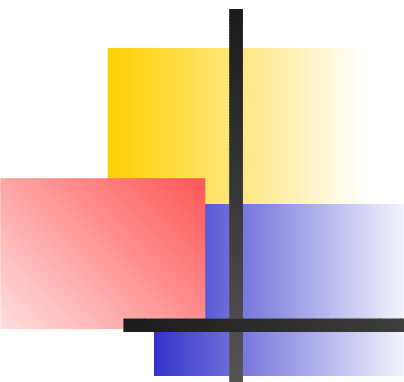




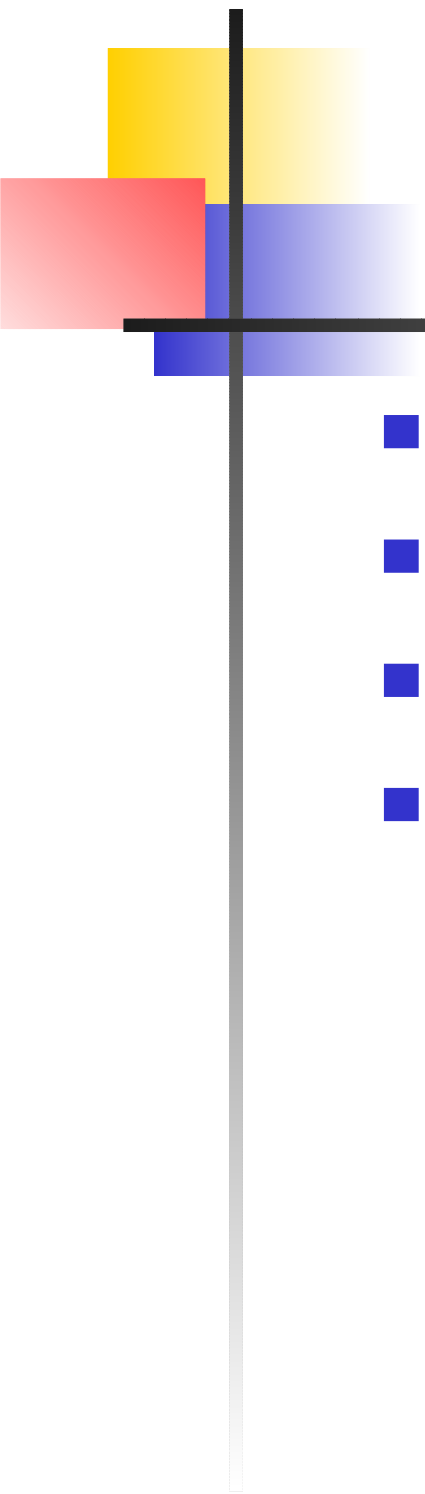
Hierarchical watersheds within the Combinatorial Pyramid framework

Luc Brun, Myriam Mokhtari



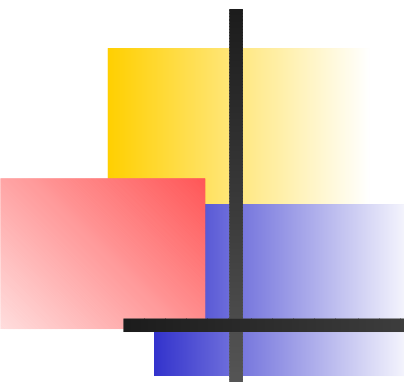
Fernand Meyer





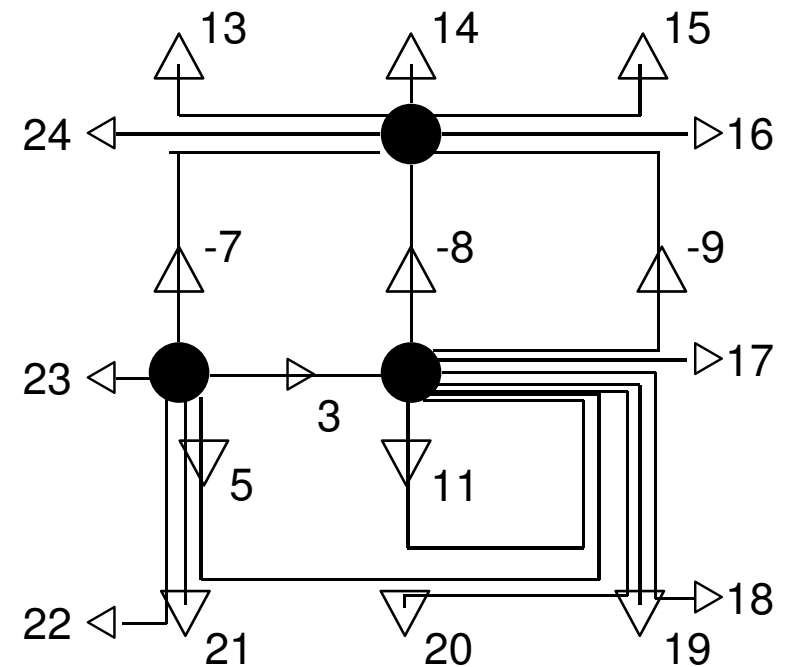
Content

- Combinatorial Pyramids,
- Watersheds,
- Our method,
- Conclusion.

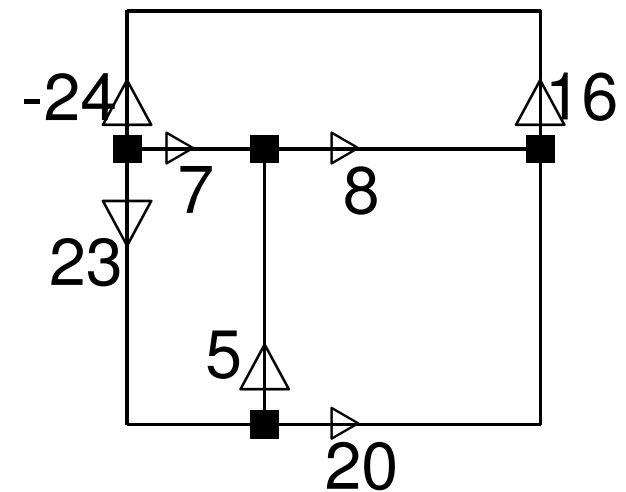
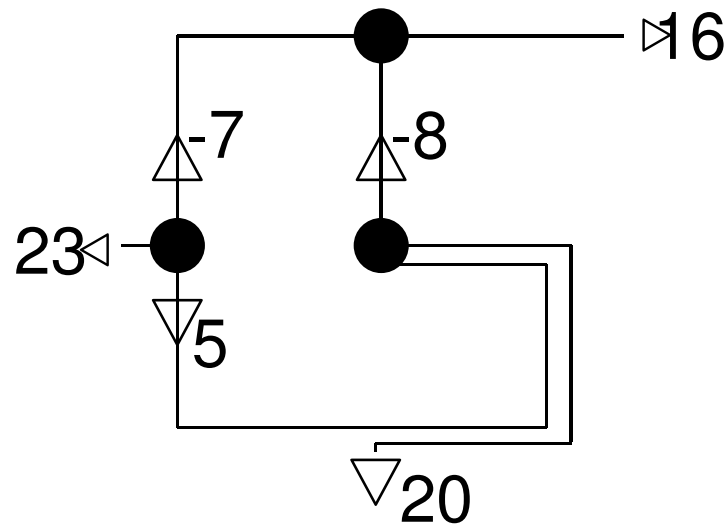


Combinatorial Pyramids

- $\mathcal{P} = (G_0, G_1 \dots, G_n)$: A stack of successively reduced combinatorial maps
 - (c.f. Grasset, Damiand, Lienhardt)
- G_0 encodes the planar sampling grid,
- G_i encodes an image's partition,
- Reduction operations :
 - Merge of regions : Contraction kernels,
 - Removal of redundant edges : Removal kernel.



Embedding



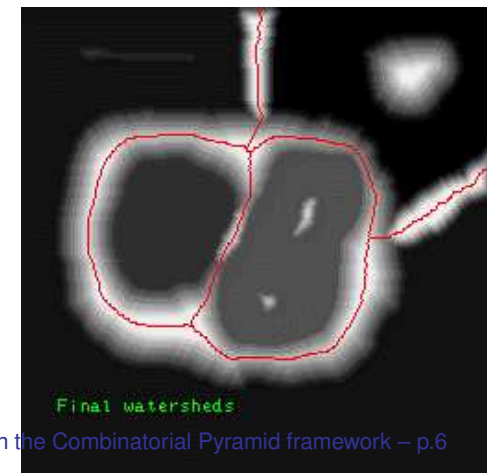
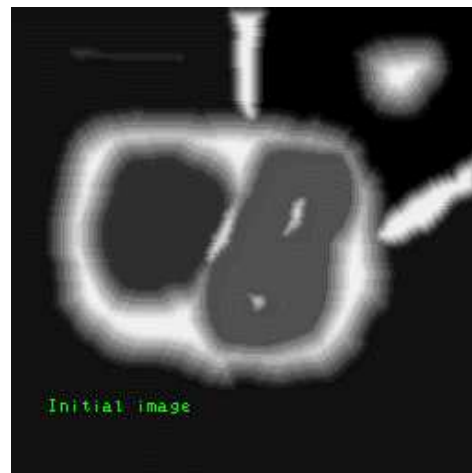
Vertex: Region,

Edge : contour (sequence of lignel),

Dual Vertex : pointel.

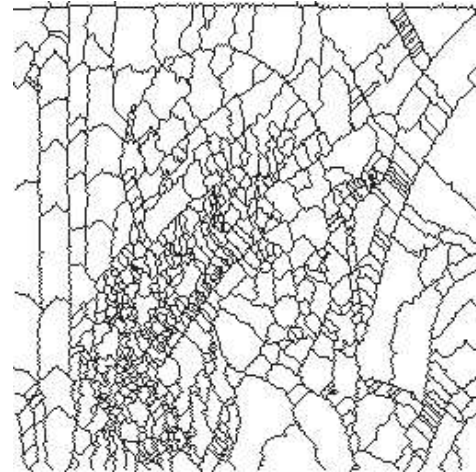
Watersheds

- Decompose the image into Basins and Watersheds
 - Every drop of water falling into a basin will converge to an unique minimum
 - A drop of water falling onto a watershed pixel may converge towards several basins.



Watersheds: Problem

- Over segmentation :

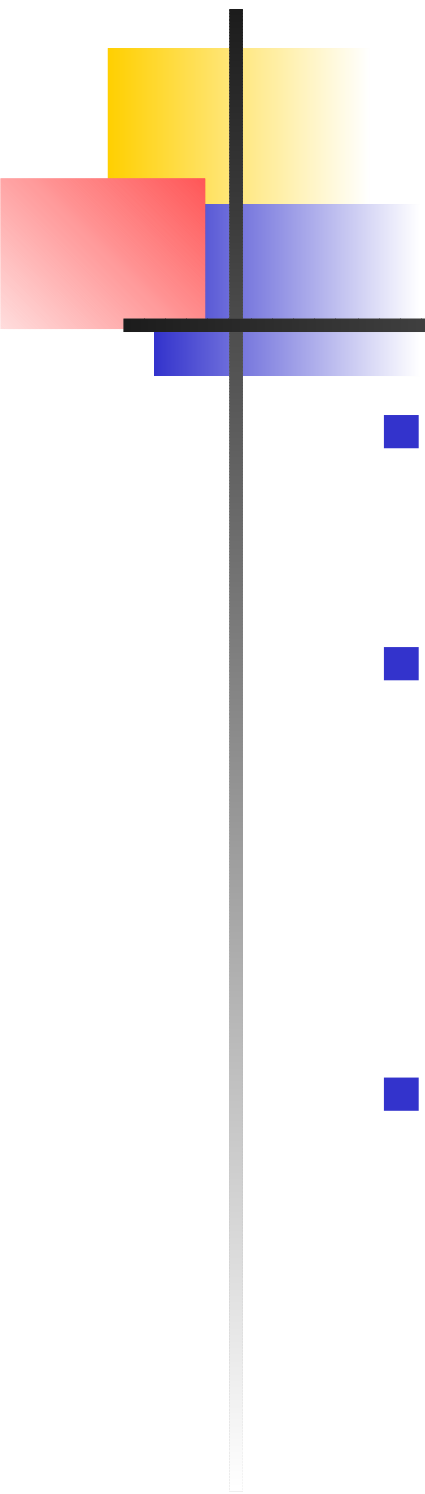


- Need for hierarchical segmentation:



Hierarchical Watersheds : Guideline

1. Get watershed pixels,
 2. Valuate each contour between two basins,
 3. Compute contours dynamic,
 4. Remove contours (merge basins) according to their dynamic.
-
- Hierarchical Watersheds : Hierarchy of basins,
 - Combinatorial Pyramids : Hierarchy of Partitions.

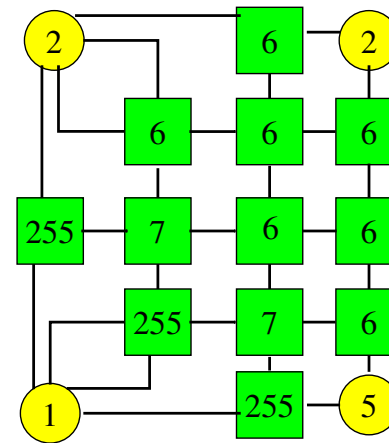


Watershed Transform

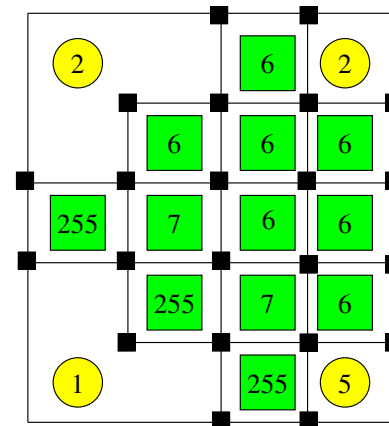
- Initial Combinatorial map :
 - G_0 4-connected sampling grid.
- Build K_0
 - Forest spanning each basin,
 - Built incrementally in parallel with some Watershed algorithms (e.g. Vincent).
- Contract G_0 with $K_0 \rightarrow G_1$.

Example

2	3	6	2
3	6	6	6
255	7	6	6
2	255	7	6
1	2	255	5



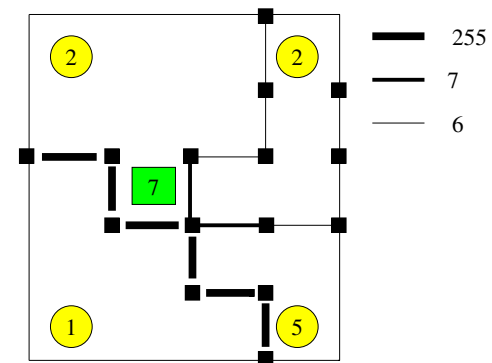
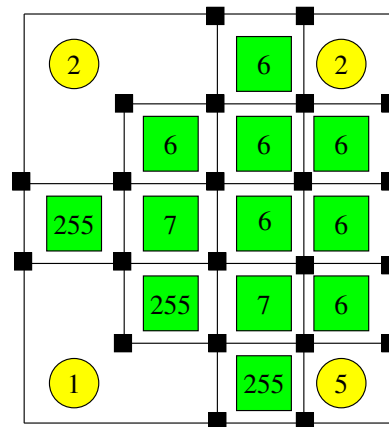
G_1



$\overline{G_1}$

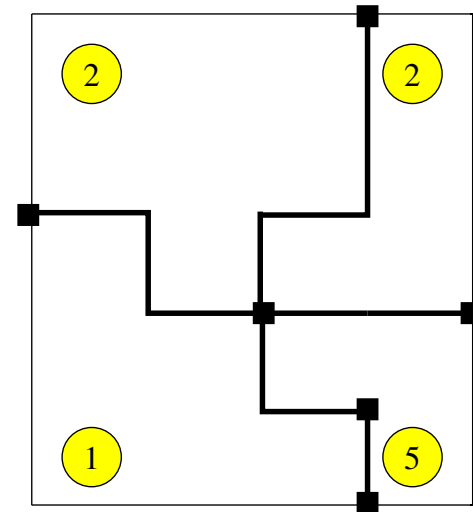
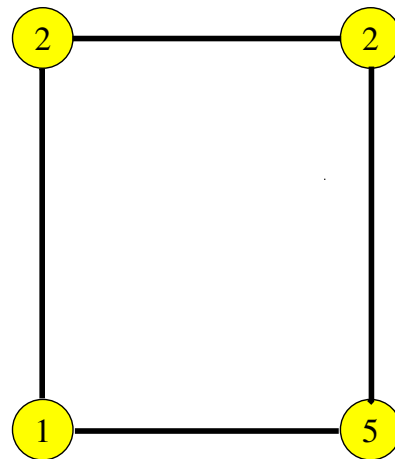
Removal of Watershed pixels

- Flooding process on the watersheds :
 $G_1, K_1 \rightarrow G_2$.
- Lignel's pass value : minimal altitude to climb to connect 2 basins.



Concatenation of boundaries

- Concatenate lignels along boundaries :
 $G_2, K_2 \rightarrow G_3$.



- Each edge corresponds to a sequence of valuated lignels.

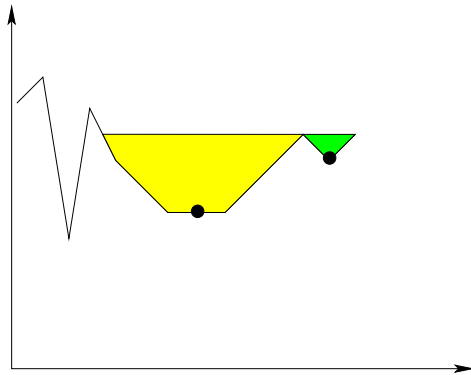
Edge's pass value

- Usual method: minimal value of the lignels.
- What is the « real » minima of a contour.
 - Use both depth and Volume information,
 - Volumic filters (Vachier)



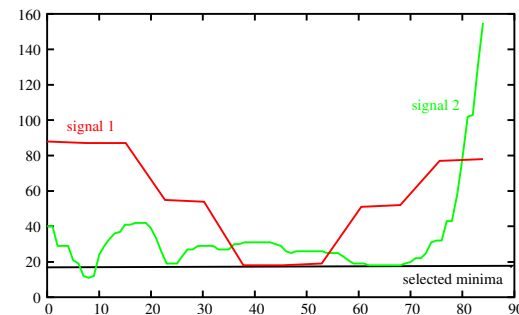
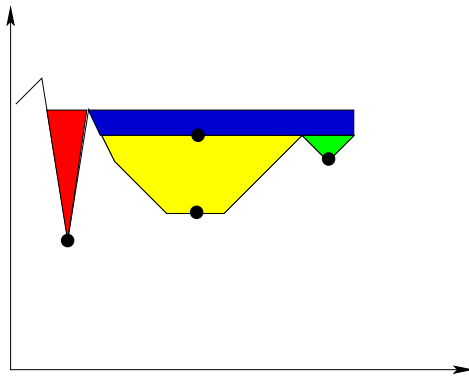
Edge's pass value

- Usual method: minimal value of the lignels.
- What is the « real » minima of a contour.
 - Use both depth and Volume information,
 - Volumic filters (Vachier)



Edge's pass value

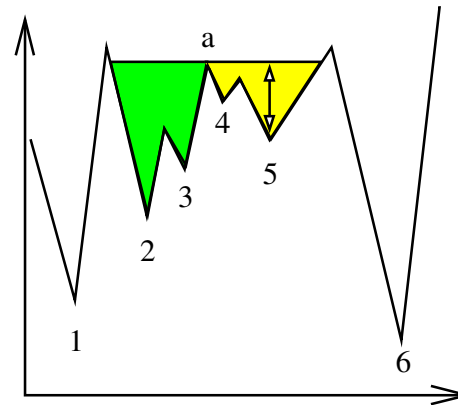
- Usual method: minimal value of the lignels.
- What is the « real » minima of a contour.
 - Use both depth and Volume information,
 - Volumic filters (Vachier)

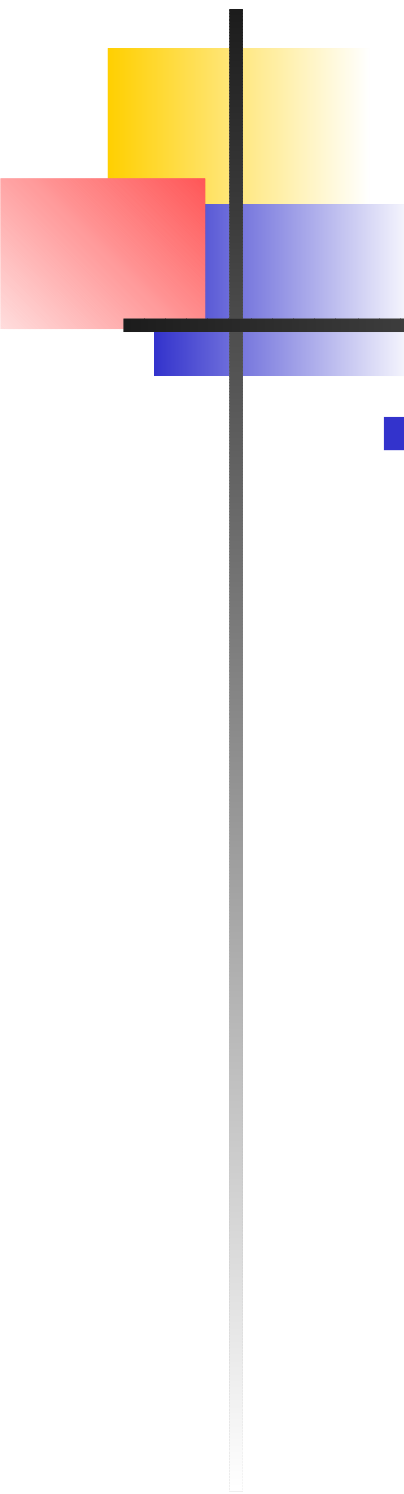


- Minima of the basin with greatest volume.

Edge's dynamic

- Floods progressively the basins
- Edge's dynamic : minimal difference between the edge's pass value and the depth of the two bordering basins.



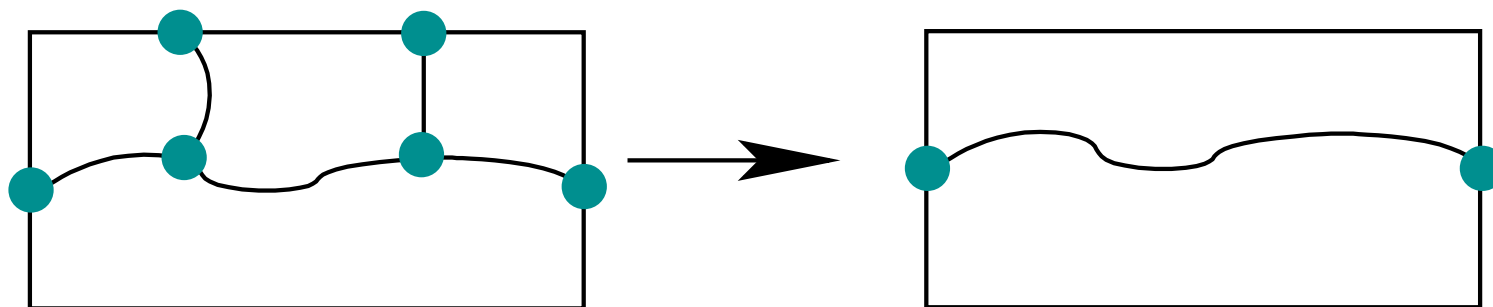


Use of edge's dynamics

- Over Segmentation $\Rightarrow$ small contours ($\lesssim 10$ pixels),

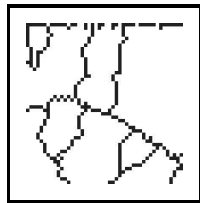
Use of edge's dynamics

- Over Segmentation $\Rightarrow$ small contours ($\lesssim 10$ pixels),
- Contours are not static objects !

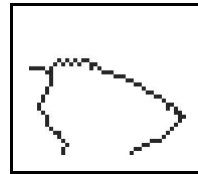


Use of edge's dynamics

- Over Segmentation $\Rightarrow$ small contours ($\lesssim 10$ pixels),
- Contours are not static objects !
- Edge's pass value (& dynamics) become more and more significant as we get up in the hierarchy.



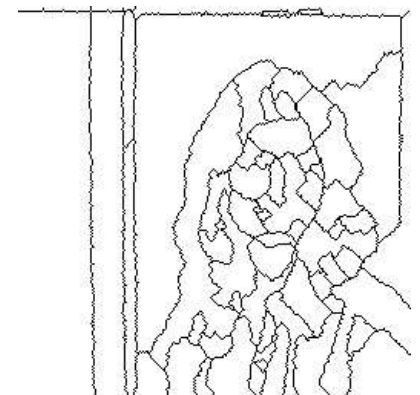
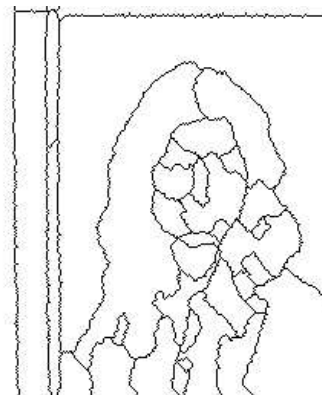
level 10

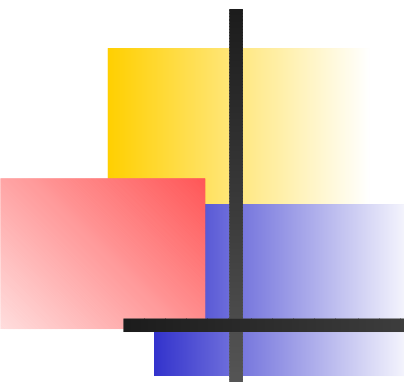


level 15

The algorithm

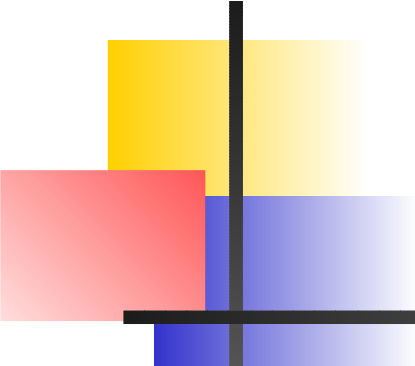
1. Compute edge's pass value. Go to step 3
2. update pass values
3. compute edge's dynamics
4. Contract the edges with the lowest dynamic.
5. If more than one region go to step 2.





Conclusion (1/2)

- Initial partition into a set of basins **and** watershed pixels.
 - Flooding of watershed pixels.
- Valuation of contours by their minimal altitude:
 - Depth of the 1D basin with maximmal altitude.
- Weak reliability of first edge's dynamics
 - Removal of the edge with the lowest dynamic,
 - update of the edge's dynamics at each level



Conclusion (2/2)

Let us return to Jules what belongs to César

- The combinatorial pyramids provide a hierarchical data structure with :
 - An efficient encoding,
 - Adjacency relationships,
 - One edge for each boundary,
 - An efficient traversal of basin's boundaries.