Belo Horizonte December 2014

Directed connected operators:

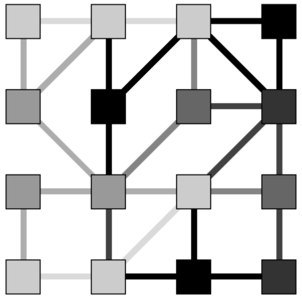
Asymmetric hierarchies for image filtering and segmentation

Benjamin Perret
Benjamin.perret@esiee.fr

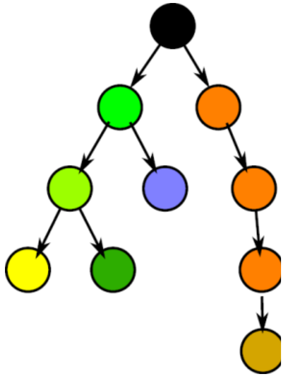
Pitch

Before

- Symmetric adjacency: undirected graph

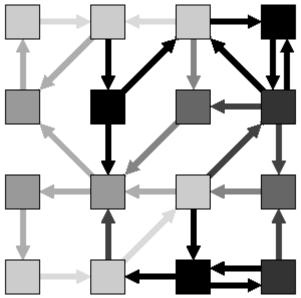


- Hierarchical representation: tree

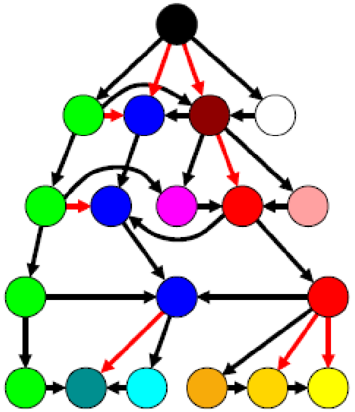


After

- Asymmetric adjacency: directed graph



- Hierarchical representation: acyclic graph

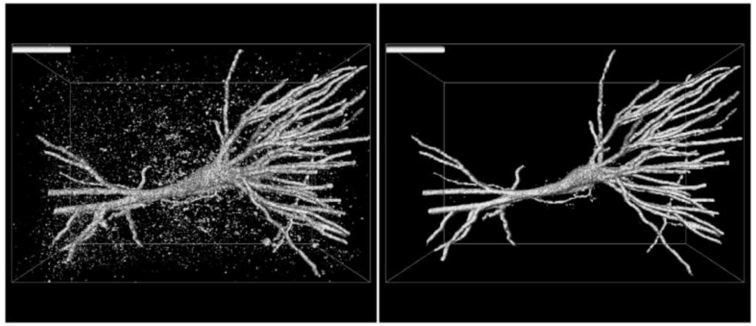


- B. Perret, J. Cousty, O. Tankyevych, H. Talbot, and N. Passat. Directed connected operators: asymmetric hierarchies for image filtering and segmentation. IEEE TPAMI 2014, DOI: 10.1109/TPAMI.2014.2366145

Connected image processing

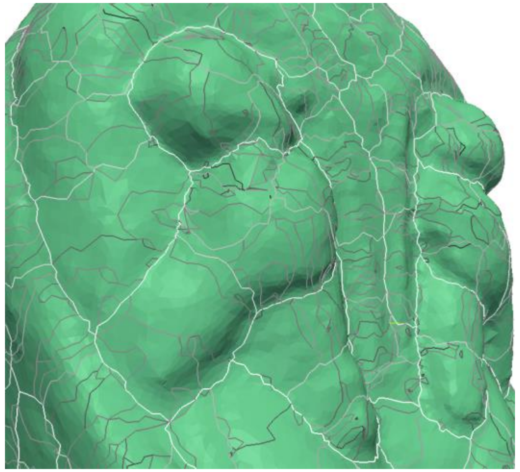
- **A simple paradigm:**
  - Only one legal operation : remove connected components
  - Don’t move contours!
- **Efficient algorithms:**
  - Mostly linear time & space complexity
- **Versatile framework:**
  - Attribute/feature based reasoning
  - Filtering, segmentation, detection, characterization, vision…

Background



- 3D cube filtering : Ouzounis et al, PAMI 2007

- Image simplification: Soille, PAMI 2008



- 3D mesh segmentation: Cousty et al, PAMI 2010

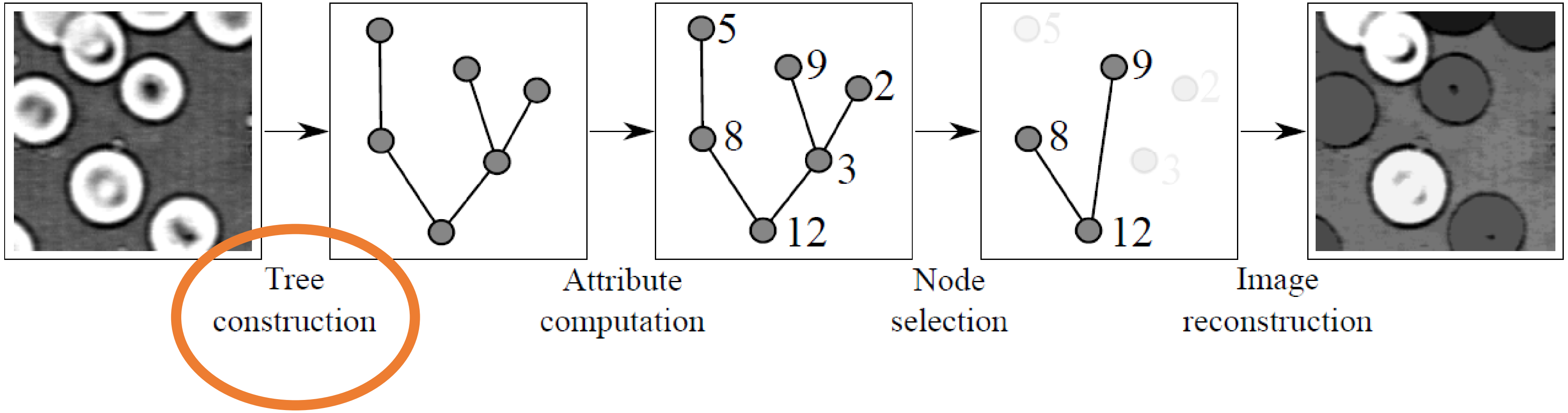
- Interactive segmentation: Passat et al, PR 2010

- Degraded document images restoration: Perret et al, TIP 2012

- Feature detection : Xu et al. TIP 2014

Connected image processing

General 4 steps procedure



ESIEE PARIS

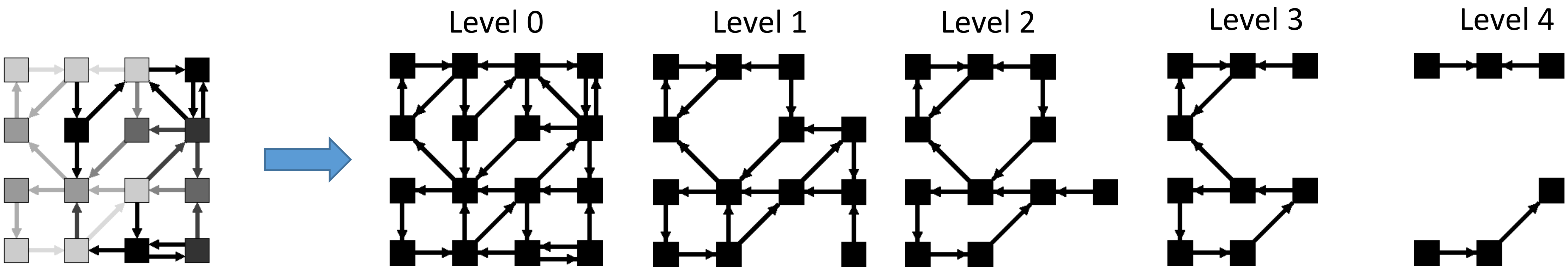
Motivations

- **Success of directed adjacency in other frameworks**
  - **Min-cuts:** Boykov et al IJCV 2006
  - **Random Walkers:** Singaraju et al CVPR 2008
  - **Shortest path forest:** Miranda et al TIP 2014
- **Pros**
  - **Handling of naturally directed information:** k-nearest neighbor
  - **Alleviate the linkage/chaining issue:** *weak* connection
  - **Injection of a priori information:** expert knowledge, learning
- **Cons**
  - **Increased complexity…**

Directed graph

- Let $G = (V, A)$ be a directed graph:
  - V is a finite set of points/vertices/nodes
  - $A \subseteq V \times V$ is the set of arcs/edges

- A node/edge weighted graph => a **stack of graphs**
  - Sequence of nested graphs



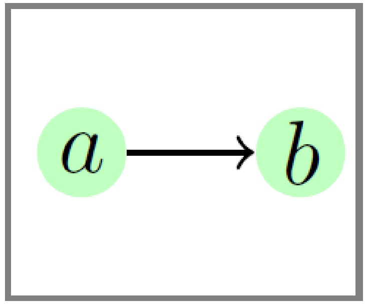
› **Node/edge weighted graph**

› **Stack of graphs**

Directed connected component

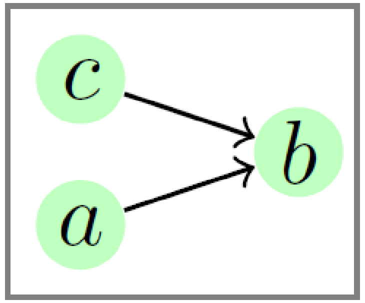
- Let $x \in V$, the **Directed connected component (DCC) of base point** x:
  The set of successors of x denoted $\mathrm{DCC}_G(x)$

A DCC may contain another DCC



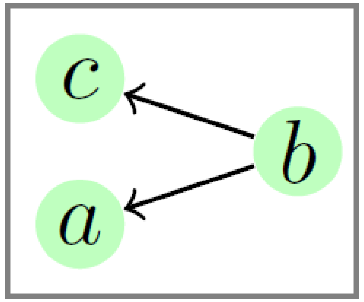
$$\mathrm{DCC}_G(a) = \{a, b\}$$
$$\mathrm{DCC}_G(b) = \{b\}$$

Two DCCs may intersect



$$\mathrm{DCC}_G(a) = \{a, b\}$$
$$\mathrm{DCC}_G(c) = \{c, b\}$$
$$\mathrm{DCC}_G(b) = \{b\}$$

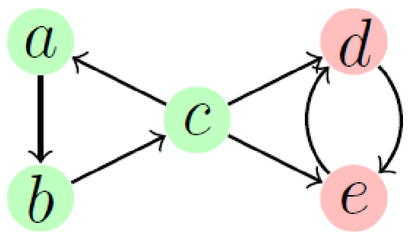
Asymmetric behavior of the DCCs



$$\mathrm{DCC}_G(a) = \{a\}$$
$$\mathrm{DCC}_G(c) = \{c\}$$
$$\mathrm{DCC}_G(b) = \{a, b, c\}$$

Strongly connected component

- A set $X \subseteq V$ is **Strongly connected :** $\forall x, y \in X$
  There is a path from x to y and from y to x in X
- A **Strongly connected component (SCC)** is a strongly connected set of maximal extent
  The SCC that contains the node x is denoted $\mathrm{SCC}_G(x)$

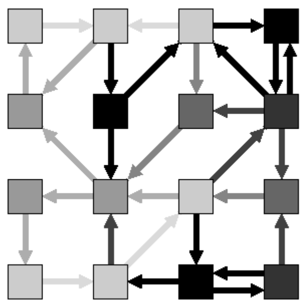


$$\mathrm{SCC}_G(a) = \mathrm{SCC}_G(b) = \mathrm{SCC}_G(c) = \{a, b, c\}$$

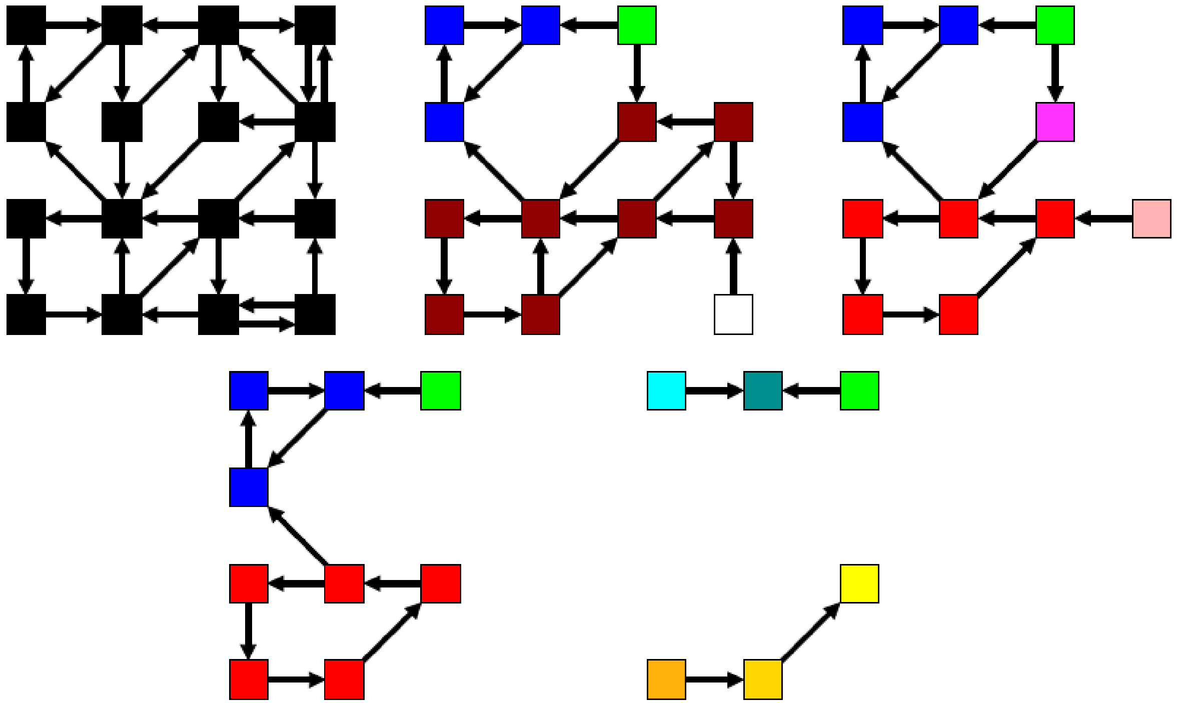
$$\mathrm{SCC}_G(d) = \mathrm{SCC}_G(e) = \{d, e\}$$

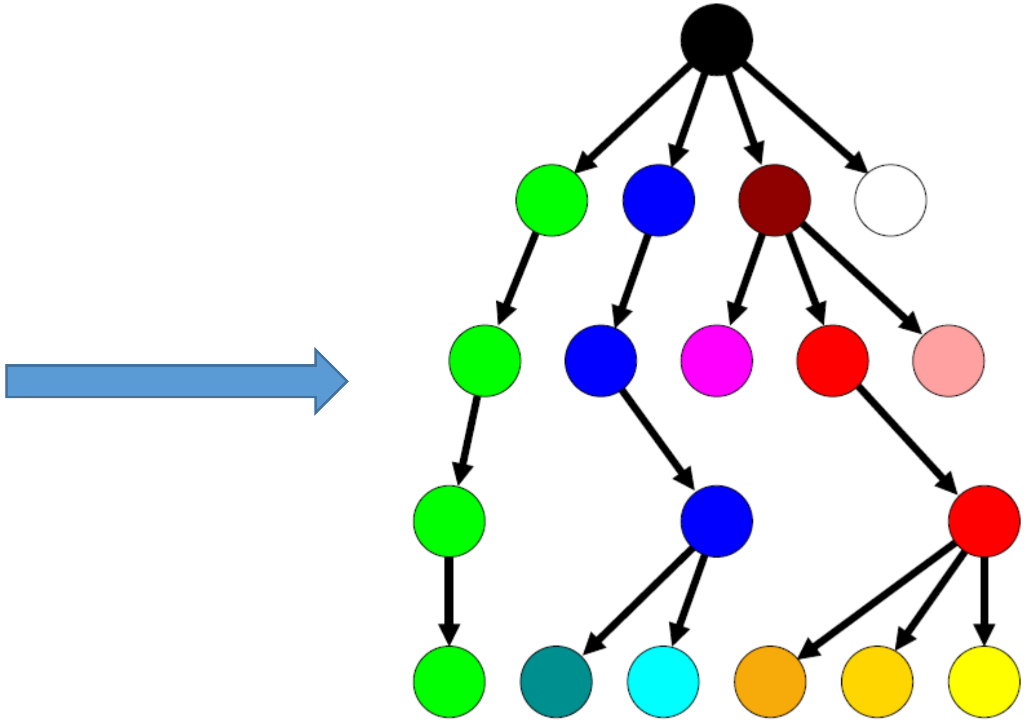
- **The set of SCCs of a graph forms a partition**

Stacking DAG



- **The SCCs of a stack of graphs forms a tree**
  - Generalization of many known hierarchical representation: component tree, quasi-flat zone hierarchy (MST), watershed hierarchy





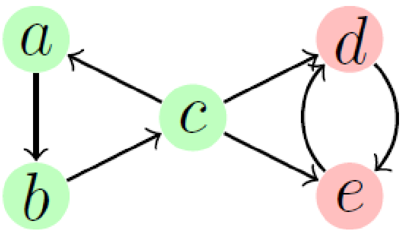
ESIEE PARIS

SCCs and DCCs

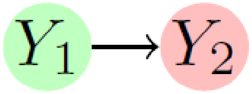
- Relation between DCCs and SCCs

  Two nodes x and y are in the same SCC iff they are the base points of the same DCC: $\mathrm{DCC}_G(x) = \mathrm{DCC}_G(y) \Leftrightarrow \mathrm{SCC}_G(x) = \mathrm{SCC}_G(y)$

- The DAG induced by the SCCs: DAG_G (The SCCs + their adjacency relation) encodes the DCCs



- **DAG of SCCs**



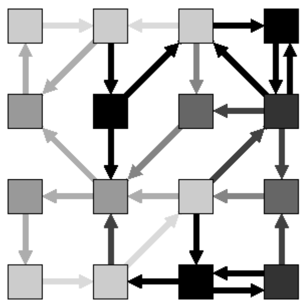
$$\mathrm{SCC}_G(a) = \mathrm{SCC}_G(b) = \mathrm{SCC}_G(c) = Y_1$$

$$\mathrm{SCC}_G(d) = \mathrm{SCC}_G(e) = Y_2$$

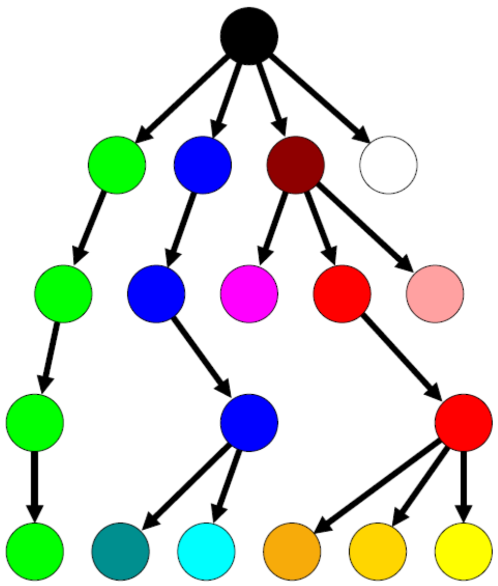
$$\mathrm{DCC}_G(a) = \bigcup \mathrm{DCC}_{\mathrm{DAG}_G}(Y1) = Y_1 \cup Y_2$$

$$\mathrm{DCC}_G(d) = \bigcup \mathrm{DCC}_{\mathrm{DAG}_G}(Y2) = Y_2$$

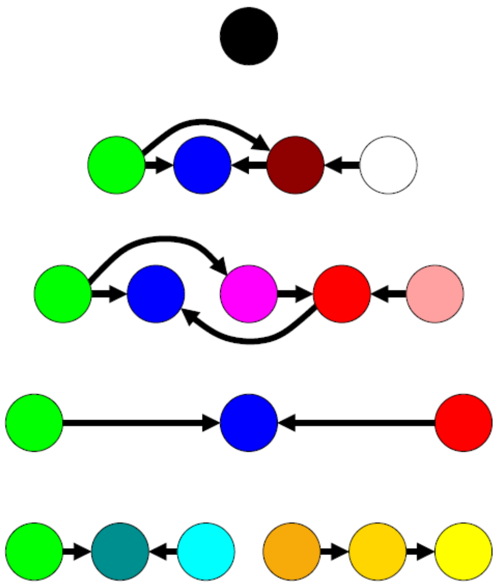
Stacking DAG



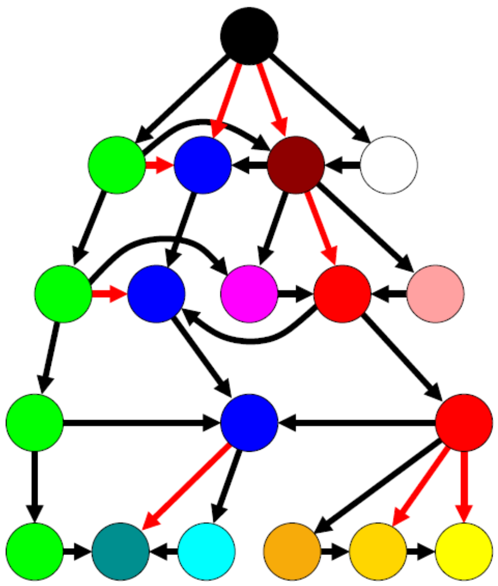
- **The DCCs hierarchy is the combination of**
  - The tree of SCCs: encodes the inter-scale relations
  - The DAGs of SCCs: encode the intra-scale relations
- **All the inter/intra scale relations among the DCCs is encoded**



› SCC tree



› SCC DAGs



› DCC Hierarchy

Algorithm

Naive algorithm

Algorithm 1: D-component hierarchy construction.

Input: $\mathcal{S} = \{\mathfrak{G}_0, \ldots, \mathfrak{G}_\ell\}$, a stack of graphs.
Output: $Label_i$, S-component labeling for each i in $\{0, \ldots, \ell\}$.
Output: Suc_i, array of adjacency lists for each i in $\{0, \ldots, \ell\}$.
Output: PAR_i, parent relation for each i in $\{0, \ldots, \ell\}$.

```
for i from ℓ to 0 do
    Label_i ← S-component labeling(𝔊_i)
    Suc_i ← adjacency lists(𝔊_i, Label_i)
    if i ≠ ℓ then
        PAR_{i+1} ← parent relation(𝔊_{i+1}, Label_{i+1}, Label_i)
```

- **Number of levels** $\mathcal{O}(\ell)$
- **Tarjan algorithm** $\mathcal{O}(|V| + |A|)$
- **Intra-scale adjacency** $\mathcal{O}(|V| + |A|)$
- **Inter-scale adjacency** $\mathcal{O}(|V|)$

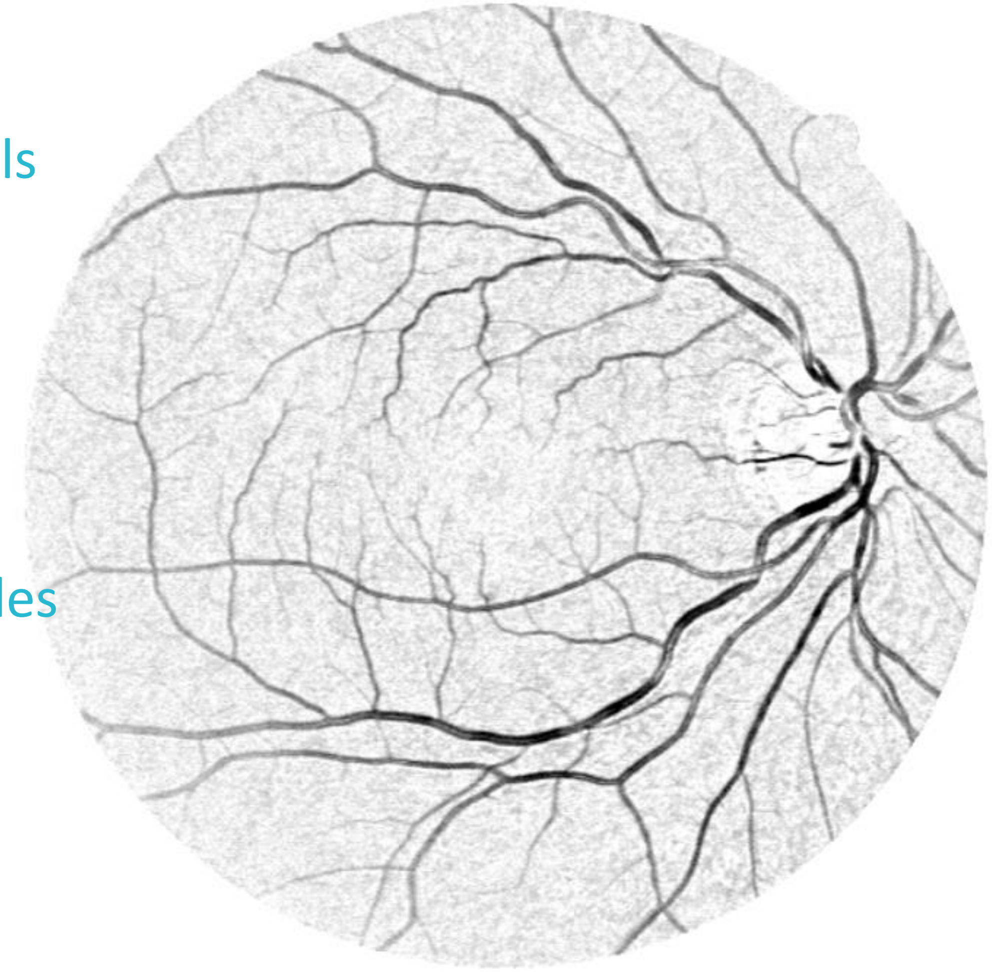
Worst case time complexity $\mathcal{O}(\ell(|V| + |A|))$

Suitable for low depth images : 8 bits

Application – Retinal image

- **Retinal image segmentation**
  - Segmentation and characterization of blood vessels
  - Diagnosis and evolution of several pathologies

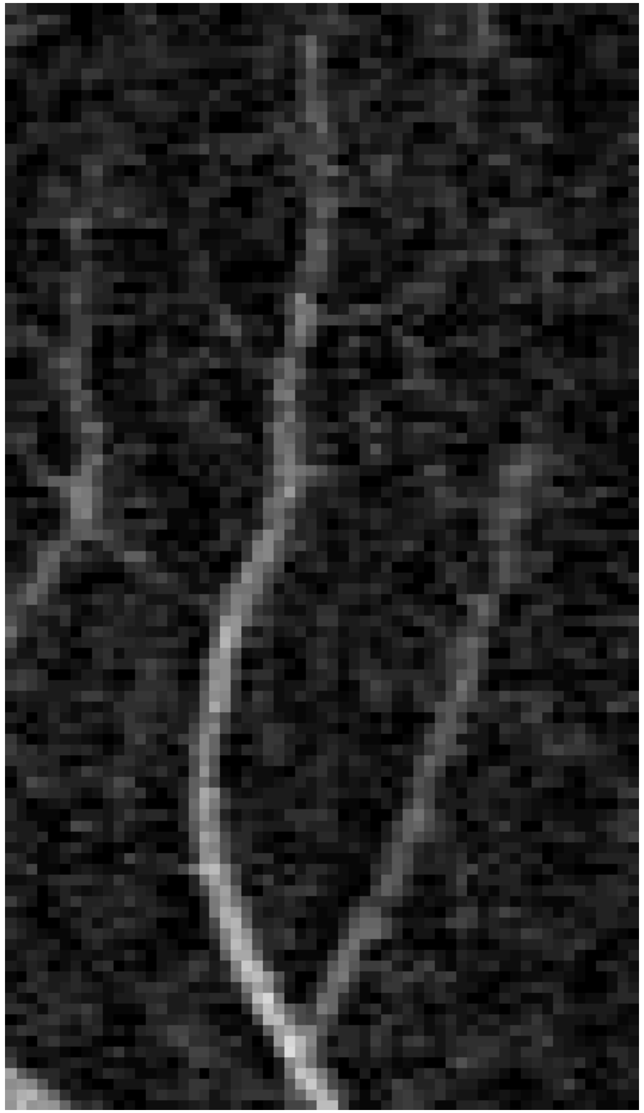
- **Detection of the thin and faint vessels**
  - High level of background noise
  - Appear as disconnected groups of pixel at low scales
  - Disappear at high scales

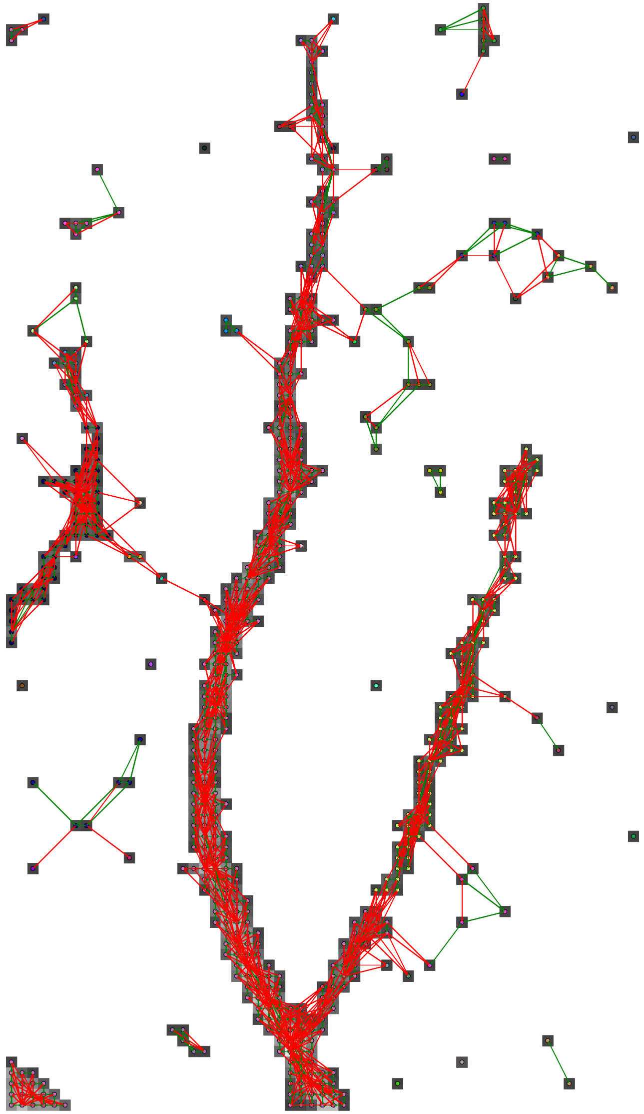


Application – Retinal image

- **Non local directed adjacency**
  - 4 adjacency
  - K-brightest neighbors

- **Allows for:**
  - Jumping over noise
  - Weakly connecting noise





› **Adjacency relation at a critical level**

Application – Retinal image

Filtering criterion

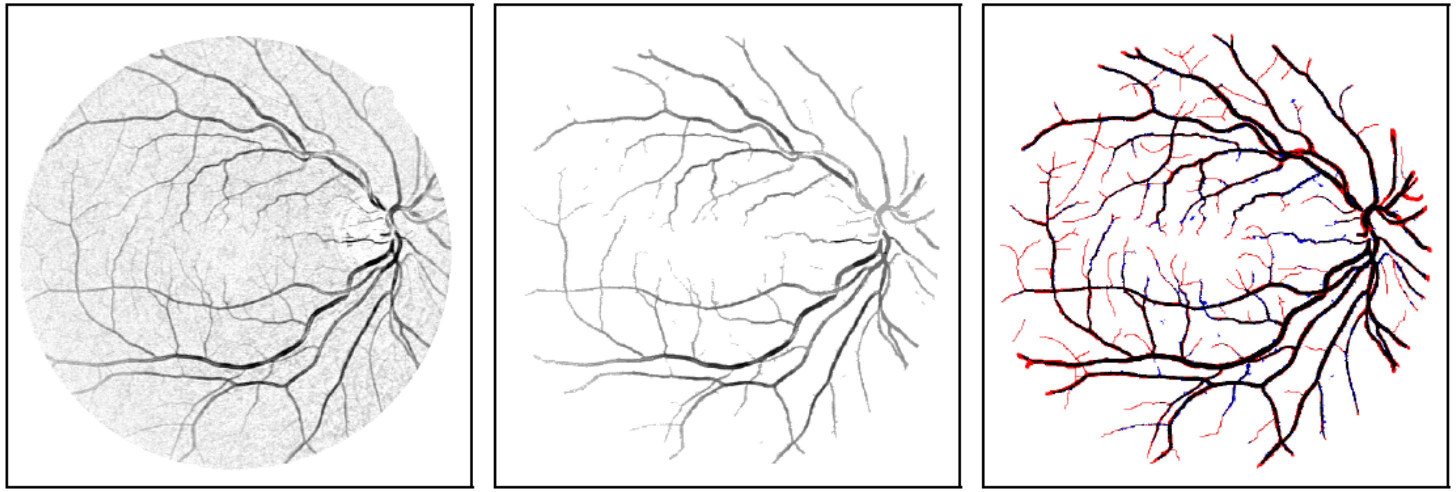
A node of the DCC graph is preserved if

$$\sigma_V(C) = \begin{cases} \text{false} & \text{if } \mathrm{Area}(\mathrm{DCC}_C) \leq t_1 \\ \text{false} & \text{if } t_1 < \mathrm{Area}(\mathrm{DCC}_C) \leq 1000 \\ & \quad \text{and } \mathrm{Elongation}(\mathrm{DCC}_C) > t_2 \\ \text{false} & \text{if } 1000 < \mathrm{Area}(\mathrm{DCC}_C) \\ & \quad \text{and } \mathrm{Hu}_1(\mathrm{DCC}_C) > t_3 \\ \text{true} & \text{otherwise} \end{cases}$$

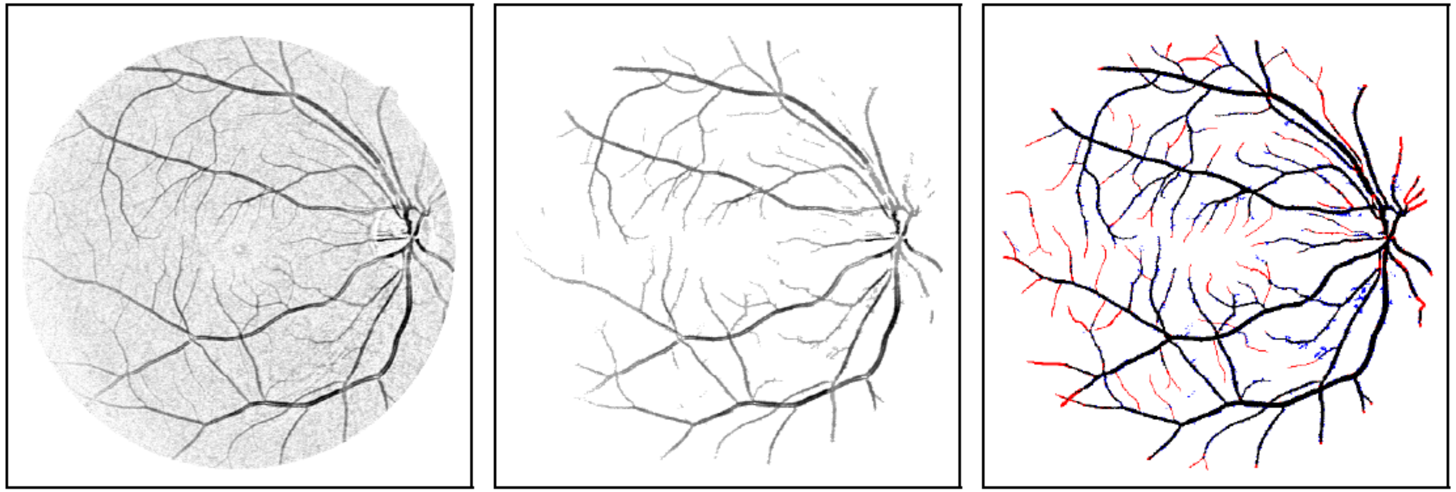
- Very small components are discarded
- Small components must be elongated
- Large components must have a « regular » distribution

Application – Retinal image

- **Examples**



(a) Image 2 of DRIVE (b) Filter $I^{H}_{\sigma_{RV}}$ (c) Segmentation



(d) Image 19 of DRIVE (e) Filter $I^{H}_{\sigma_{RV}}$ (f) Segmentation

Application – Retinal image

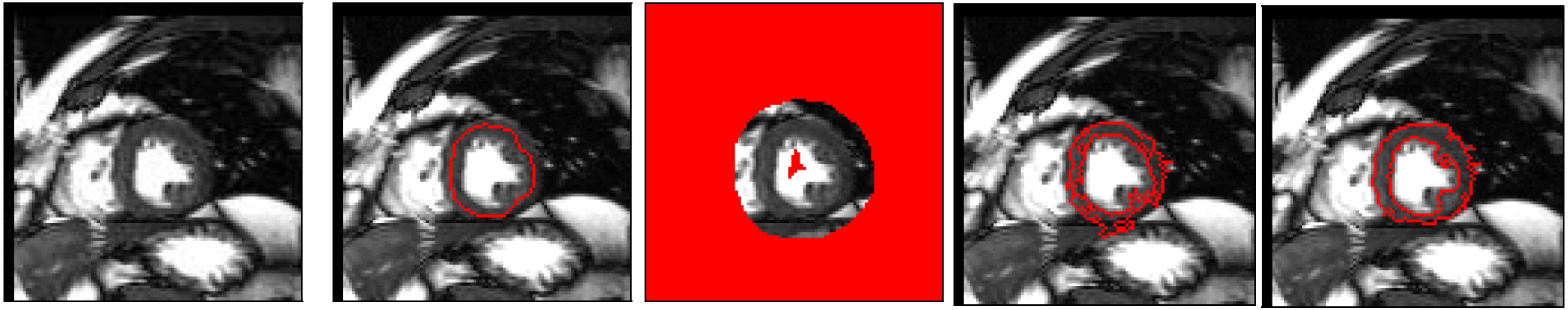
Evaluation on DRIVE database

Method	TPR	TNR	Accuracy (σ)
2nd expert	0.7761	0.9725	0.9473 (0.0048)
RNL D-components (15,0.2,1)	0.7079	0.9790	0.9442 (0.0063)
NL D-components (15,0.2,1)	0.7046	0.9790	0.9439 (0.0064)
NL S-components (30,0.15,1.3)	0.7024	0.9789	0.9434 (0.0070)
NL symmetric (Max) (15,0.2,1.3)	0.6528	0.9828	0.9404 (0.0083)
NL symmetric (Min) (15,0.15,1.35)	0.6980	0.9786	0.9425 (0.0067)
Xu [65] (local symmetric)	0.6924	0.9779	0.9413 (0.0078)
Mendonça [66]	0.7344	0.9764	0.9452 (0.0062)
Staal [64]	0.7193	0.9773	0.9442 (0.0065)

- Non-local directed adjacency with DCC & regularization
- Non-local directed adjacency with DCC
- Non-local directed adjacency with SCC
- Non-local symmetric adjacency 1
- Non-local symmetric adjacency 2
- Local symmetric adjacency (« better » criterion)
- State of the art (learning approach)

Application – Heart segmentation

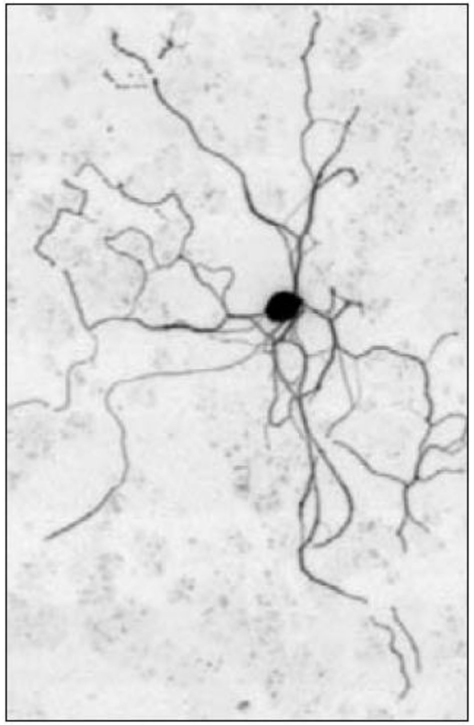
- **Supervised segmentation of the myocardium**
  - Select the largest DCCs that intersect the object marker but not the background marker
- **Directed gradient with modified weights**
  - Indeed equivalent to directed shortest path forest by Miranda et al TIP 2014

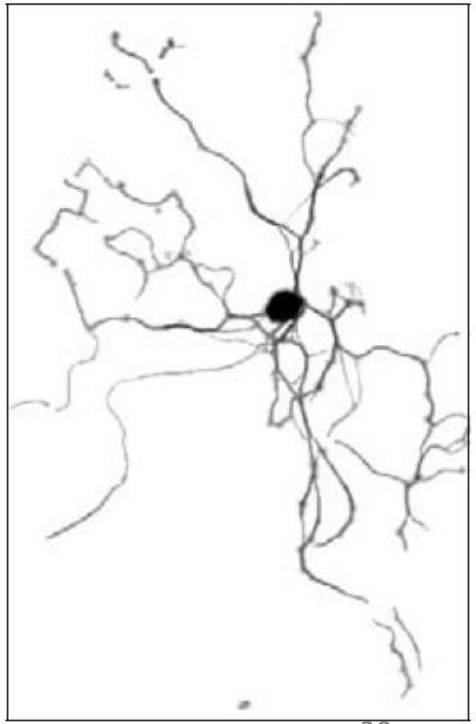


(a) Original (b) O: myocardium (c) B: background (d) Symmetric result (f) Directed result

Application – Neurite filtering

- **Segmentation of neurites in toxicology assays**
- **Directed adjacency constructed from a vesselness classification (tube, blob, background)**
  - Tubes connect to blobs
  - Background connects to tubes and blobs
- **Filtering rule**
  - A tube connect two structures
  - Tubes have thin connections





- Neurite image
- Vesselness classification
- Filtered image

Conclusion

- **Directed component hierarchy**
  - Generalization of many known hierarchical image representations
- **Construction algorithm**
  - Efficient in most cases
  - More efficient algorithms in preparation (support of 64bits images)
- **Demonstrated on several applications**
  - State of the art performance with rather “simple” strategies

Python code online !

http://perso.esiee.fr/~perretb/dc-hierarchy.html