Le point de vue d'un théoricien sur l'intérêt de la généricité pour le traitement d'images

Pourquoi suis-je intéressé par Olena, ou le changement des représentations des images

Laurent Najman

Séminaire EPITA - 21 mars 2012

[1]Université Paris-Est
LIGM A3SI ESIEE Paris, France

Mathematical morphology

Definition

- Lattice: $(E, \leq)$ *a space* E *with an* ordering *relationship* $\leq$
- *for all* x *and* y*, a larger element* $x \vee y$ *and a smaller element* $x \wedge y$.
- Dilation: *an operator* δ *such that* $\delta(\bigvee X_i) = \bigvee \delta(X_i)$
- Erosion: *an operator* ε *such that* $\varepsilon(\bigwedge X_i) = \bigwedge \delta(X_i)$

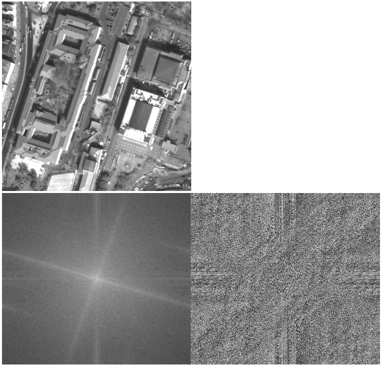
Important idea

The structure of the space E can be whatever is needed by the applications.

Image representations

Decomposition into primitive or fundamental elements that can be more easily interpreted:

- Functional decomposition;
- Multiresolution decomposition;
- Multi-scale representation;
- Skeleton representation;
- Threshold decomposition;
- Hierarchical representations.



Amplitude Phase

Image representations

Decomposition into primitive or fundamental elements that can be more easily interpreted:

- Functional decomposition;
- Multiresolution decomposition;
- Multi-scale representation;
- Skeleton representation;
- Threshold decomposition;
- Hierarchical representations.

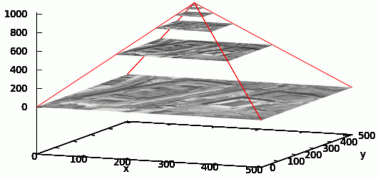


Image representations

Decomposition into primitive or fundamental elements that can be more easily interpreted:

- Functional decomposition;
- Multiresolution decomposition;
- Multi-scale representation;
- Skeleton representation;
- Threshold decomposition;
- Hierarchical representations.

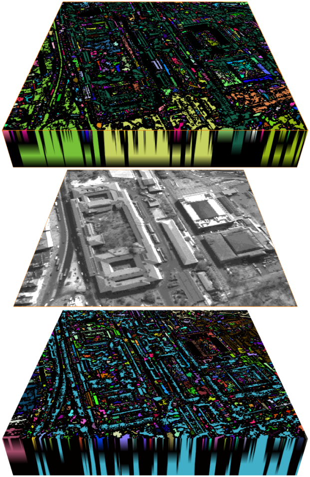


Image representations

Decomposition into primitive or fundamental elements that can be more easily interpreted:

- Functional decomposition;
- Multiresolution decomposition;
- Multi-scale representation;
- Skeleton representation;
- Threshold decomposition;
- Hierarchical representations.

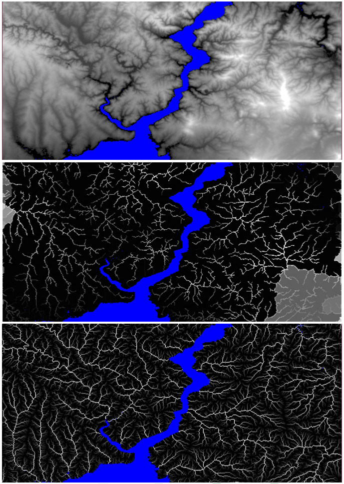


Image representations

Decomposition into primitive or fundamental elements that can be more easily interpreted:

- Functional decomposition;
- Multiresolution decomposition;
- Multi-scale representation;
- Skeleton representation;
- Threshold decomposition;
- Hierarchical representations.

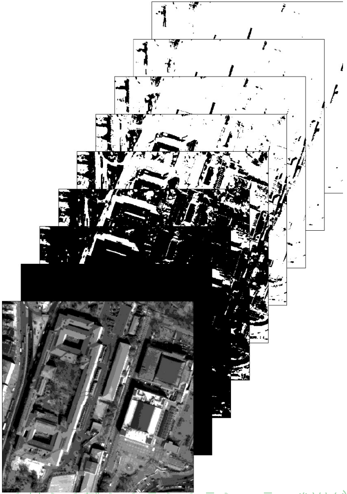


Image representations

Decomposition into primitive or fundamental elements that can be more easily interpreted:

- Functional decomposition;
- Multiresolution decomposition;
- Multi-scale representation;
- Skeleton representation;
- Threshold decomposition;
- Hierarchical representations.

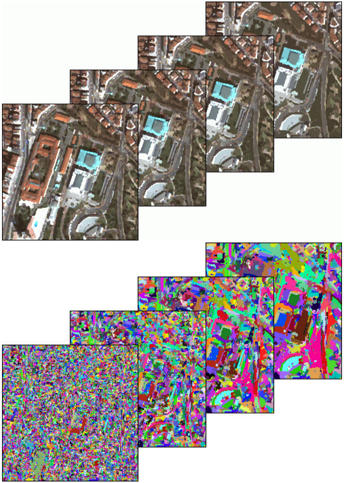


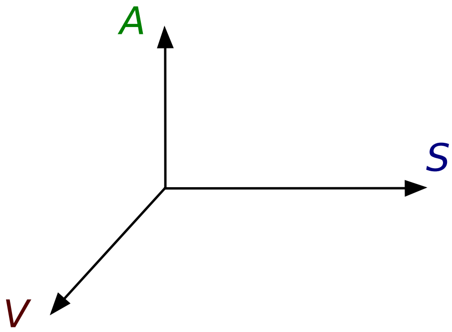
Image representations

Decomposition into primitive or fundamental elements that can be more easily interpreted:

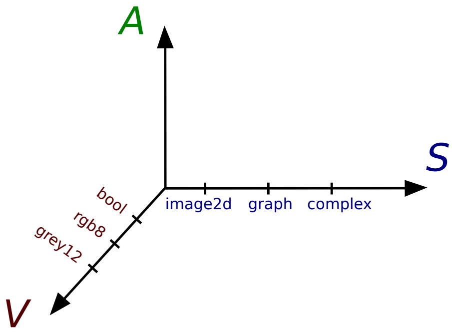
- Functional decomposition;
- Multiresolution decomposition;
- Multi-scale representation;
- Skeleton representation;
- Threshold decomposition;
- Hierarchical representations.

Not mutually exclusive.
Properties inherited from those of underlying operations.
Choice driven by the application needs.

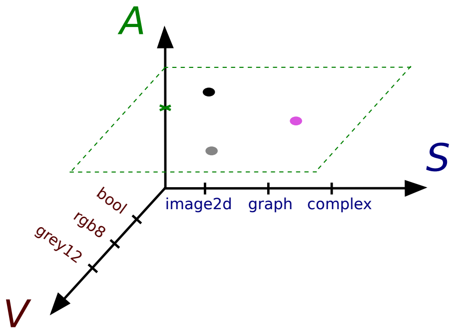
Problem due to diversity of images representations



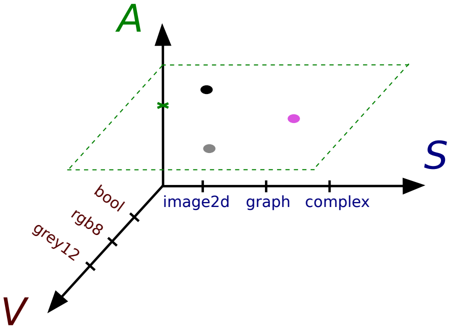
Problem due to diversity of images representations



Problem due to diversity of images representations

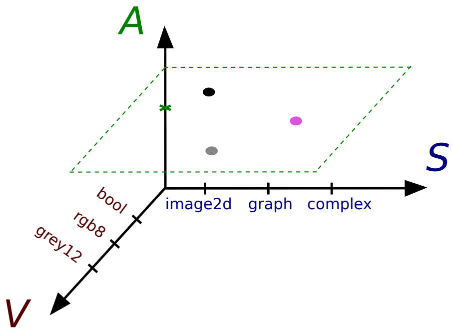


Problem due to diversity of images representations



- High combinatorial complexity: $A \times S(\times V)$ implementations.

Problem due to diversity of images representations

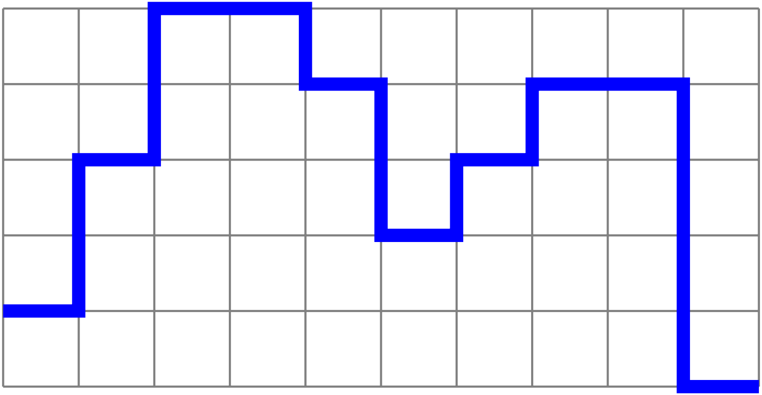


- High combinatorial complexity: $A \times S (\times V)$ implementations.
- The Olena project is a (the?) solution to that problem.

1 Tree-based representations
- Max-tree, Min-tree and inclusion tree
- Increasing attributes
- Non-increasing attributes

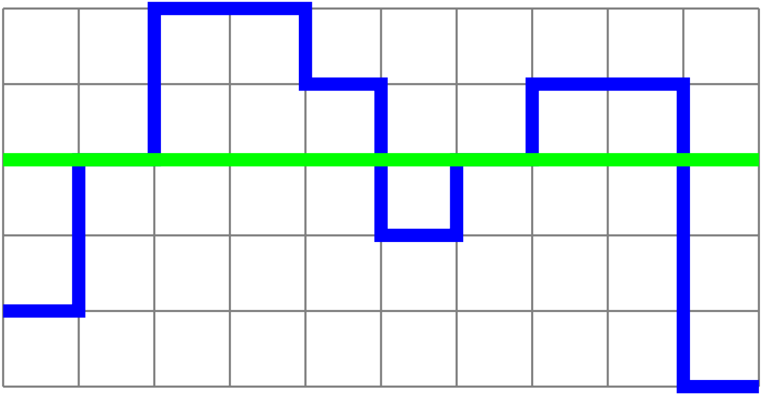
2 Watersheds and hierarchies
- Watershed-based hierarchical segmentation schemes
- Hierarchical clustering
- Hierarchical image segmentation schemes
- Hierarchical segmentation as a watershed-based scheme

Level sets and components



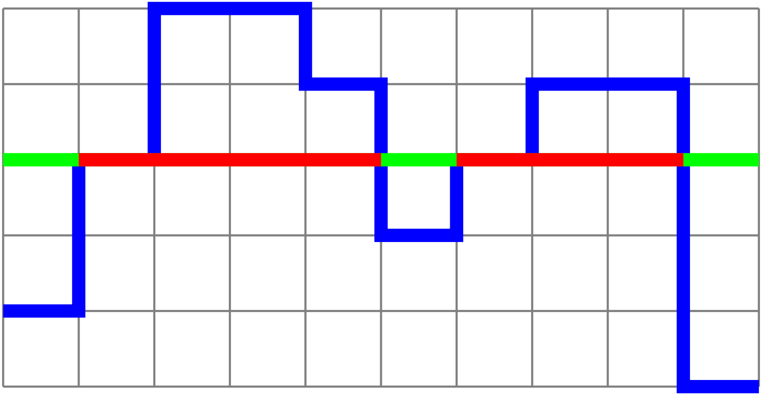
$y = F(x)$

Level sets and components



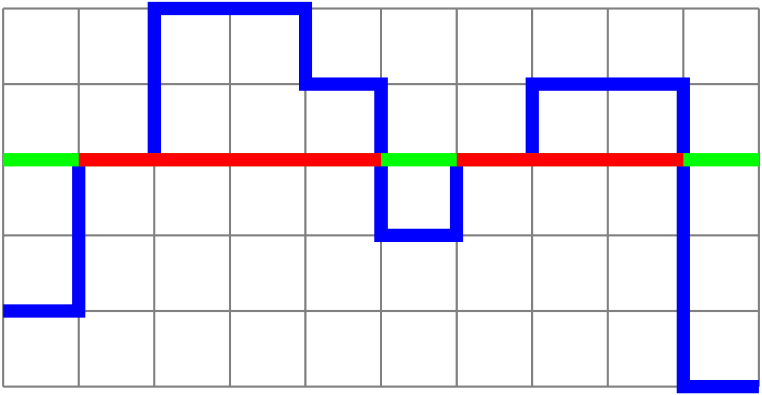
$F(x) \geq 3$

Level sets and components



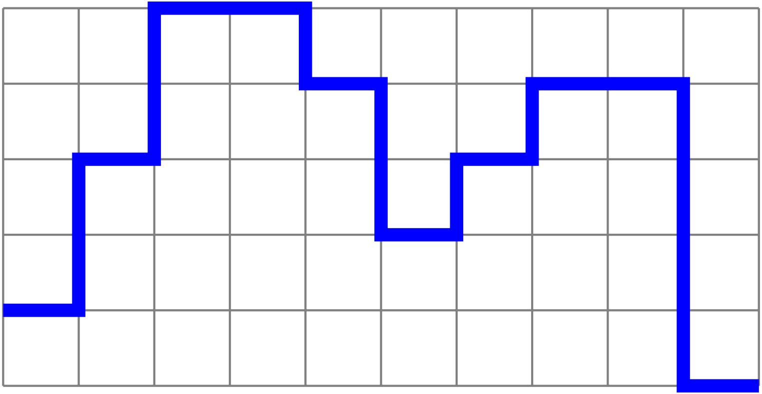
$$F_3 = \{x \mid F(x) \geq 3\}$$

Level sets and components

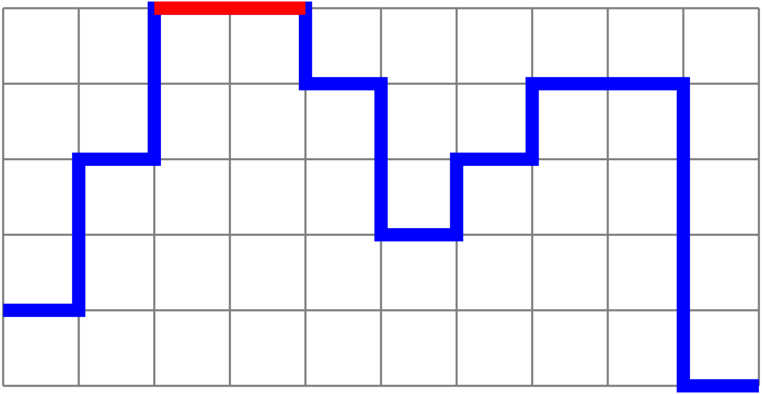


$$F_k = \{x \mid F(x) \geq k\}$$

(Max) component tree

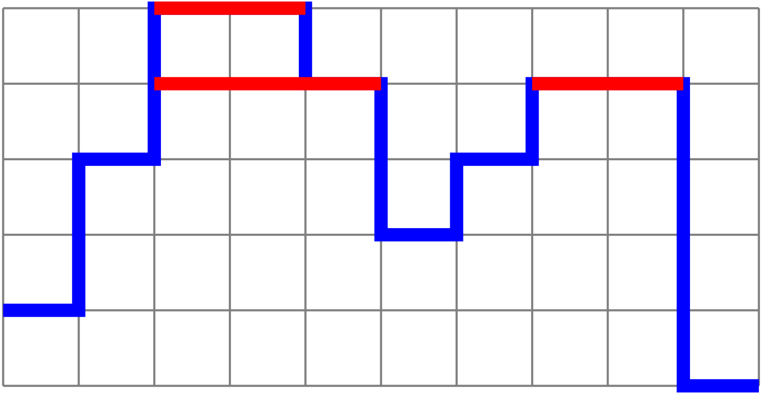


(Max) component tree



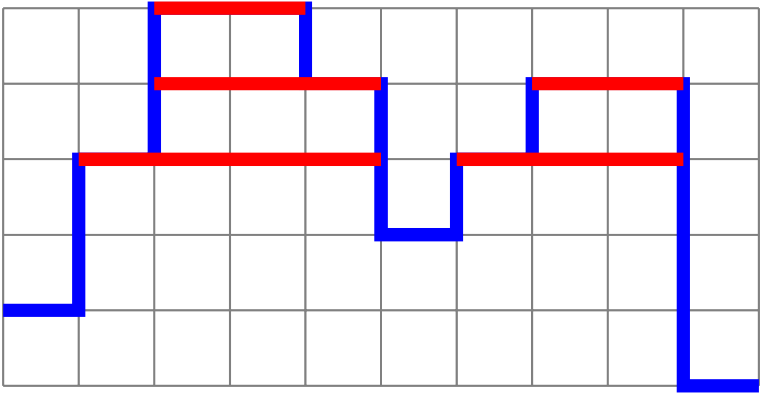
$$F_5 = \{x \mid F(x) \geq 5\}$$

(Max) component tree



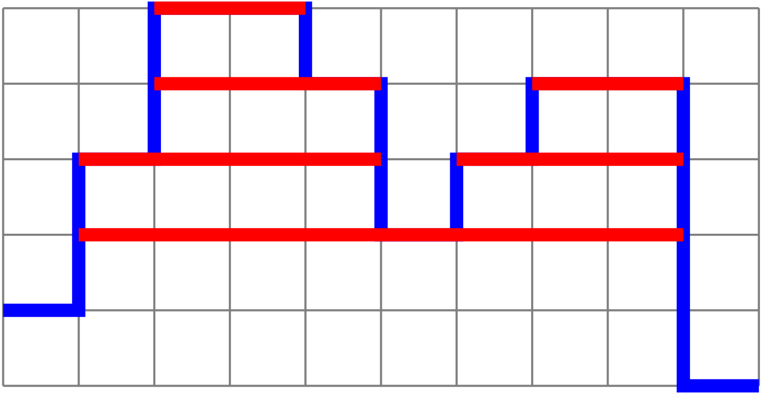
$$F_4 = \{x \mid F(x) \geq 4\}$$

(Max) component tree



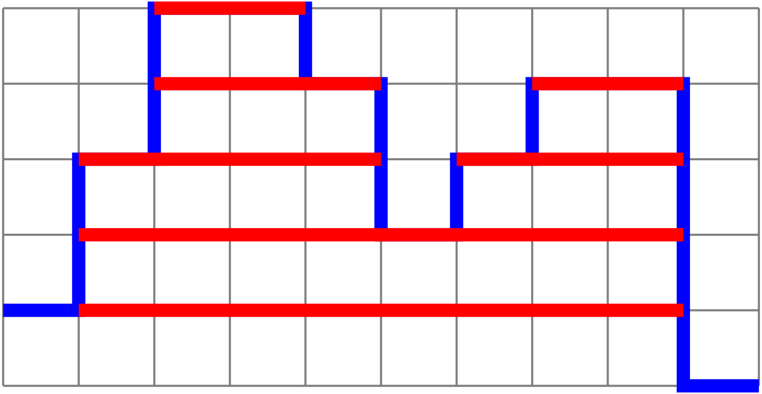
$$F_3 = \{x \mid F(x) \geq 3\}$$

(Max) component tree



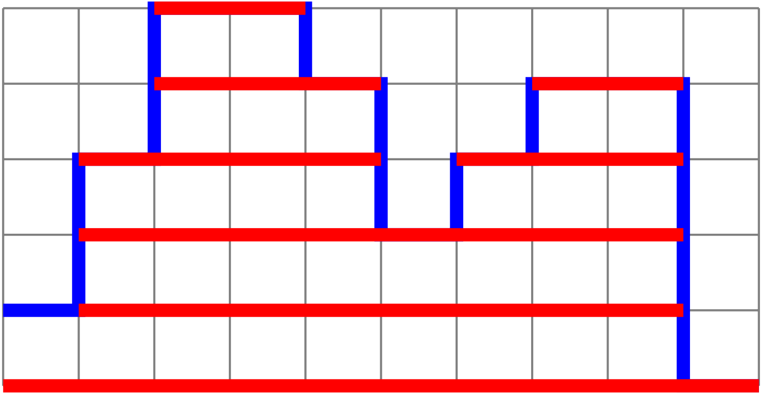
$$F_2 = \{x \mid F(x) \geq 2\}$$

(Max) component tree



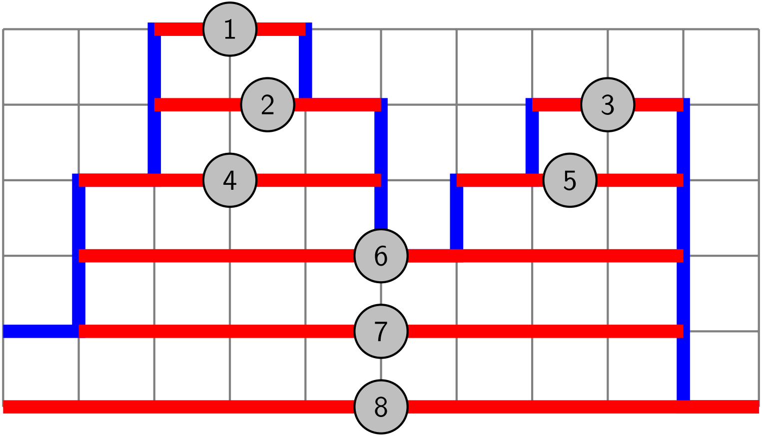
$$F_1 = \{x \mid F(x) \geq 1\}$$

(Max) component tree

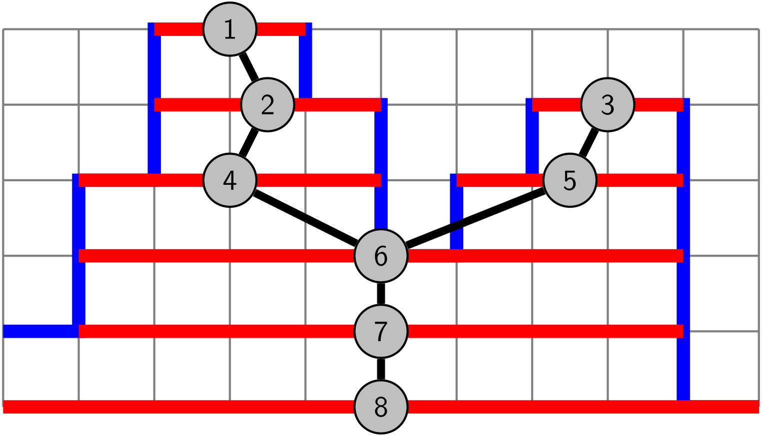


$$F_0 = \{x \mid F(x) \geq 0\}$$

(Max) component tree

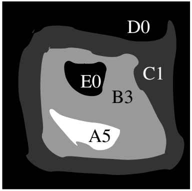


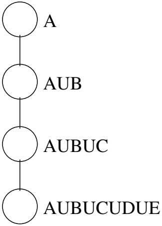
(Max) component tree



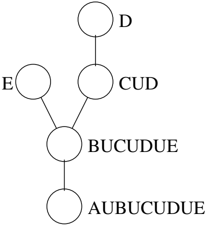
Components + inclusion relationship = component tree

Min-tree, max-tree and inclusion tree

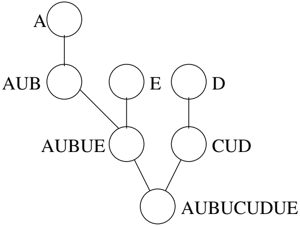




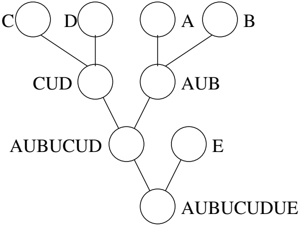
Max−tree



Min−tree

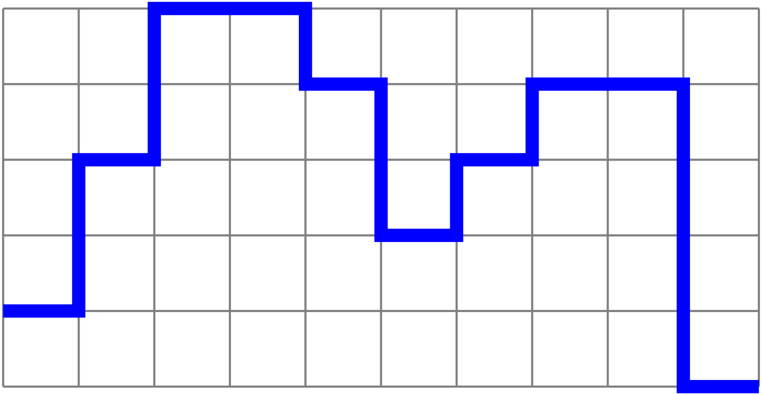


Inclusion Tree

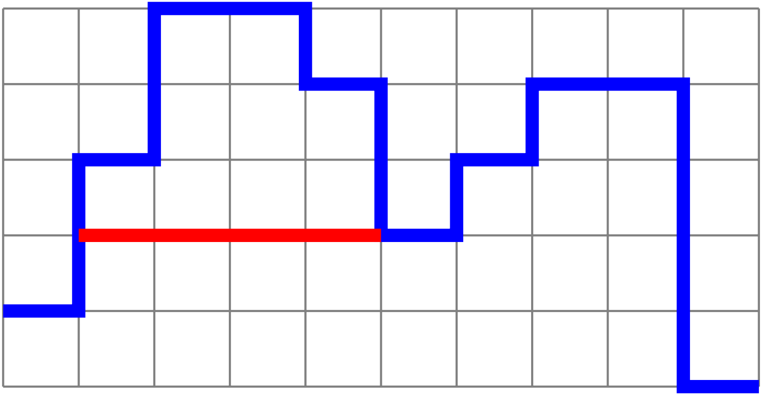


Binary Partition Tree

Attributes

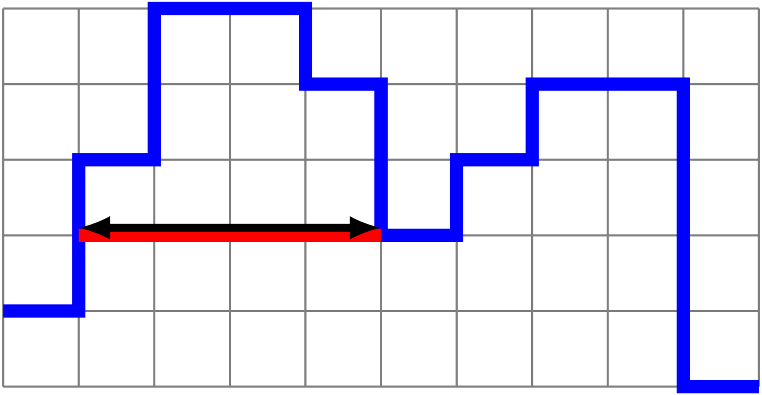


Attributes



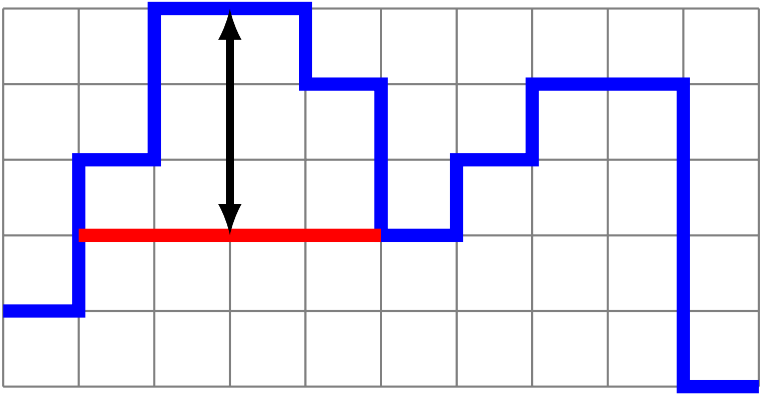
A connected component

Attributes



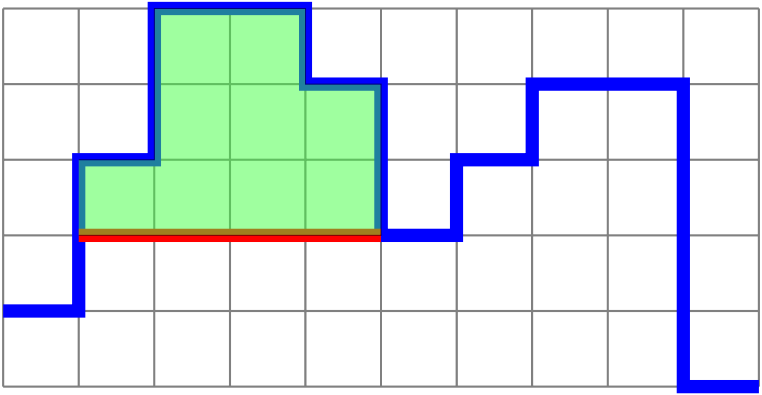
Area

Attributes



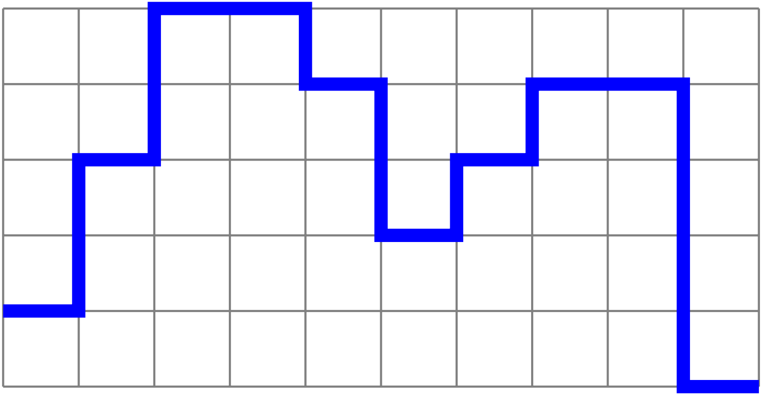
Height

Attributes

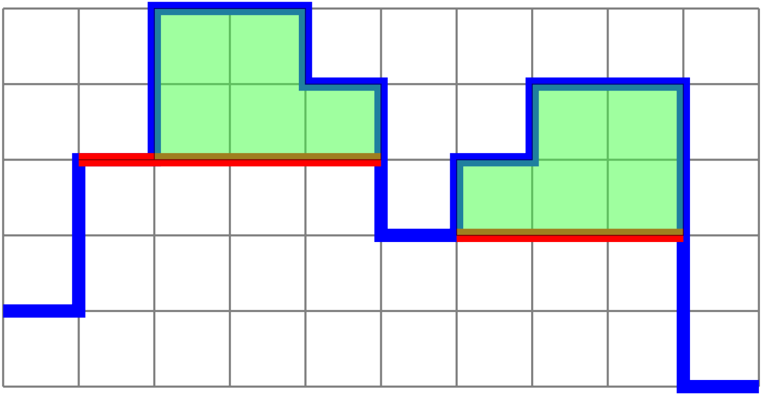


Volume

Filtering

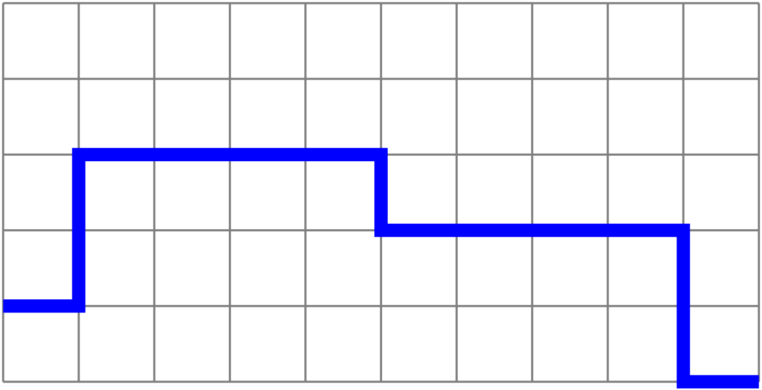


Filtering



Volume ≤ 5

Filtering



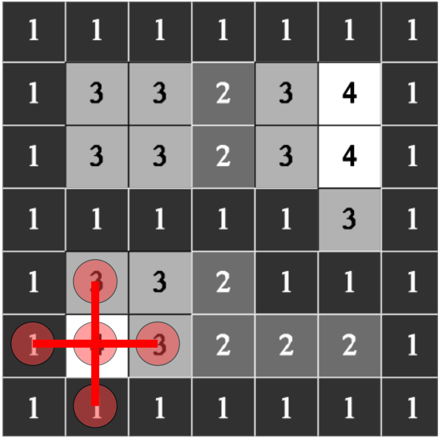
Filtered function

On a 2D image

1	1	1	1	1	1	1
1	3	3	2	3	4	1
1	3	3	2	3	4	1
1	1	1	1	1	3	1
1	3	3	2	1	1	1
1	4	3	2	2	2	1
1	1	1	1	1	1	1

A matrix

On a 2D image



4-connectivity

On a 2D image

1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1

$$F_1 = \{x \mid F(x) \geq 1\}$$

On a 2D image

0	0	0	0	0	0	0
0	1	1	1	1	1	0
0	1	1	1	1	1	0
0	0	0	0	0	1	0
0	1	1	1	0	0	0
0	1	1	1	1	1	0
0	0	0	0	0	0	0

$$F_2 = \{x \mid F(x) \geq 2\}$$

On a 2D image

0	0	0	0	0	0	0
0	1	1	0	1	1	0
0	1	1	0	1	1	0
0	0	0	0	0	1	0
0	1	1	0	0	0	0
0	1	1	0	0	0	0
0	0	0	0	0	0	0

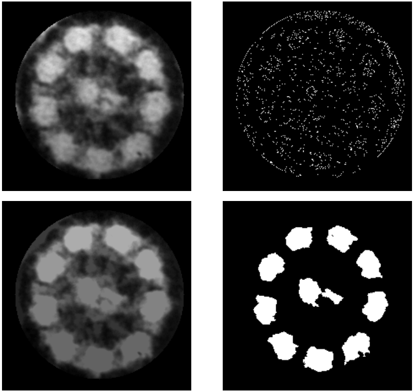
$$F_3 = \{x \mid F(x) \geq 3\}$$

On a 2D image

0	0	0	0	0	0	0
0	0	0	0	0	1	0
0	0	0	0	0	1	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	1	0	0	0	0	0
0	0	0	0	0	0	0

$$F_4 = \{x \mid F(x) \geq 4\}$$

Application



Question

- *Increasing criterion (here, volume)*
- *How to process non-increasing criteria?*

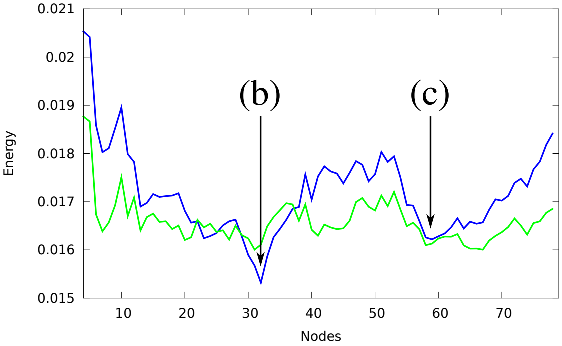
An example of a non increasing criterion: Yongchao's energy

Important idea

$$E(u, \partial\tau) = E_{int}(u, \partial\tau) + E_{ext}(u, \partial\tau) + E_{con}(u, \partial\tau)$$

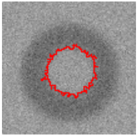
$$\begin{aligned}
V(u, \mathcal{R}) &= \sum_{p \in \mathcal{R}} (u(p) - \overline{u}(\mathcal{R}))^2 \\
E_{ext}(u, \partial\tau) &= \frac{V(u,\, \mathcal{R}^{\varepsilon}_{in}(\partial\tau)) + V(u,\, \mathcal{R}^{\varepsilon}_{out}(\partial\tau))}{V(u,\, \mathcal{R}^{\varepsilon}_{in}(\partial\tau) \cup \mathcal{R}^{\varepsilon}_{out}(\partial\tau))}. \\
E_{int}(u, \partial\tau) &= \sum_{e \in \partial\tau} |curv(u)(e)| \;/\; L(\partial\tau), \\
E_{con}(u, \partial\tau) &= 1 \;/\; L(\partial\tau).
\end{aligned}$$

An example of a non increasing criterion: Yongchao's energy

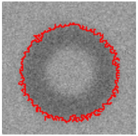


(a): Energy in a branch of the tree

- blue : our energy
- green : snake energy

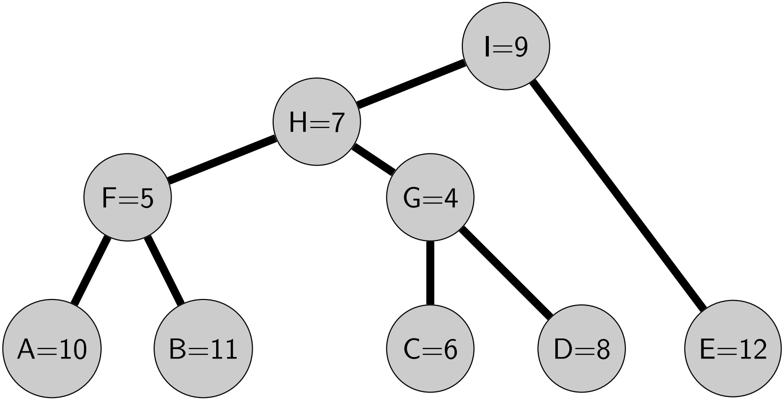


(b)

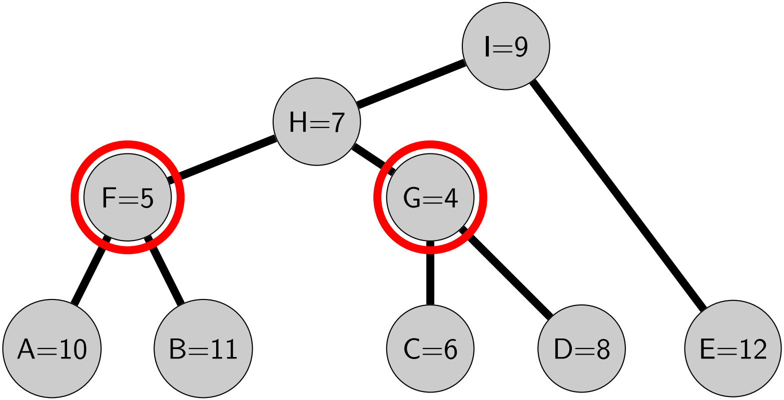


(c)

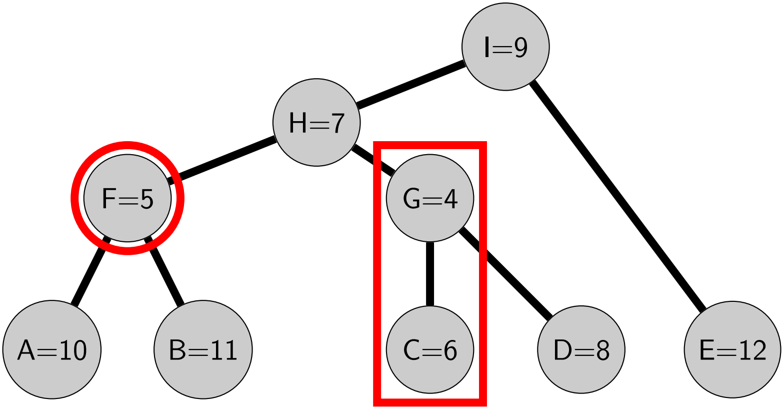
Processing non-increasing criterion



Processing non-increasing criterion

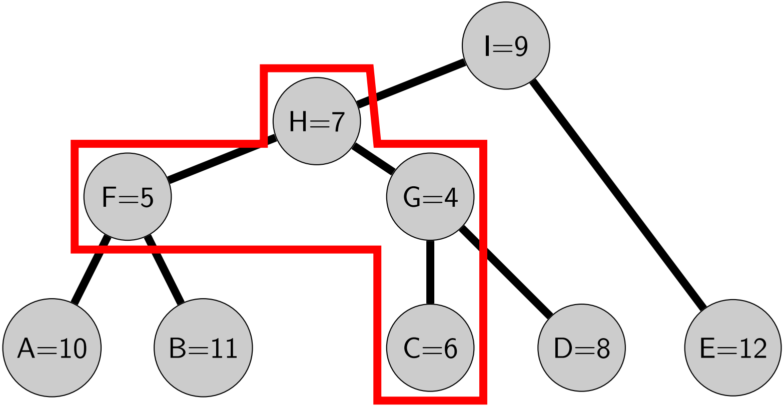


Processing non-increasing criterion



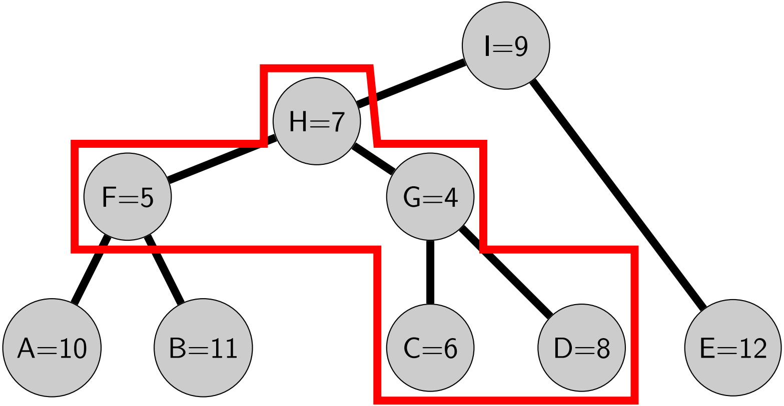
Level 6

Processing non-increasing criterion



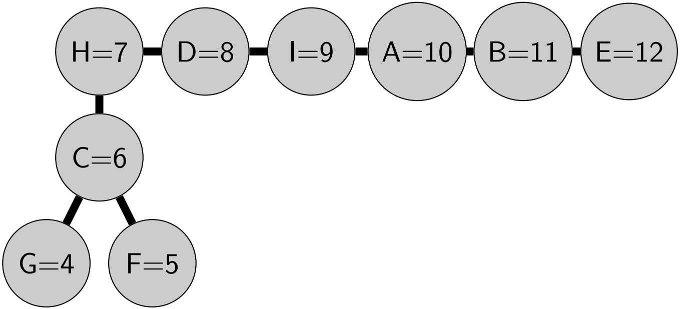
Level 7

Processing non-increasing criterion



Level 8

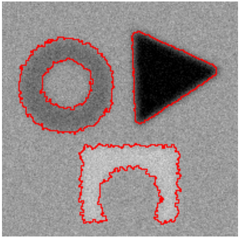
Processing non-increasing criterion: through min-tree

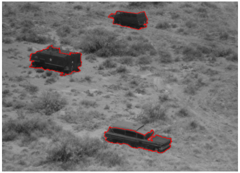


Important idea

Computing a Min-Tree on a node-weighted graph instead of a matrix image
Easy thanks to Olena!

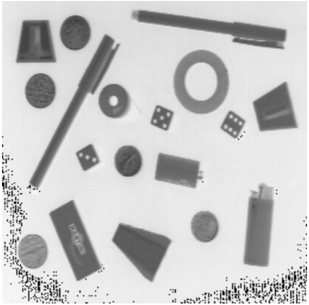
Spotting objects with Yongchao's criterion



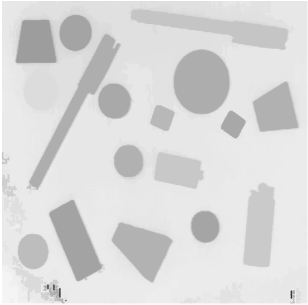


Meaningfull minima of the energy

Filtering in space of shapes with Yongchao's criterion

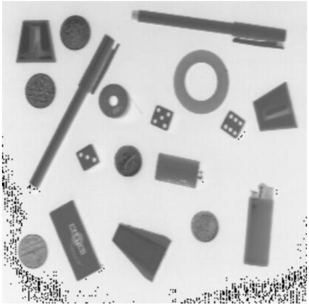


Original image



Filter height=0.1

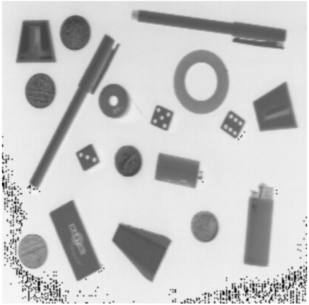
Filtering in space of shapes with Yongchao's criterion



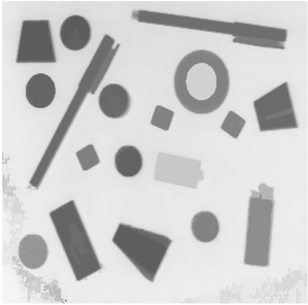
Original image

Filter volume=2

Filtering in space of shapes with Yongchao's criterion



Original image



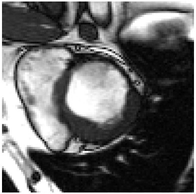
Filter volume=0.9

Watershed segmentation

- 1978: introduction of “the” watershed as a segmentation tool.

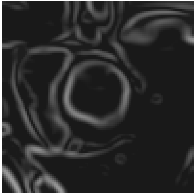
Watershed segmentation

- 1978: introduction of “the” watershed as a segmentation tool.



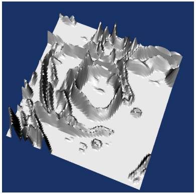
Watershed segmentation

- 1978: introduction of "the" watershed as a segmentation tool.



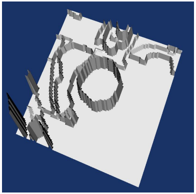
Watershed segmentation

- 1978: introduction of "the" watershed as a segmentation tool.



Watershed segmentation

- 1978: introduction of “the” watershed as a segmentation tool.



Watershed segmentation

- 1978: introduction of “the” watershed as a segmentation tool.

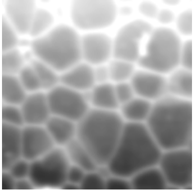
Watershed segmentation

- 1978: introduction of "the" watershed as a segmentation tool.

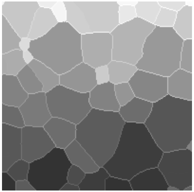
Hypothesis

- There exists numerous watershed definitions and algorithms.
- The image is seen as a graph with values on nodes.

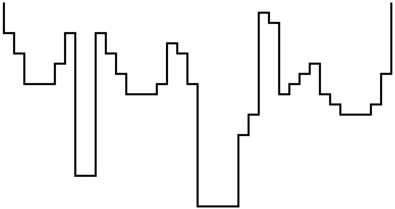
Illustration: topological watershed [Bertrand 2005, Couprie *et al.* 2005, JMIV]



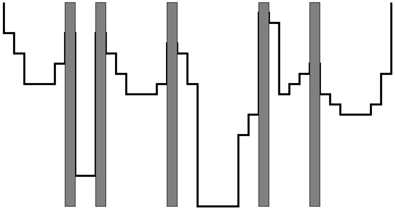
Topological



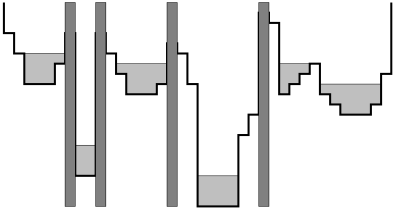
Hierarchies: floodings and watersheds



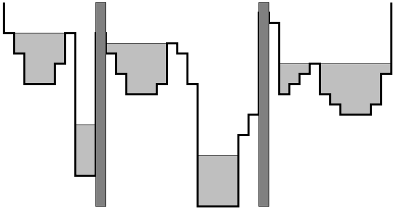
Hierarchies: floodings and watersheds



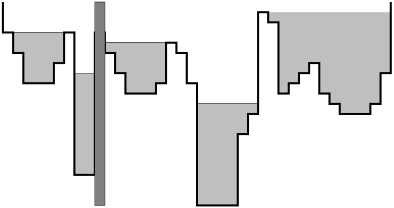
Hierarchies: floodings and watersheds



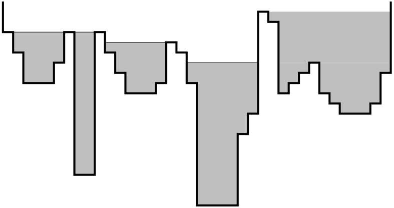
Hierarchies: floodings and watersheds



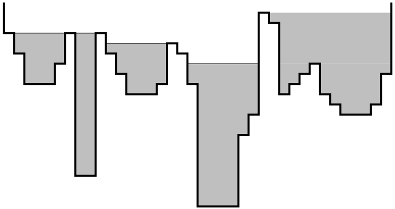
Hierarchies: floodings and watersheds



Hierarchies: floodings and watersheds



Hierarchies: floodings and watersheds

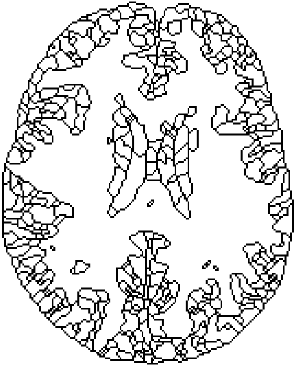


Important idea

- *There exists numerous criterions for flooding a surface.*
- *Flooding can be done through the min-(component-)tree.*
- *Among those criterions, notably: depth, surface, volume.*
- *[Beucher, ISMM, 1994 - Najman & Schmitt, PAMI, 1996 - Meyer et al., An. Telecom, 1997]*

Saliency map

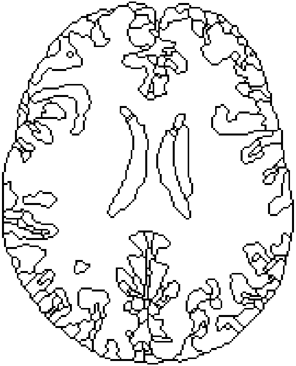
(a) Original image



(b) Some contours

Saliency map

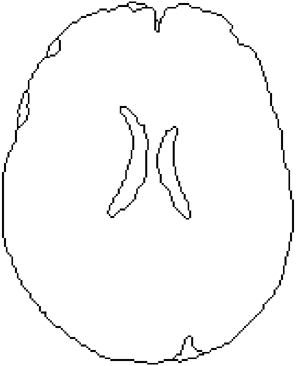
(a) Original image



(b) Some contours

Saliency map

(a) Original image

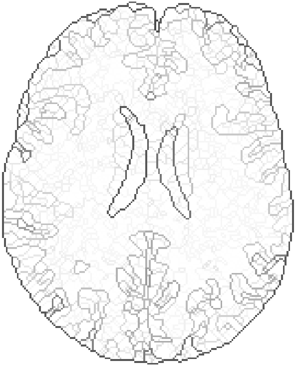


(b) Some contours

Saliency map

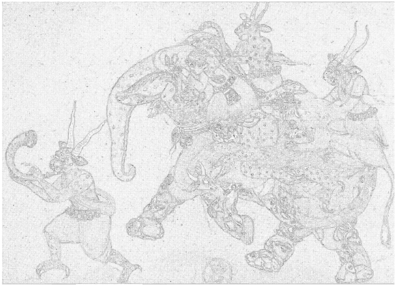
Stacking the contours gives a saliency map [Najman & Schmitt, PAMI, 1996]

(a) Original image



(b) A saliency map

Some examples



Depth driven hierarchy

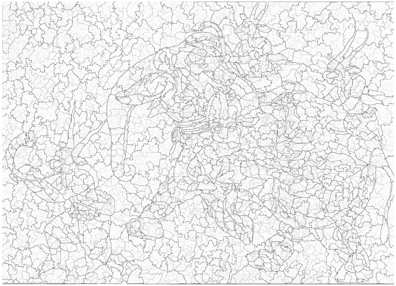
One of the segmentations

Some examples

Area driven hierarchy

One of the segmentations

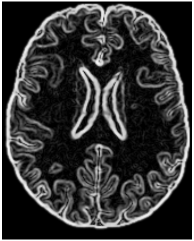
Some examples

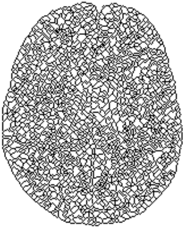


Volume driven hierarchy

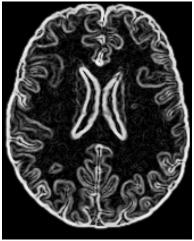
One of the segmentations

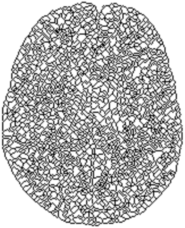
Region merging problems on pixels

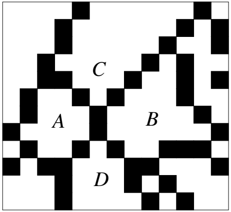




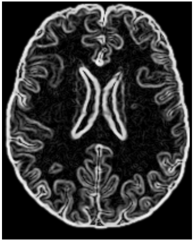
Region merging problems on pixels

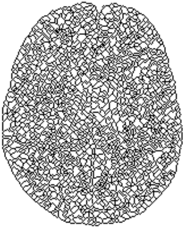


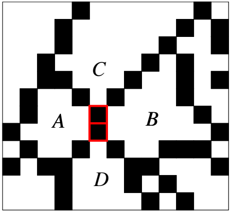




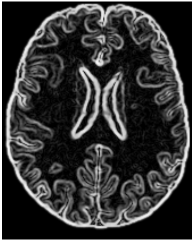
Region merging problems on pixels

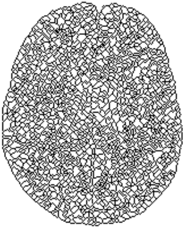


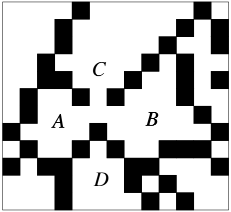




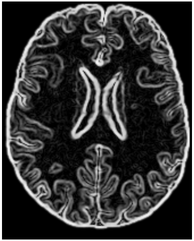
Region merging problems on pixels

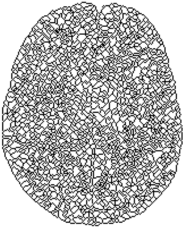


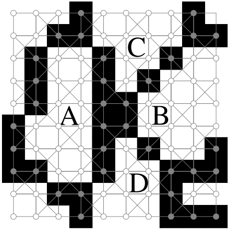




Region merging problems on pixels



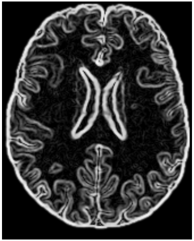


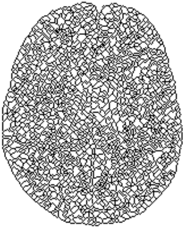


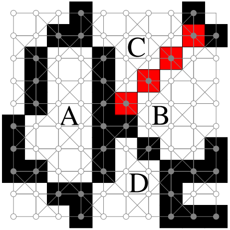
Important idea

- *On the nodes:* **Fusion graphs** *[Cousty et al. - JMIV - 2008, Cousty et al. - DAM - 2008]*
- *Thanks to Olena, this new grid is easy to deal with!*

Region merging problems on pixels



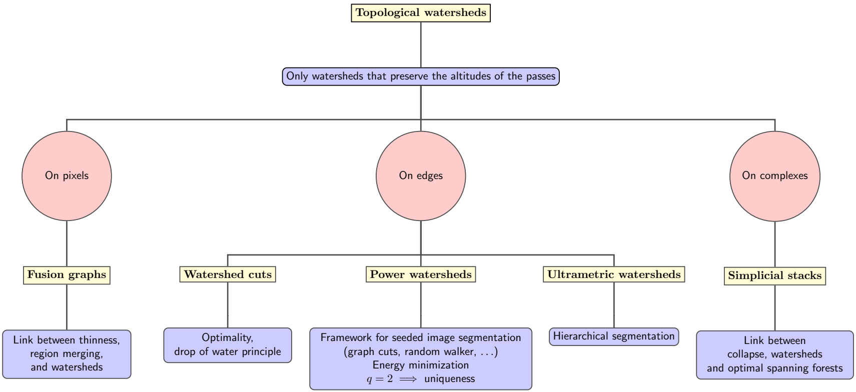




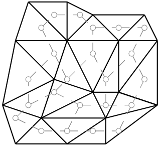
Important idea

- *On the nodes:* **Fusion graphs** *[Cousty et al. - JMIV - 2008, Cousty et al. - DAM - 2008]*
- *Thanks to Olena, this new grid is easy to deal with!*

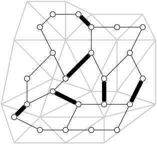
A family of watersheds



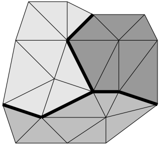
Surface segmentation by watershed



Mesh



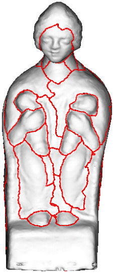
Cut



Segmentation

Statue

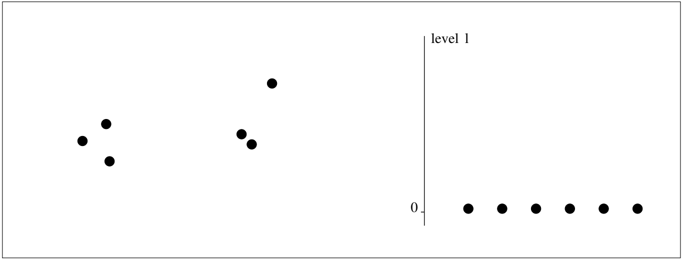
Watershed



Filtered watershed

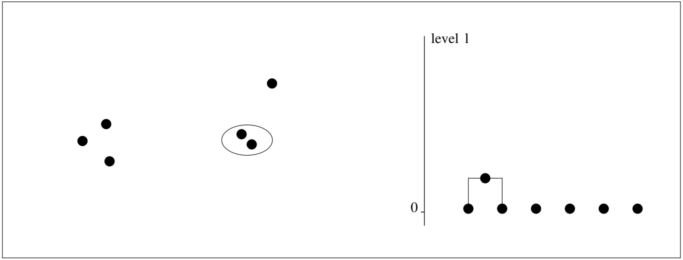
Definition and representation

- Sequence of nested clusters such that a cluster at a given level is formed by unioning clusters existing at the previous level;
- The level, denoted by λ, is a non-negative real number controlling the coarseness degree of the clustering;
- Dendrograms (sometimes called taxonomic trees) are commonly used to represent hierarchies [Sokal & Sneath, 1963]:



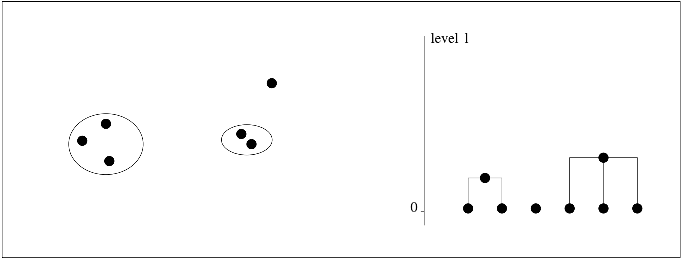
Definition and representation

- Sequence of nested clusters such that a cluster at a given level is formed by unioning clusters existing at the previous level;
- The level, denoted by λ, is a non-negative real number controlling the coarseness degree of the clustering;
- Dendrograms (sometimes called taxonomic trees) are commonly used to represent hierarchies [Sokal & Sneath, 1963]:



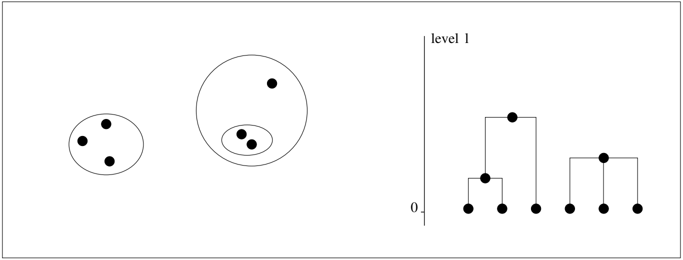
Definition and representation

- Sequence of nested clusters such that a cluster at a given level is formed by unioning clusters existing at the previous level;
- The level, denoted by λ, is a non-negative real number controlling the coarseness degree of the clustering;
- Dendrograms (sometimes called taxonomic trees) are commonly used to represent hierarchies [Sokal & Sneath, 1963]:



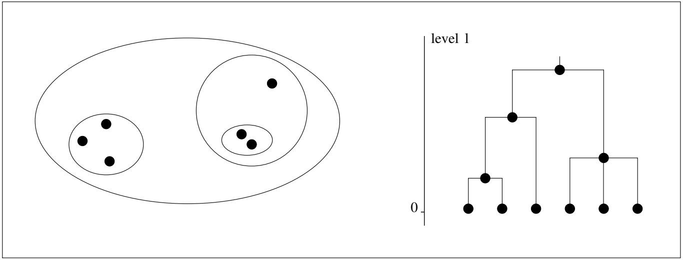
Definition and representation

- Sequence of nested clusters such that a cluster at a given level is formed by unioning clusters existing at the previous level;
- The level, denoted by λ, is a non-negative real number controlling the coarseness degree of the clustering;
- Dendrograms (sometimes called taxonomic trees) are commonly used to represent hierarchies [Sokal & Sneath, 1963]:



Definition and representation

- Sequence of nested clusters such that a cluster at a given level is formed by unioning clusters existing at the previous level;
- The level, denoted by λ, is a non-negative real number controlling the coarseness degree of the clustering;
- Dendrograms (sometimes called taxonomic trees) are commonly used to represent hierarchies [Sokal & Sneath, 1963]:



Some existing methods [**not** based on watersheds]

- Segmentation tree [Horowitz & Pavlidis, JACM 1976];
- Hierarchical stepwise optimisation [Beaulieu & Goldberg, PAMI 1986];
- Shortest spanning tree segmentation [Morris et al., IEE Proc. 1986];
- Pyramid of region adjacency graphs [Montanvert et al. 1991];
- Graph weighted hierarchy [Kropatsch & Haximusa, SPIE-5299 2004];
- Scale-sets: cuts minimising an energy based on complexity and distortion measures [Guigues *et al*, IJCV 2006];
- Constrained connectivity [Soille, PAMI 2008].

Picture Segmentation by a Tree Traversal Algorithm

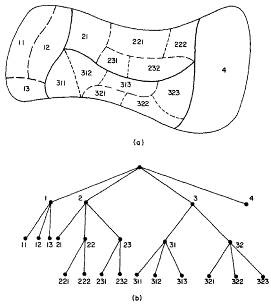


FIG. 2. (a) Example of a directed region segmentation; (b) tree representing the above segmentation

Some existing methods [**not** based on watersheds]

- Segmentation tree
  [Horowitz & Pavlidis, JACM 1976];
- Hierarchical stepwise optimisation
  [Beaulieu & Goldberg, PAMI 1986];
- Shortest spanning tree segmentation
  [Morris et al., IEE Proc. 1986];
- Pyramid of region adjacency graphs
  [Montanvert et al. 1991];
- Graph weighted hierarchy [Kropatsch & Haximusa, SPIE-5299 2004];
- Scale-sets: cuts minimising an energy based on complexity and distortion measures [Guigues *et al*, IJCV 2006];
- Constrained connectivity [Soille, PAMI 2008].

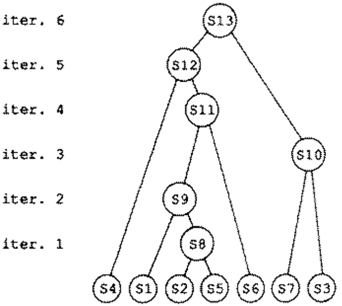


Fig. 3. Sequence of segment merges.

Some existing methods [**not** based on watersheds]

- Segmentation tree
  [Horowitz & Pavlidis, JACM 1976];
- Hierarchical stepwise optimisation
  [Beaulieu & Goldberg, PAMI 1986];
- Shortest spanning tree segmentation
  [Morris et al., IEE Proc. 1986];
- Pyramid of region adjacency graphs
  [Montanvert et al. 1991];
- Graph weighted hierarchy [Kropatsch & Haximusa, SPIE-5299 2004];
- Scale-sets: cuts minimising an energy based on complexity and distortion measures [Guigues *et al*, IJCV 2006];
- Constrained connectivity [Soille, PAMI 2008].

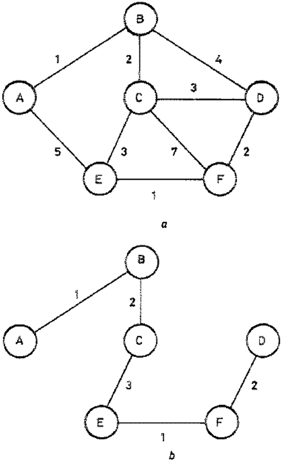
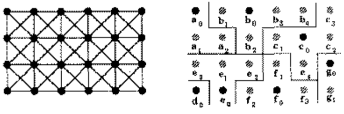


Fig. 1 *Example of a graph and its SST*
a Example graph
b SST of graph

IEE PROCEEDINGS, Vol. 133, Pt. F, No. 2, APRIL 1986

Some existing methods [**not** based on watersheds]

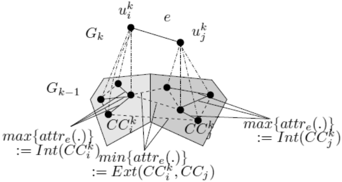
- Segmentation tree [Horowitz & Pavlidis, JACM 1976];
- Hierarchical stepwise optimisation [Beaulieu & Goldberg, PAMI 1986];
- Shortest spanning tree segmentation [Morris et al., IEE Proc. 1986];
- Pyramid of region adjacency graphs [Montanvert et al. 1991];
- Graph weighted hierarchy [Kropatsch & Haximusa, SPIE-5299 2004];
- Scale-sets: cuts minimising an energy based on complexity and distortion measures [Guigues *et al*, IJCV 2006];
- Constrained connectivity [Soille, PAMI 2008].



(a) (b)

Some existing methods [**not** based on watersheds]

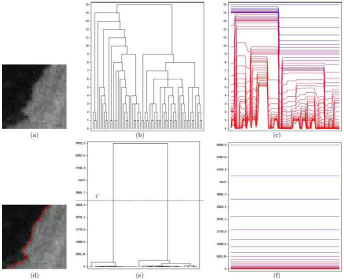
- Segmentation tree [Horowitz & Pavlidis, JACM 1976];
- Hierarchical stepwise optimisation [Beaulieu & Goldberg, PAMI 1986];
- Shortest spanning tree segmentation [Morris et al., IEE Proc. 1986];
- Pyramid of region adjacency graphs [Montanvert et al. 1991];
- Graph weighted hierarchy [Kropatsch & Haximusa, SPIE-5299 2004];
- Scale-sets: cuts minimising an energy based on complexity and distortion measures [Guigues *et al*, IJCV 2006];
- Constrained connectivity [Soille, PAMI 2008].



e) Internal and External contrast.

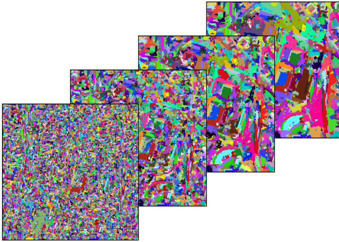
Some existing methods [**not** based on watersheds]

- Segmentation tree [Horowitz & Pavlidis, JACM 1976];
- Hierarchical stepwise optimisation [Beaulieu & Goldberg, PAMI 1986];
- Shortest spanning tree segmentation [Morris et al., IEE Proc. 1986];
- Pyramid of region adjacency graphs [Montanvert et al. 1991];
- Graph weighted hierarchy [Kropatsch & Haximusa, SPIE-5299 2004];
- Scale-sets: cuts minimising an energy based on complexity and distortion measures [Guigues *et al*, IJCV 2006];
- Constrained connectivity [Soille, PAMI 2008].



Some existing methods [**not** based on watersheds]

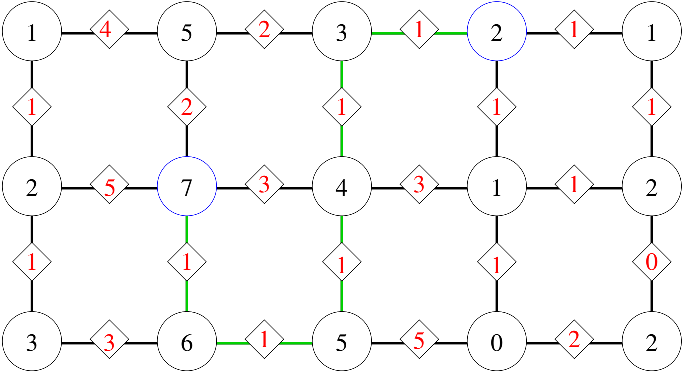
- Segmentation tree [Horowitz & Pavlidis, JACM 1976];
- Hierarchical stepwise optimisation [Beaulieu & Goldberg, PAMI 1986];
- Shortest spanning tree segmentation [Morris et al., IEE Proc. 1986];
- Pyramid of region adjacency graphs [Montanvert et al. 1991];
- Graph weighted hierarchy [Kropatsch & Haximusa, SPIE-5299 2004];
- Scale-sets: cuts minimising an energy based on complexity and distortion measures [Guigues *et al*, IJCV 2006];
- Constrained connectivity [Soille, PAMI 2008].



α-connectivity (introduction)

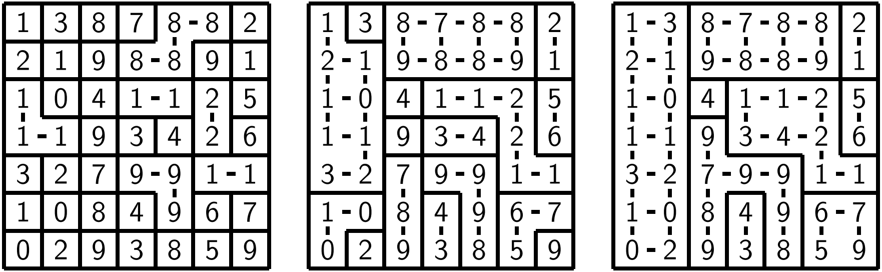
Graph $G = (V, E)$, f for the node weights, F for the edge weights.

Example with $F(\{p, q\} \in E) = \mid f(p) - f(q) \mid$:



α-connectivity

example with $F(\{p,q\} \in E) = | f(p) - f(q) |$



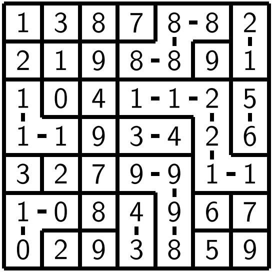
$\alpha = 0$ $\alpha = 1$ $\alpha = 2$

$\alpha\text{-CC}(p) = \{p\} \cup \Big\{ q \,|$ there exists a path $\langle p = p_1, \ldots, p_n = q \rangle$, such that $F(\{p_i, p_{i+1}\}) \leq \alpha$ for all $1 \leq i < n \Big\}$.

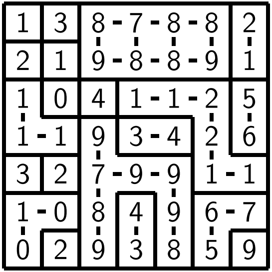
$d_A(p,q) = \min\{\alpha \mid p$ and q belong to the same $\alpha\text{-CC}\}$ is an **ultrametric**.

Constrained connectivity with global range constraint: (α, ω)-connectivity [Soille, PAMI 2008]

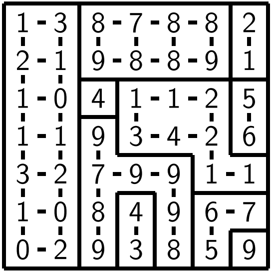
$$(\alpha, \omega)\text{-CC}(p) = \max\left\{\alpha_i\text{-CC}(p) \mid \alpha_i \leq \alpha \text{ and } \mathrm{R}\Big(\alpha_i\text{-CC}(p)\Big) \leq \omega\right\}$$



$\alpha = \omega = 1$



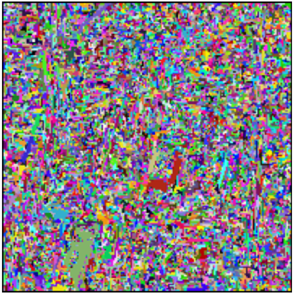
$\alpha = \omega = 2$



$\alpha = \omega = 3$

$d_\Omega(p, q) = \min\{\mathrm{R}(\alpha\text{-CC}(p)) \mid q \in \alpha\text{-CC}(p)\}$ is an ultrametric.

(α, ω)-CC



Main claim

Important idea

- *Any hierarchical segmentation scheme can be seen as a watershed-based hierarchical scheme.*

Main claim

Important idea

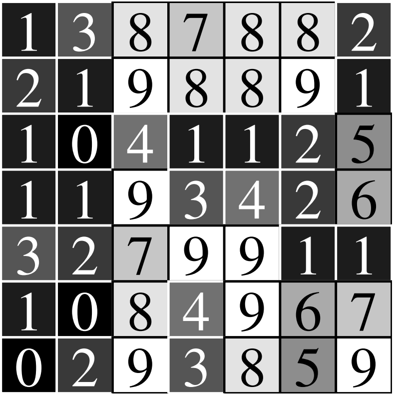
- *Any hierarchical segmentation scheme can be seen as a watershed-based hierarchical scheme.*
- *The trick is to consider edge-weighted graphs instead of node-weighted graphs.*

Main claim

Important idea

- *Any hierarchical segmentation scheme can be seen as a watershed-based hierarchical scheme.*
- *The trick is to consider edge-weighted graphs instead of node-weighted graphs.*
- *Thanks to Olena, any algorithm is easily translated from node-weighted graphs to edge-weighted graphs.*

Looking at edges



(a) Original image

Looking at edges

Doubling the graph (I like to split hairs … and pixels)

1	3	8	7	8	8	2
2	1	9	8	8	9	1
1	0	4	1	1	2	5
1	1	9	3	4	2	6
3	2	7	9	9	1	1
1	0	8	4	9	6	7
0	2	9	3	8	5	9

(a) Original image

1	1	3	3	8	8	7	7	8	8	8	8	2	2
1	1	3	3	8	8	7	7	8	8	8	8	2	2
2	2	1	1	9	9	8	8	8	8	9	9	1	1
2	2	1	1	9	9	8	8	8	8	9	9	1	1
1	1	0	0	4	4	1	1	1	1	2	2	5	5
1	1	0	0	4	4	1	1	1	1	2	2	5	5
1	1	1	1	9	9	3	3	4	4	2	2	6	6
1	1	1	1	9	9	3	3	4	4	2	2	6	6
3	3	2	2	7	7	9	9	9	9	1	1	1	1
3	3	2	2	7	7	9	9	9	9	1	1	1	1
1	1	0	0	8	8	4	4	9	9	6	6	7	7
1	1	0	0	8	8	4	4	9	9	6	6	7	7
0	0	2	2	9	9	3	3	8	8	5	5	9	9
0	0	2	2	9	9	3	3	8	8	5	5	9	9

(b) Double graph

Flat zones == null gradient

Looking at edges

1	1	3	3	8	8	7	7	8	8	8	8	2	2
1	1	3	3	8	8	7	7	8	8	8	8	2	2
2	2	1	1	9	9	8	8	8	8	9	9	1	1
2	2	1	1	9	9	8	8	8	8	9	9	1	1
1	1	0	0	4	4	1	1	1	1	2	2	5	5
1	1	0	0	4	4	1	1	1	1	2	2	5	5
1	1	1	1	9	9	3	3	4	4	2	2	6	6
1	1	1	1	9	9	3	3	4	4	2	2	6	6
3	3	2	2	7	7	9	9	9	9	1	1	1	1
3	3	2	2	7	7	9	9	9	9	1	1	1	1
1	1	0	0	8	8	4	4	9	9	6	6	7	7
1	1	0	0	8	8	4	4	9	9	6	6	7	7
0	0	2	2	9	9	3	3	8	8	5	5	9	9
0	0	2	2	9	9	3	3	8	8	5	5	9	9

(b) Double graph

Looking at edges

Doubling the graph again (to visualize the gradient)

1	1	3	3	8	8	7	7	8	8	8	8	2	2
1	1	3	3	8	8	7	7	8	8	8	8	2	2
2	2	1	1	9	9	8	8	8	8	9	9	1	1
2	2	1	1	9	9	8	8	8	8	9	9	1	1
1	1	0	0	4	4	1	1	1	1	2	2	5	5
1	1	0	0	4	4	1	1	1	1	2	2	5	5
1	1	1	1	9	9	3	3	4	4	2	2	6	6
1	1	1	1	9	9	3	3	4	4	2	2	6	6
3	3	2	2	7	7	9	9	9	9	1	1	1	1
3	3	2	2	7	7	9	9	9	9	1	1	1	1
1	1	0	0	8	8	4	4	9	9	6	6	7	7
1	1	0	0	8	8	4	4	9	9	6	6	7	7
0	0	2	2	9	9	3	3	8	8	5	5	9	9
0	0	2	2	9	9	3	3	8	8	5	5	9	9

(b) Double graph

0	0	0	2	0	0	0	5	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	6	0	0	0	0
0	0	0	2	0	0	0	5	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	6	0	0	0	0
0	0	0	2	0	0	0	5	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	6	0	0	0	0
1	1	1	2	2	2	2	8	1	1	1	1	1	1	1	1	0	0	0	1	1	1	1	8	1	1	1	1
0	0	0	1	0	0	0	8	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	8	0	0	0	0
0	0	0	1	0	0	0	8	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	8	0	0	0	0
0	0	0	1	0	0	0	8	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	8	0	0	0	0
1	1	1	1	1	1	1	8	5	5	5	7	7	7	7	7	7	7	7	7	7	7	7	8	4	4	4	4
0	0	0	1	0	0	0	4	0	0	0	3	0	0	0	0	0	0	0	1	0	0	0	3	0	0	0	0
0	0	0	1	0	0	0	4	0	0	0	3	0	0	0	0	0	0	0	1	0	0	0	3	0	0	0	0
0	0	0	1	0	0	0	4	0	0	0	3	0	0	0	0	0	0	0	1	0	0	0	3	0	0	0	0
0	0	0	1	1	1	1	8	5	5	5	6	2	2	2	3	3	3	3	3	0	0	0	4	1	1	1	1
0	0	0	0	0	0	0	8	0	0	0	6	0	0	0	1	0	0	0	2	0	0	0	4	0	0	0	0
0	0	0	0	0	0	0	8	0	0	0	6	0	0	0	1	0	0	0	2	0	0	0	4	0	0	0	0
0	0	0	0	0	0	0	8	0	0	0	6	0	0	0	1	0	0	0	2	0	0	0	4	0	0	0	0
2	2	2	2	1	1	1	8	2	2	2	6	6	6	6	6	5	5	5	8	1	1	1	5	5	5	5	5
0	0	0	1	0	0	0	5	0	0	0	2	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	5	0	0	0	2	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	5	0	0	0	2	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0
2	2	2	2	2	2	2	8	1	1	1	5	5	5	5	5	0	0	0	8	5	5	5	6	6	6	6	6
0	0	0	1	0	0	0	8	0	0	0	4	0	0	0	5	0	0	0	3	0	0	0	1	0	0	0	0
0	0	0	1	0	0	0	8	0	0	0	4	0	0	0	5	0	0	0	3	0	0	0	1	0	0	0	0
0	0	0	1	0	0	0	8	0	0	0	4	0	0	0	5	0	0	0	3	0	0	0	1	0	0	0	0
1	1	1	2	2	2	2	8	1	1	1	6	1	1	1	5	1	1	1	3	1	1	1	4	2	2	2	2
0	0	0	2	0	0	0	7	0	0	0	6	0	0	0	5	0	0	0	3	0	0	0	4	0	0	0	0
0	0	0	2	0	0	0	7	0	0	0	6	0	0	0	5	0	0	0	3	0	0	0	4	0	0	0	0
0	0	0	2	0	0	0	7	0	0	0	6	0	0	0	5	0	0	0	3	0	0	0	4	0	0	0	0
0	0	0	2	0	0	0	7	0	0	0	6	0	0	0	5	0	0	0	3	0	0	0	4	0	0	0	0

(c) Gradient

Looking at edges

0	0	0	2	0	0	0	5	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	6	0	0	0	0
0	0	0	2	0	0	0	5	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	6	0	0	0	0
0	0	0	2	0	0	0	5	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	6	0	0	0	0
1	1	1	2	2	2	2	8	1	1	1	1	1	1	1	1	0	0	0	1	1	1	1	8	1	1	1	1
0	0	0	1	0	0	0	8	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	8	0	0	0	0
0	0	0	1	0	0	0	8	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	8	0	0	0	0
0	0	0	1	0	0	0	8	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	8	0	0	0	0
1	1	1	1	1	1	1	8	5	5	5	7	7	7	7	7	7	7	7	7	7	7	7	8	4	4	4	4
0	0	0	1	0	0	0	4	0	0	0	3	0	0	0	0	0	0	0	1	0	0	0	3	0	0	0	0
0	0	0	1	0	0	0	4	0	0	0	3	0	0	0	0	0	0	0	1	0	0	0	3	0	0	0	0
0	0	0	1	0	0	0	4	0	0	0	3	0	0	0	0	0	0	0	1	0	0	0	3	0	0	0	0
0	0	0	1	1	1	1	8	5	5	5	6	2	2	2	3	3	3	3	3	0	0	0	4	1	1	1	1
0	0	0	0	0	0	0	8	0	0	0	6	0	0	0	1	0	0	0	2	0	0	0	4	0	0	0	0
0	0	0	0	0	0	0	8	0	0	0	6	0	0	0	1	0	0	0	2	0	0	0	4	0	0	0	0
0	0	0	0	0	0	0	8	0	0	0	6	0	0	0	1	0	0	0	2	0	0	0	4	0	0	0	0
2	2	2	2	1	1	1	8	2	2	2	6	6	6	6	6	5	5	5	8	1	1	1	5	5	5	5	5
0	0	0	1	0	0	0	5	0	0	0	2	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	5	0	0	0	2	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	5	0	0	0	2	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0
2	2	2	2	2	2	2	8	1	1	1	5	5	5	5	5	0	0	0	8	5	5	5	6	6	6	6	6
0	0	0	1	0	0	0	8	0	0	0	4	0	0	0	5	0	0	0	3	0	0	0	1	0	0	0	0
0	0	0	1	0	0	0	8	0	0	0	4	0	0	0	5	0	0	0	3	0	0	0	1	0	0	0	0
0	0	0	1	0	0	0	8	0	0	0	4	0	0	0	5	0	0	0	3	0	0	0	1	0	0	0	0
1	1	1	2	2	2	2	8	1	1	1	6	1	1	1	5	1	1	1	3	1	1	1	4	2	2	2	2
0	0	0	2	0	0	0	7	0	0	0	6	0	0	0	5	0	0	0	3	0	0	0	4	0	0	0	0
0	0	0	2	0	0	0	7	0	0	0	6	0	0	0	5	0	0	0	3	0	0	0	4	0	0	0	0
0	0	0	2	0	0	0	7	0	0	0	6	0	0	0	5	0	0	0	3	0	0	0	4	0	0	0	0
0	0	0	2	0	0	0	7	0	0	0	6	0	0	0	5	0	0	0	3	0	0	0	4	0	0	0	0

(c) Gradient

Looking at edges

Watershed: propagate the pass altitude

0	0	0	2	0	0	0	5	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	6	0	0	0	0
0	0	0	2	0	0	0	5	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	6	0	0	0	0
0	0	0	2	0	0	0	5	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	6	0	0	0	0
1	1	1	2	2	2	2	8	1	1	1	1	1	1	1	1	0	0	0	1	1	1	1	8	1	1	1	1
0	0	0	1	0	0	0	8	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	8	0	0	0	0
0	0	0	1	0	0	0	8	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	8	0	0	0	0
0	0	0	1	0	0	0	8	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	8	0	0	0	0
1	1	1	1	1	1	1	8	5	5	5	7	7	7	7	7	7	7	7	7	7	7	7	8	4	4	4	4
0	0	0	1	0	0	0	4	0	0	0	3	0	0	0	0	0	0	0	1	0	0	0	3	0	0	0	0
0	0	0	1	0	0	0	4	0	0	0	3	0	0	0	0	0	0	0	1	0	0	0	3	0	0	0	0
0	0	0	1	0	0	0	4	0	0	0	3	0	0	0	0	0	0	0	1	0	0	0	3	0	0	0	0
0	0	0	1	1	1	1	8	5	5	5	6	2	2	2	3	3	3	3	3	0	0	0	4	1	1	1	1
0	0	0	0	0	0	0	8	0	0	0	6	0	0	0	1	0	0	0	2	0	0	0	4	0	0	0	0
0	0	0	0	0	0	0	8	0	0	0	6	0	0	0	1	0	0	0	2	0	0	0	4	0	0	0	0
0	0	0	0	0	0	0	8	0	0	0	6	0	0	0	1	0	0	0	2	0	0	0	4	0	0	0	0
2	2	2	2	1	1	1	8	2	2	2	6	6	6	6	6	5	5	5	8	1	1	1	5	5	5	5	5
0	0	0	1	0	0	0	5	0	0	0	2	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	5	0	0	0	2	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	5	0	0	0	2	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0
2	2	2	2	2	2	2	8	1	1	1	5	5	5	5	5	0	0	0	8	5	5	5	6	6	6	6	6
0	0	0	1	0	0	0	8	0	0	0	4	0	0	0	5	0	0	0	3	0	0	0	1	0	0	0	0
0	0	0	1	0	0	0	8	0	0	0	4	0	0	0	5	0	0	0	3	0	0	0	1	0	0	0	0
0	0	0	1	0	0	0	8	0	0	0	4	0	0	0	5	0	0	0	3	0	0	0	1	0	0	0	0
1	1	1	2	2	2	2	8	1	1	1	6	1	1	1	5	1	1	1	3	1	1	1	4	2	2	2	2
0	0	0	2	0	0	0	7	0	0	0	6	0	0	0	5	0	0	0	3	0	0	0	4	0	0	0	0
0	0	0	2	0	0	0	7	0	0	0	6	0	0	0	5	0	0	0	3	0	0	0	4	0	0	0	0
0	0	0	2	0	0	0	7	0	0	0	6	0	0	0	5	0	0	0	3	0	0	0	4	0	0	0	0
0	0	0	2	0	0	0	7	0	0	0	6	0	0	0	5	0	0	0	3	0	0	0	4	0	0	0	0

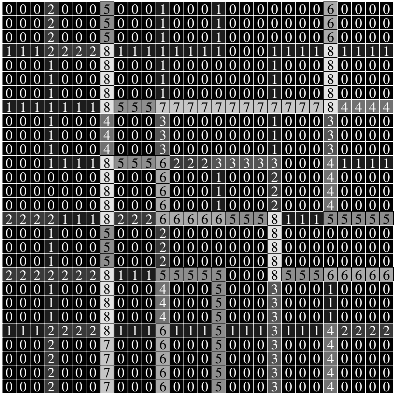
(c) Gradient

0	0	0	2	0	0	0	5	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	5	0	0	0	0
0	0	0	2	0	0	0	5	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	5	0	0	0	0
0	0	0	2	0	0	0	5	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	5	0	0	0	0
1	1	1	2	2	2	2	5	1	1	1	1	1	1	1	1	0	0	0	1	1	1	1	5	1	1	1	1
0	0	0	1	0	0	0	5	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	5	0	0	0	0
0	0	0	1	0	0	0	5	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	5	0	0	0	0
0	0	0	1	0	0	0	5	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	5	0	0	0	0
1	1	1	1	1	1	1	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	4	4	4	4
0	0	0	1	0	0	0	4	0	0	0	3	0	0	0	0	0	0	0	1	0	0	0	3	0	0	0	0
0	0	0	1	0	0	0	4	0	0	0	3	0	0	0	0	0	0	0	1	0	0	0	3	0	0	0	0
0	0	0	1	0	0	0	4	0	0	0	3	0	0	0	0	0	0	0	1	0	0	0	3	0	0	0	0
0	0	0	1	1	1	1	5	5	5	5	5	2	2	2	2	2	2	2	2	0	0	0	3	1	1	1	1
0	0	0	0	0	0	0	5	0	0	0	5	0	0	0	1	0	0	0	2	0	0	0	3	0	0	0	0
0	0	0	0	0	0	0	5	0	0	0	5	0	0	0	1	0	0	0	2	0	0	0	3	0	0	0	0
0	0	0	0	0	0	0	5	0	0	0	5	0	0	0	1	0	0	0	2	0	0	0	3	0	0	0	0
1	1	1	1	1	1	1	5	2	2	2	5	5	5	5	5	5	5	5	5	1	1	1	3	3	3	3	3
0	0	0	1	0	0	0	5	0	0	0	2	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	5	0	0	0	2	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	5	0	0	0	2	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0
2	2	2	2	2	2	2	5	1	1	1	4	4	4	4	4	0	0	0	5	5	5	5	5	5	5	5	5
0	0	0	1	0	0	0	5	0	0	0	4	0	0	0	4	0	0	0	3	0	0	0	1	0	0	0	0
0	0	0	1	0	0	0	5	0	0	0	4	0	0	0	4	0	0	0	3	0	0	0	1	0	0	0	0
0	0	0	1	0	0	0	5	0	0	0	4	0	0	0	4	0	0	0	3	0	0	0	1	0	0	0	0
1	1	1	2	2	2	2	5	1	1	1	4	1	1	1	4	1	1	1	3	1	1	1	2	2	2	2	2
0	0	0	2	0	0	0	5	0	0	0	4	0	0	0	4	0	0	0	3	0	0	0	2	0	0	0	0
0	0	0	2	0	0	0	5	0	0	0	4	0	0	0	4	0	0	0	3	0	0	0	2	0	0	0	0
0	0	0	2	0	0	0	5	0	0	0	4	0	0	0	4	0	0	0	3	0	0	0	2	0	0	0	0
0	0	0	2	0	0	0	5	0	0	0	4	0	0	0	4	0	0	0	3	0	0	0	2	0	0	0	0

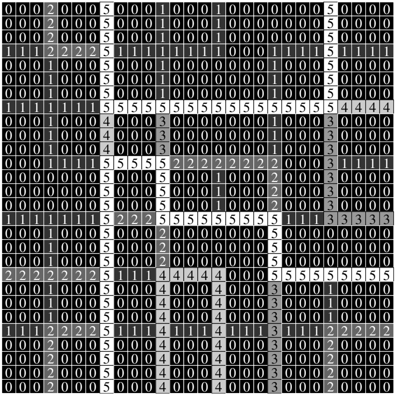
(d) Watershed

Looking at edges

the watershed is the saliency map of the α-connectivity

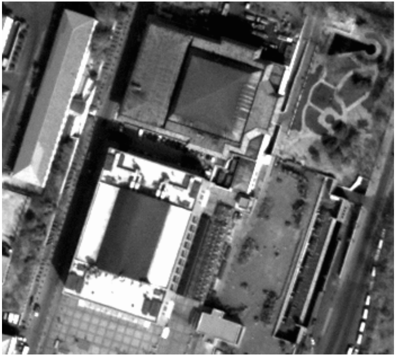


(c) Gradient



(d) Watershed

Application



Original image

α-connectivity saliency map

Main result - a new class of watersheds: ultrametric watersheds

Theorem

- *Saliency maps can be characterized as ultrametric watersheds*
- *Ultrametric watersheds have a computable definition*
- *There exists a bijection between the set of ultrametric watersheds and the set of hierarchical segmentations.*
- *[Najman, ISMM 2009, JMIV 2011]*

Ultrametric watersheds: formal definitions

- If $S \subset E$, $\overline{S} = E \setminus S$.
- $F[\lambda] = \{v \in E \mid F(v) \leq \lambda\}$.
- An edge $u \in \overline{E(X)}$ is said to be *W-simple (for X)* if X has the same number of connected components as $X + u$.
- An edge u such that $F(u) = \lambda$ is said to be *W-destructible (for F) with lowest value λ_0* if there exists λ_0 such that, for all λ_1, $\lambda_0 < \lambda_1 \leq \lambda$, u is W-simple for $F[\lambda_1]$ and if u is not W-simple for $F[\lambda_0]$.
- A *topological watershed (on G)* is a map that contains no W-destructible edges.
- A map F is an *ultrametric watershed* if F is a topological watershed, and if furthemore, for any minimum X of F, $F(X) = 0$.

Ultrametric watersheds: some properties

The *connection value* is the number $F(x,y) = \min_{\pi \in \Pi(x,y)} \max\{F(u) | u \in \pi\}$, where $\Pi(x,y)$ is the set of all paths linking x to y in G. If X and Y are two subgraphs of G, we set $F(X,Y) = \min\{F(x,y) \mid x \in X, y \in Y\}$.

Theorem

A map F is a topological watershed if and only if:

(i) *Its minima form a segmentation of G;*

(ii) *for any edge $v = \{x,y\}$, if there exist X and Y in $\mathcal{M}(F)$, $X \neq Y$, such that $x \in V(X)$ and $y \in V(Y)$, then $F(v) = F(X,Y)$.*

Property

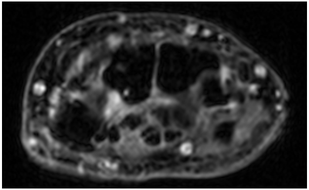
A map F is an ultrametric watershed if and only if for all $\lambda \geq 0$, $F[\lambda]$ is a segmentation of G.

Illustration of main theorem

Novel potential methodology

Illustration of main theorem

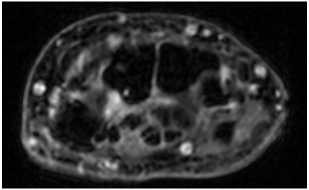
Novel potential methodology



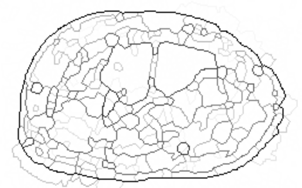
(a) Original image

Illustration of main theorem

Novel potential methodology



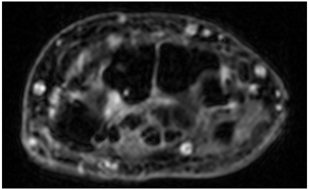
(a) Original image



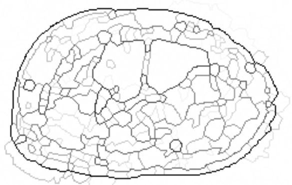
(b) Ultrametric watershed

Illustration of main theorem

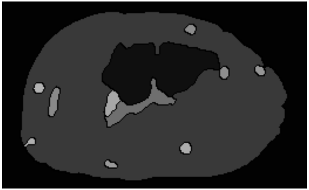
Novel potential methodology



(a) Original image



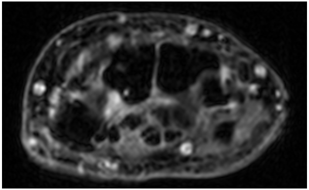
(b) Ultrametric watershed



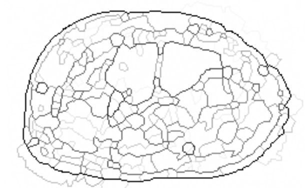
(c) One of the segmentations

Illustration of main theorem

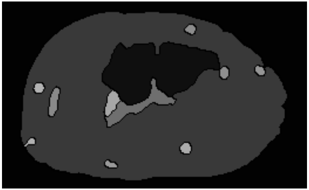
Novel potential methodology



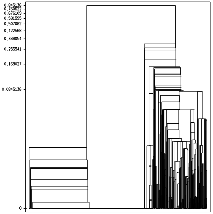
(a) Original image



(b) Ultrametric watershed



(c) One of the segmentations



(d) Dendrogram

Illustration of main theorem

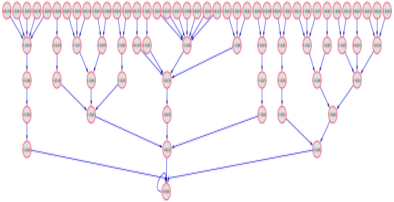
Result

Main theorem: the dendrogram can be replaced by an ultrametric watershed

Looking at min-tree of edge-maps

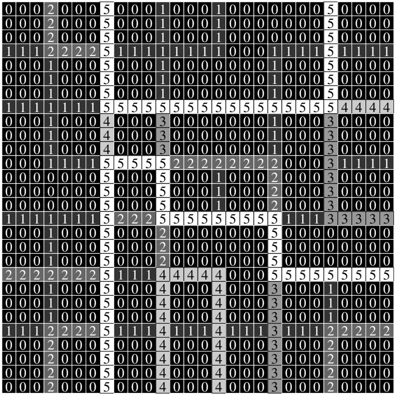
The min-tree of a saliency map : the connected components of all the thresholds of a saliency map

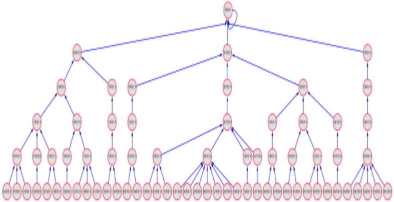
0	0	0	2	0	0	0	5	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	5	0	0	0	0
0	0	0	2	0	0	0	5	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	5	0	0	0	0
0	0	0	2	0	0	0	5	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	5	0	0	0	0
1	1	1	2	2	2	2	5	1	1	1	1	1	1	1	1	0	0	0	1	1	1	1	5	1	1	1	1
0	0	0	1	0	0	0	5	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	5	0	0	0	0
0	0	0	1	0	0	0	5	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	5	0	0	0	0
0	0	0	1	0	0	0	5	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	5	0	0	0	0
1	1	1	1	1	1	1	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	4	4	4	4
0	0	0	1	0	0	0	4	0	0	0	3	0	0	0	0	0	0	0	1	0	0	0	3	0	0	0	0
0	0	0	1	0	0	0	4	0	0	0	3	0	0	0	0	0	0	0	1	0	0	0	3	0	0	0	0
0	0	0	1	0	0	0	4	0	0	0	3	0	0	0	0	0	0	0	1	0	0	0	3	0	0	0	0
0	0	0	1	1	1	1	5	5	5	5	5	2	2	2	2	2	2	2	2	0	0	0	3	1	1	1	1
0	0	0	0	0	0	0	5	0	0	0	5	0	0	0	1	0	0	0	2	0	0	0	3	0	0	0	0
0	0	0	0	0	0	0	5	0	0	0	5	0	0	0	1	0	0	0	2	0	0	0	3	0	0	0	0
0	0	0	0	0	0	0	5	0	0	0	5	0	0	0	1	0	0	0	2	0	0	0	3	0	0	0	0
1	1	1	1	1	1	1	5	2	2	2	5	5	5	5	5	5	5	5	5	1	1	1	3	3	3	3	3
0	0	0	1	0	0	0	5	0	0	0	2	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	5	0	0	0	2	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	5	0	0	0	2	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0
2	2	2	2	2	2	2	5	1	1	1	4	4	4	4	4	0	0	0	5	5	5	5	5	5	5	5	5
0	0	0	1	0	0	0	5	0	0	0	4	0	0	0	4	0	0	0	3	0	0	0	1	0	0	0	0
0	0	0	1	0	0	0	5	0	0	0	4	0	0	0	4	0	0	0	3	0	0	0	1	0	0	0	0
0	0	0	1	0	0	0	5	0	0	0	4	0	0	0	4	0	0	0	3	0	0	0	1	0	0	0	0
1	1	1	2	2	2	2	5	1	1	1	4	1	1	1	4	1	1	1	3	1	1	1	2	2	2	2	2
0	0	0	2	0	0	0	5	0	0	0	4	0	0	0	4	0	0	0	3	0	0	0	2	0	0	0	0
0	0	0	2	0	0	0	5	0	0	0	4	0	0	0	4	0	0	0	3	0	0	0	2	0	0	0	0
0	0	0	2	0	0	0	5	0	0	0	4	0	0	0	4	0	0	0	3	0	0	0	2	0	0	0	0
0	0	0	2	0	0	0	5	0	0	0	4	0	0	0	4	0	0	0	3	0	0	0	2	0	0	0	0



Looking at min-tree of edge-maps

It is a dendrogram (with more information)





Constrained connectivity as a flooding

- The range constraint is increasing on the min-tree of the gradient

Constrained connectivity as a flooding

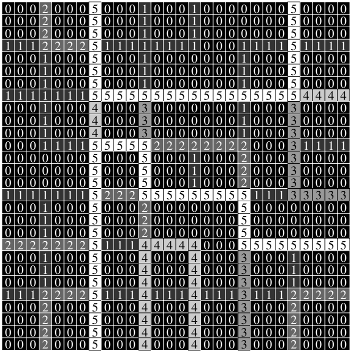
- The range constraint is increasing on the min-tree of the gradient
- The range constraint is a flooding criterion

Constrained connectivity as a flooding

- The range constraint is increasing on the min-tree of the gradient
- The range constraint is a flooding criterion
- By flooding a watershed of the gradient with the range constraint, we obtain the constrained connectivity saliency map

Constrained connectivity as a flooding

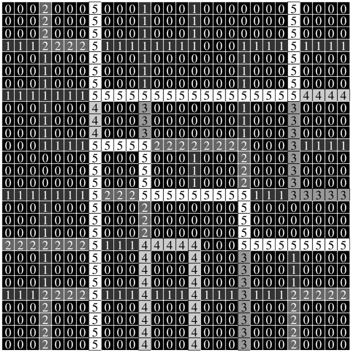
- The range constraint is increasing on the min-tree of the gradient
- The range constraint is a flooding criterion
- By flooding a watershed of the gradient with the range constraint, we obtain the constrained connectivity saliency map



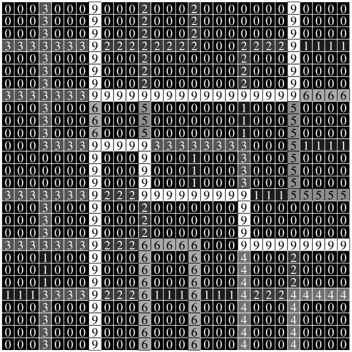
α-connectivity

Constrained connectivity as a flooding

- The range constraint is increasing on the min-tree of the gradient
- The range constraint is a flooding criterion
- By flooding a watershed of the gradient with the range constraint, we obtain the constrained connectivity saliency map



α-connectivity



Constrained connectivity

Application

α-connectivity

Constrained connectivity

Conclusion

Important idea

- *Different (hierarchical) image representations*
- *Choose what is best adapted to the problem at hand*
- *Change the space of image representations*
- *Use Olena to alleviate the burden of programmation*

Conclusion

Important idea

- *Different (hierarchical) image representations*
- *Choose what is best adapted to the problem at hand*
- *Change the space of image representations*
- *Use Olena to alleviate the burden of programmation*

- **Component tree / dendrogram** == to know **when** regions are merging

Conclusion

Important idea

- *Different (hierarchical) image representations*
- *Choose what is best adapted to the problem at hand*
- *Change the space of image representations*
- *Use Olena to alleviate the burden of programmation*

- **Component tree / dendrogram** == to know **when** regions are merging
- **Minimum spanning tree / Binary partition tree** == to give an **order** in the merging of regions

Conclusion

Important idea

- *Different (hierarchical) image representations*
- *Choose what is best adapted to the problem at hand*
- *Change the space of image representations*
- *Use Olena to alleviate the burden of programmation*

- **Component tree / dendrogram** == to know **when** regions are merging
- **Minimum spanning tree / Binary partition tree** == to give an **order** in the merging of regions
- **Ultrametric watersheds / saliency maps** == to **visualize** the hierarchy and to know the **neighborhood** graph of the regions

Conclusion

Important idea

- *Different (hierarchical) image representations*
- *Choose what is best adapted to the problem at hand*
- *Change the space of image representations*
- *Use Olena to alleviate the burden of programmation*

- **Component tree / dendrogram** == to know **when** regions are merging
- **Minimum spanning tree / Binary partition tree** == to give an **order** in the merging of regions
- **Ultrametric watersheds / saliency maps** == to **visualize** the hierarchy and to know the **neighborhood** graph of the regions
- **Ultrametric watersheds / saliency maps** == to **see** any hierarchy as an **image**