b) (c)

a: an image with seeds X in red and blue, b (resp. c): MSF cut (resp. min-cut) relative to X (white) where F is the gradient of (a) (resp. its inverse) [from Allène et al., IVC 2010]

Weight transformation

Thinnings: watershed algorithm

Weight transformation

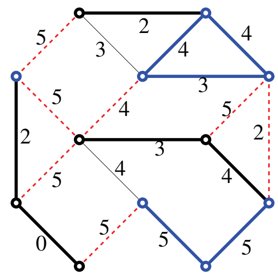
Let g be a decreasing (resp. increasing) map in $\mathbb{R}^+$

J. Serra, J. Cousty, B.S. Daya Sagar : Course on Math. Morphology 31/34

Weight transformation

Let g be a decreasing (resp. increasing) map in $\mathbb{R}^+$

- X MINimum SF for F iff X is a MAXimum SF (resp. MINSF) for $g \circ F$

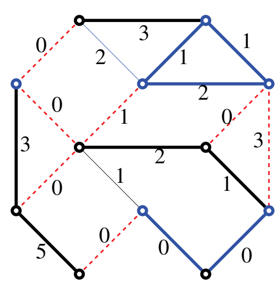


Thinnings: watershed algorithm

Weight transformation

Let g be a decreasing (resp. increasing) map in $\mathbb{R}^+$

- X MINimum SF for F iff X is a MAXimum SF (resp. MINSF) for $g \circ F$



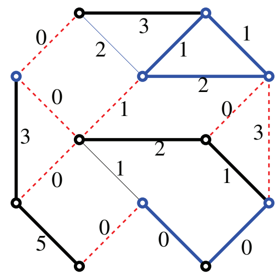
J. Serra, J. Cousty, B.S. Daya Sagar : Course on Math. Morphology 31/34

Weight transformation

Let g be a decreasing (resp. increasing) map in $\mathbb{R}^+$

- X MINimum SF for F iff X is a MAXimum SF (resp. MINSF) for $g \circ F$

Let F^p : $F^p(u) = [F(u)]^p$



Thinnings: watershed algorithm

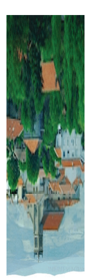
Weight transformation

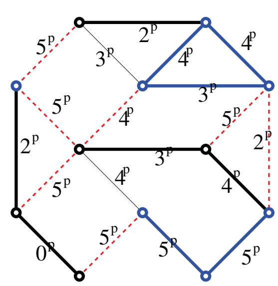
Let g be a decreasing (resp. increasing) map in $\mathbb{R}^+$

- X MINimum SF for F iff X is a MAXimum SF (resp. MINSF) for $g \circ F$

Let F^p : $F^p(u) = [F(u)]^p$

- X MAXSF for F^p iff X MAXSF for F





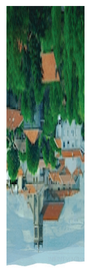
Weight transformation

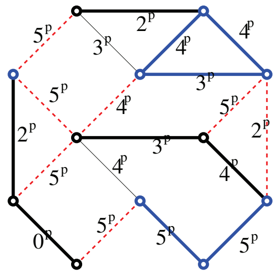
Let g be a decreasing (resp. increasing) map in $\mathbb{R}^+$

- X MINimum SF for F iff X is a MAXimum SF (resp. MINSF) for $g \circ F$

Let F^p : $F^p(u) = [F(u)]^p$

- X MAXSF for F^p iff X MAXSF for F
- Property not verified by min-cuts





Watershed & min-cuts

Theorem

- *There exists a real k such that for any $p \geq k$*
  - *any min-cut for F^p is a MAXSF cut for F^p*

- Allène et al., *Some links between extremum spanning forests, watersheds and min-cuts*, IVC 2010

Watershed & min-cuts: illustration [Allène2010]

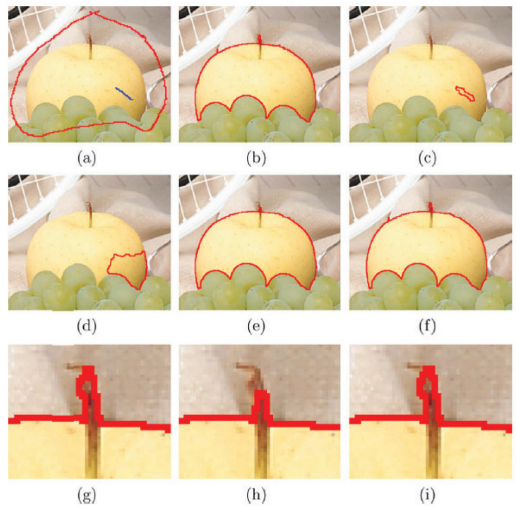


Fig. 12. Color image segmentation using: (a) markers superimposed to the original image; (b) MaxSF cut on P; (c) min-cut on P; (d) min-cut on $P^{[1.4]}$; (e) min-cut on $P^{[2]}$; (f) min-cut on $P^{[3]}$; (g) zoom of MaxSF cut on P; (h) zoom of min-cut on $P^{[2]}$; and (i) zoom of min-cut on $P^{[3]}$.

Random walks

- Similar results hold true for random walks segmentation
- See L. Najman's talk next week

Summary

- Defining watershed in discrete spaces is difficult
  - Grayscale image as vertex weighted graphs
  - Region merging problems
  - The large familly of watersheds
- Watershed in edge weighted graphs
  - Watershed cuts: a consistent framework
  - Minimum spanning forests: watershed optimality
  - Thinnings: watershed algorithms
- Seeded segmentation
  - Watershed in practical applications
  - Comparison with other method of combinatorial optimization