# Structural methods for image and video analysis

## PART I: Content extraction and recognition (in images)

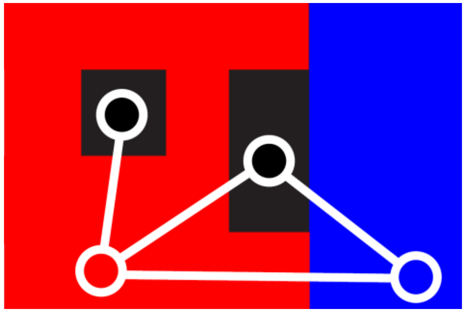

**Jean-Yves Ramel**

**Laboratoire Informatique de Tours - FRANCE**

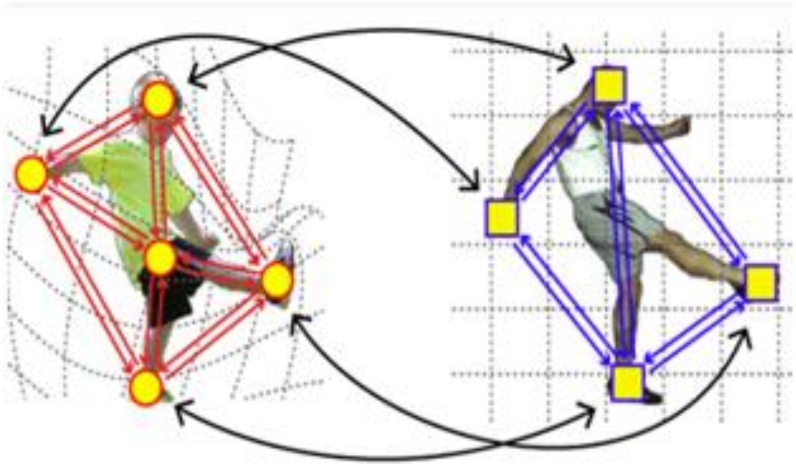
# Outline

- Introduction
  - Terminology and Notations
- Extraction of structural primitives inside images
  - Types of structural primitives
  - From primitives to graphs
- Using graphs for image segmentation
  - Interactive image segmentation
  - Learning and Using a priori information
  - Region spotting with Graphs
- Using graphs for object recognition
  - Exact and Inexact Graph Matching
  - Graph similarity, Graph embedding
  - Learning to match graphs
- Conclusion

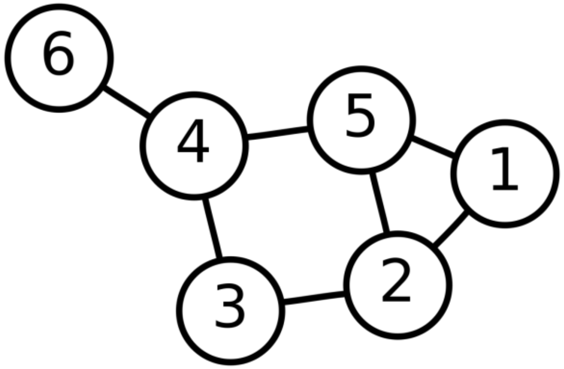
# Graphs : A powerful representation tool …



- Topology
  - Nodes = primitives, elements, parts
  - Edges = relations
- Attributes
  - Statistical : observations, distributions, …
  - Geometrical : metrics (distances, angles, similarities)
  - Positions : absolutes or relatives
  - Visual features : discriminative elements
- Trying to ensure
  - Stability (invariance)
  - Tolerance : noises, variations
  - Classes discrimination
- But
  - Symbolic VS numerical
  - Discretisation



# What is a structural representation?




# Introduction

- Terminology and notation

Definition and notation of a graph:

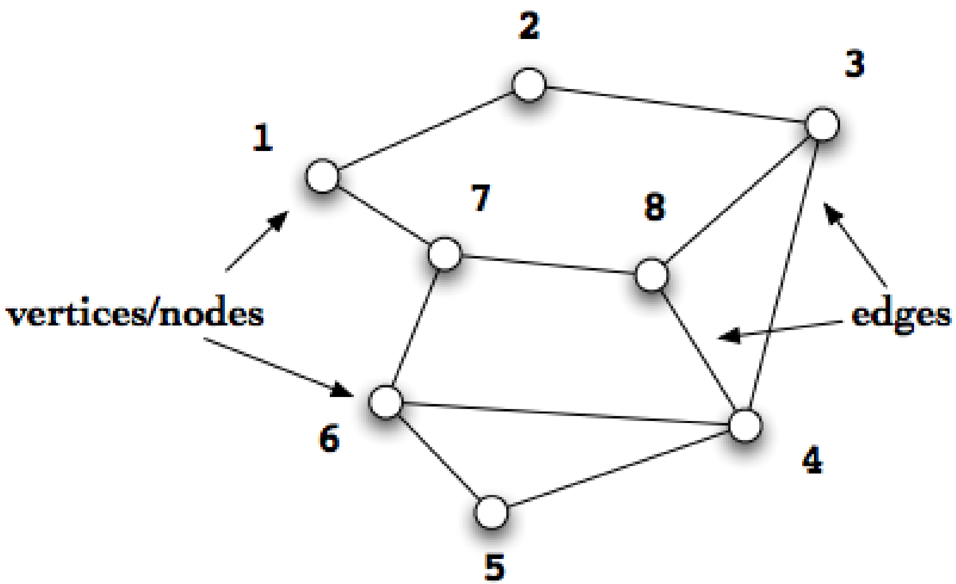
**Definition**

Let $L_V$ and $L_E$ denote the set of node and edge labels, respectively. A labeled graph G is a 4-tuple $G = (V, E, \mu, \xi)$ , where

- $V$ is the set of nodes,
- $E \subseteq V \times V$ is the set of edges
- $\mu : V \to L_V$ is a function assigning labels to the nodes, and
- $\xi : E \to L_E$ is a function assigning labels to the edges.

- Let $G_1 = (V_1, E_1, \mu_1, \xi_1)$ be the source graph
- And $G_2 = (V_2, E_2, \mu_2, \xi_2)$ the target graph
- With $V_1 = (u_1, ..., u_n)$ and $V_2 = (v_1, ..., v_m)$ respectively

# Adjacency matrix

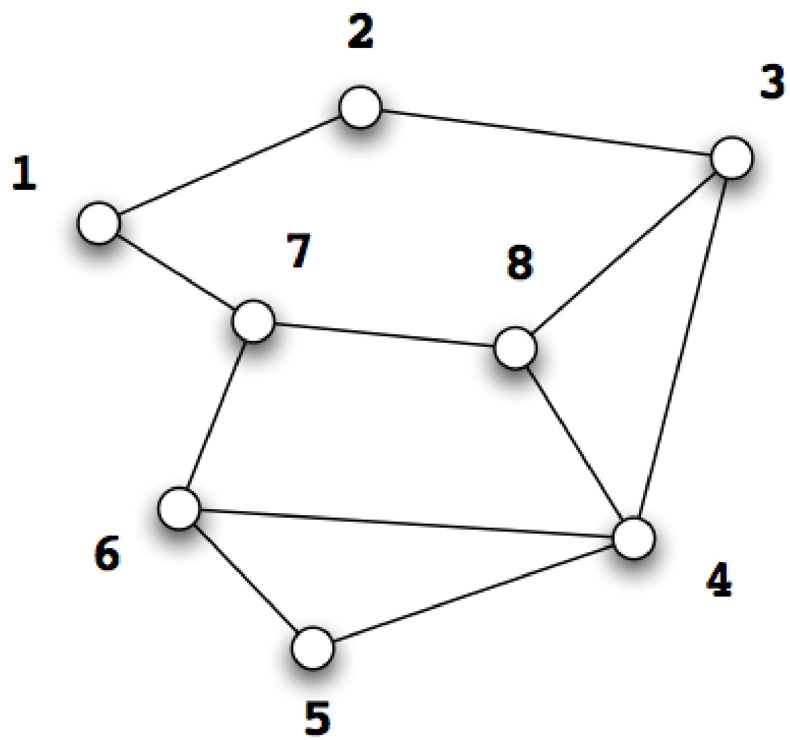



$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \end{bmatrix}$$

**Undirected Graph G(V, E)**

sub-matrix of $A$ = a subgraph of $G$

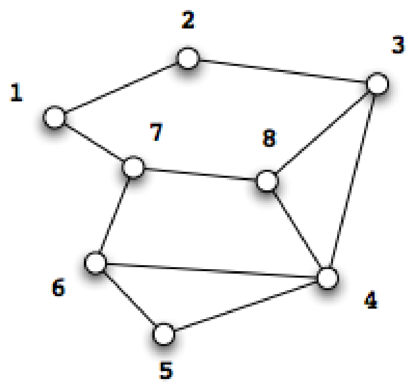
# Degree matrix




$$D = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 \end{bmatrix}$$

Normalized Adjacency matrix $\tilde{A} = D^{-1}A$ is a stochastic matrix (each row sums to one)

# Laplacian Matrix




$$L = \begin{bmatrix} 2 & -1 & 0 & 0 & 0 & 0 & -1 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 3 & -1 & 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 4 & -1 & -1 & 0 & -1 \\ 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & -1 & 3 & -1 & 0 \\ -1 & 0 & 0 & 0 & 0 & -1 & 3 & -1 \\ 0 & 0 & -1 & -1 & 0 & 0 & -1 & 3 \end{bmatrix}$$

$$L = D - A$$

Normalized version

$$\tilde{L} = D^{-\frac{1}{2}}(D - A)D^{-\frac{1}{2}}$$

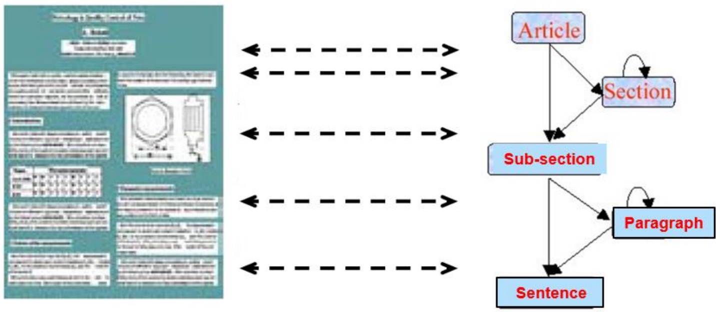
Spectrum bounded between $0$ and $2$

# EXTRACTION OF STRUCTURAL PRIMITIVES INSIDE IMAGES

# Structural representations

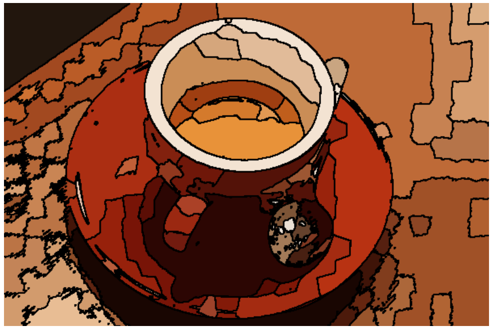
## From pixels to EoC

- **Elementary Element of Content constituting an image**
  - Vectors, regions, points, CC, …
  - Size, shape, colors of the elements



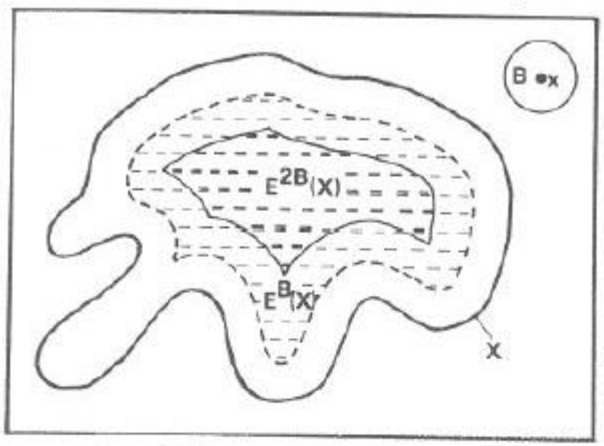

- **Physical structure**
  - Decomposition into sub-parts
  - Relation between EoC : neighbourhood, distances,…
  - Image = Superposition of layers

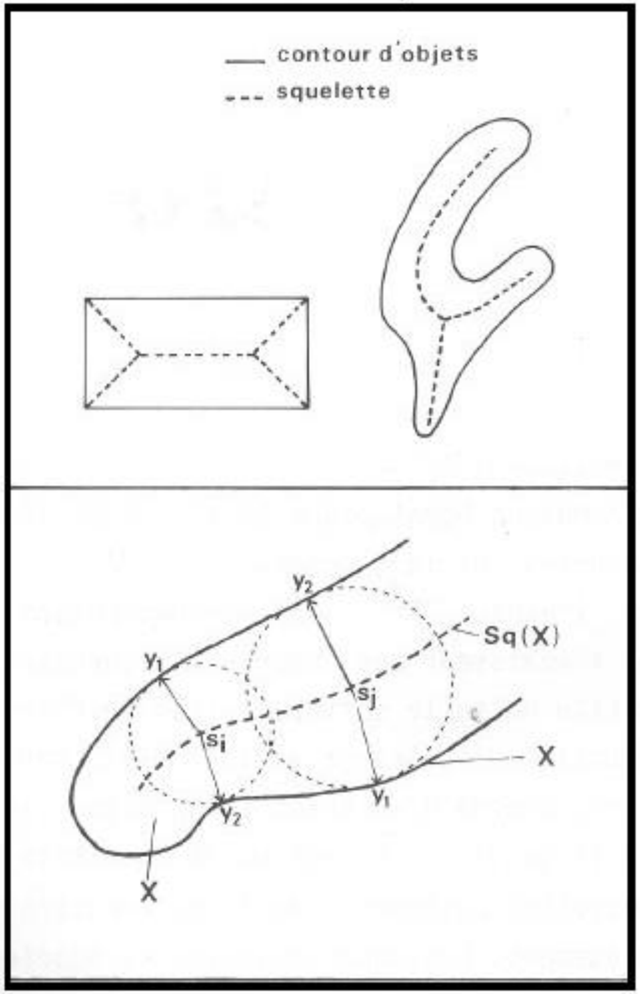

- **Logical structure**
  - A priori knowledge – atlas
  - Semantical aspects
  - Knowledge about the possible content and organization of the images
  - Hierarchical organization of primitives



# For binary images- Skeletonization

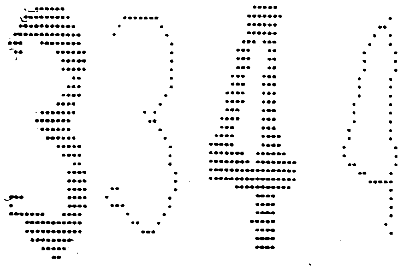
- **Definition (**skeleton or topological skeleton)



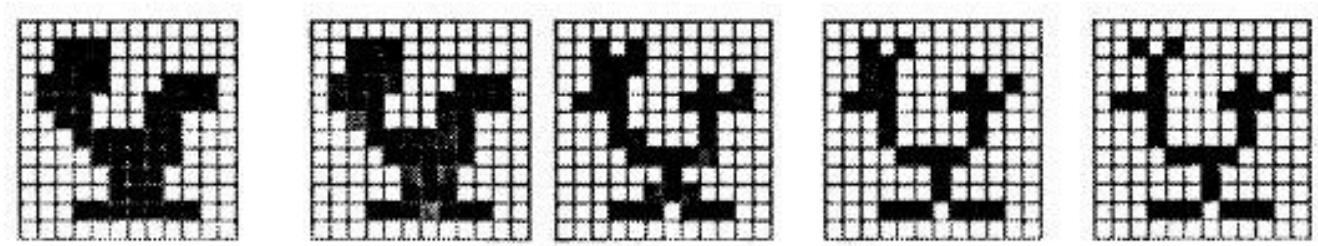



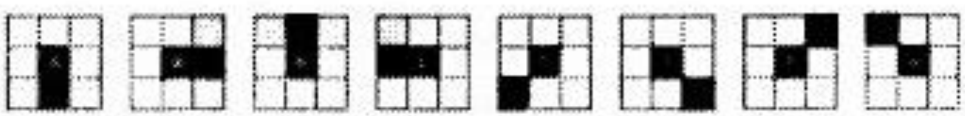

- A thin version of the shape defined by successive pixels equidistant to the boundaries
- Emphasizes geometrical and topological properties such as topology, length, width and connectivity
- Distance transform: Together with the distance of its points to the shape boundary, the skeleton contain all the information necessary to reconstruct the shape

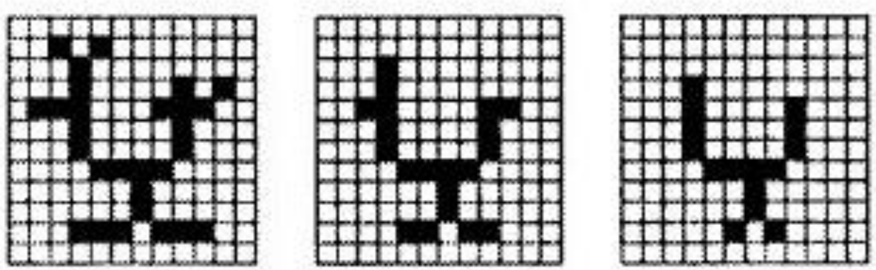
# Skeletonization

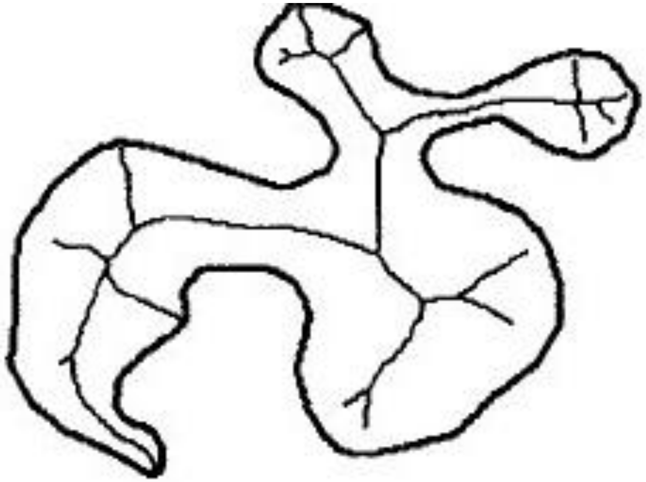


## Extraction algorithm: Iterative erosions



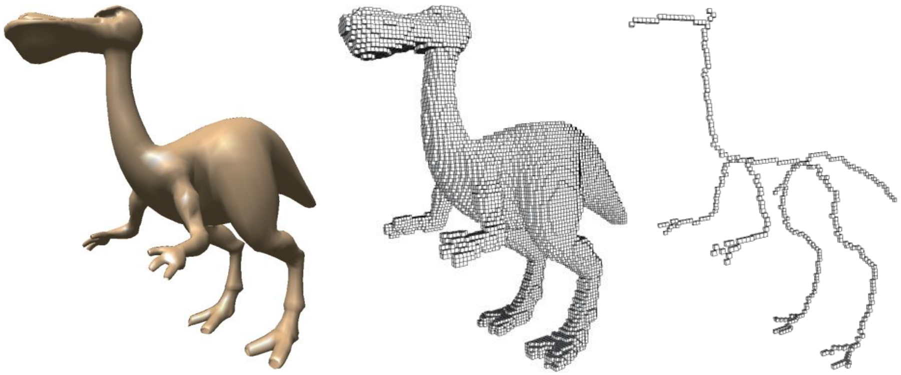


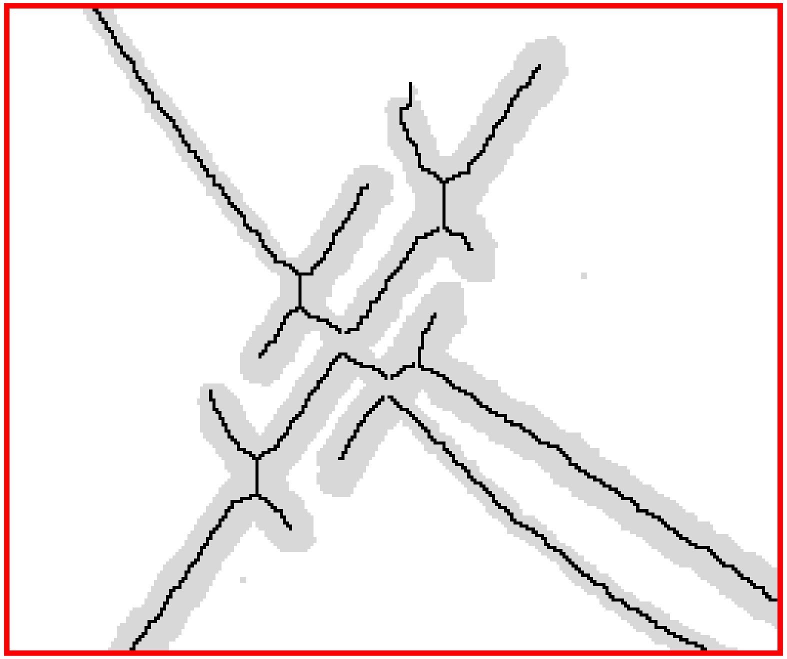


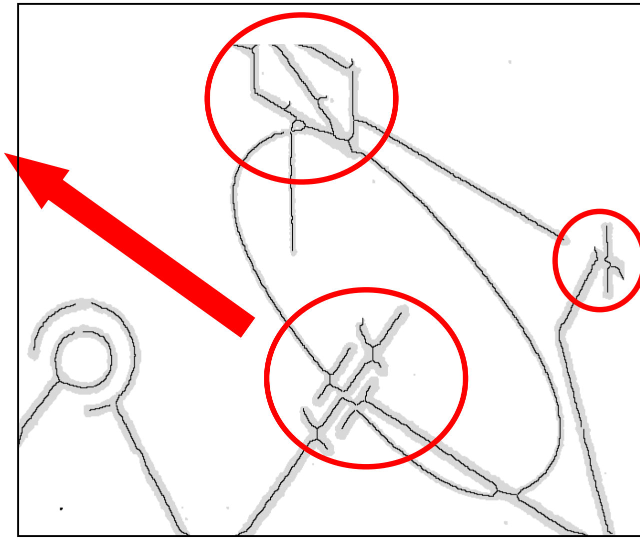


# Problems with skeleton . . .

- Junctions, intersections, barbules, …
- Time consuming, not robust
- But used a lot…

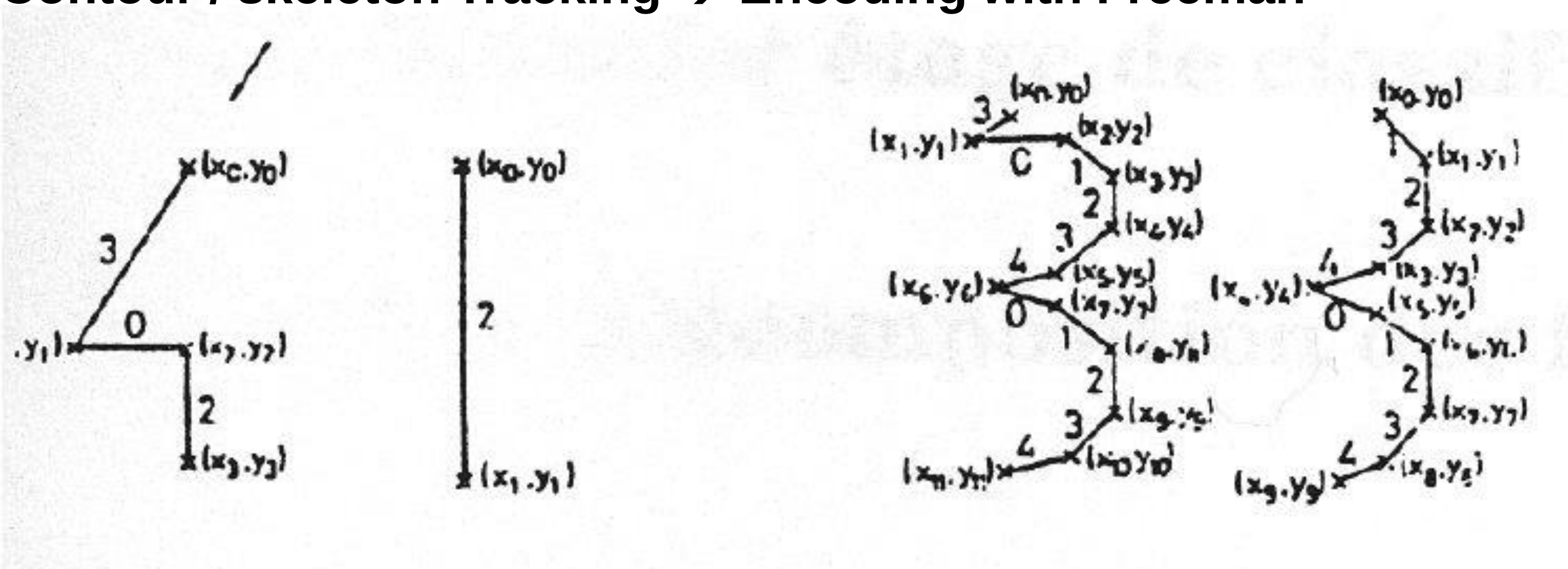


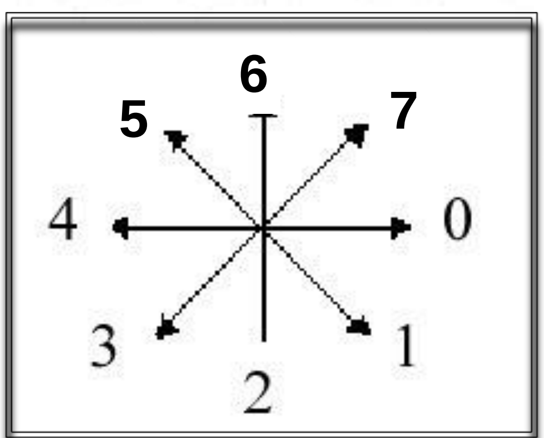




# Skeleton representation?

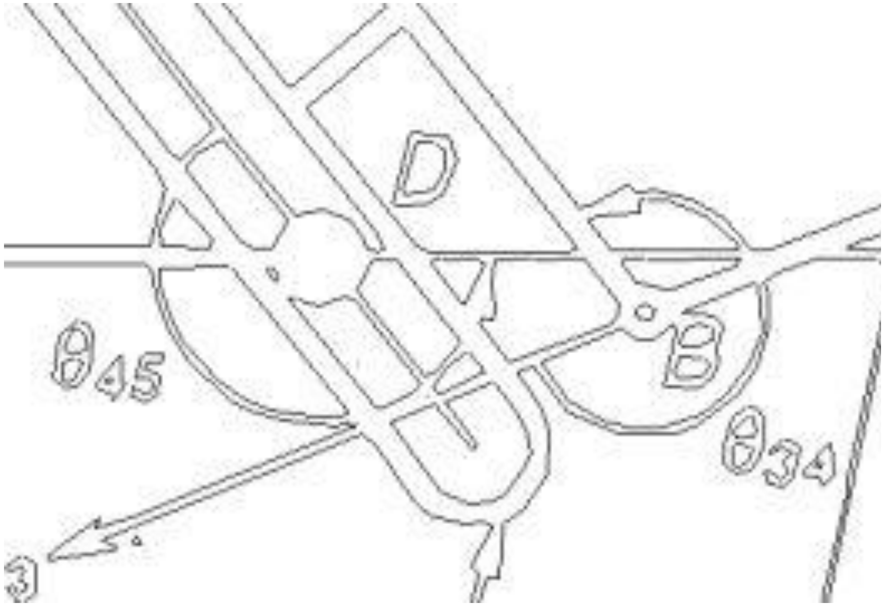
## Contour / skeleton Tracking → Encoding with Freeman

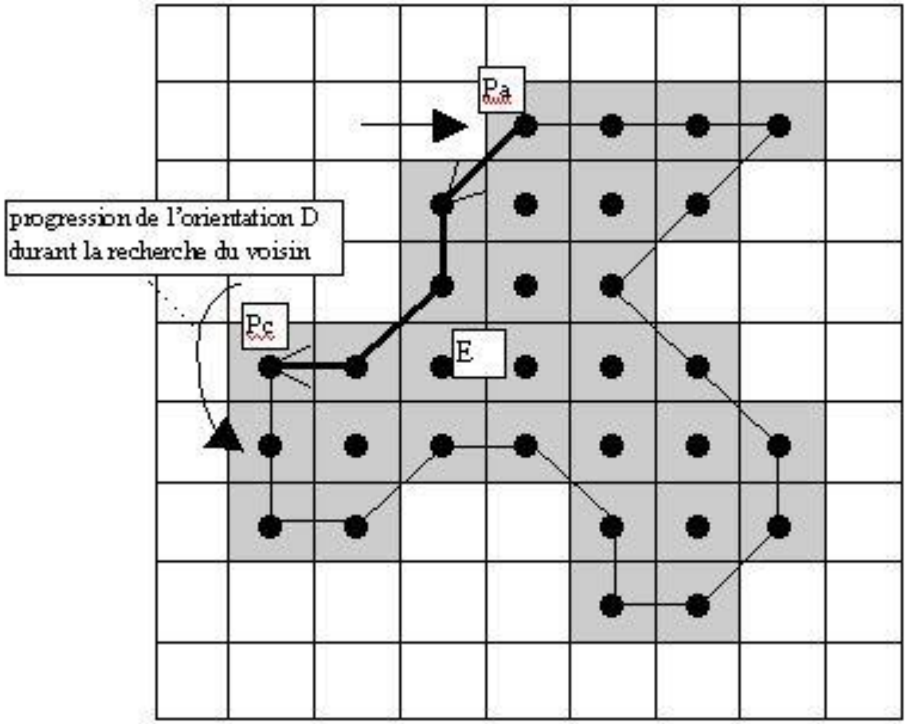




# Polygonal approximation→ Vectorisation

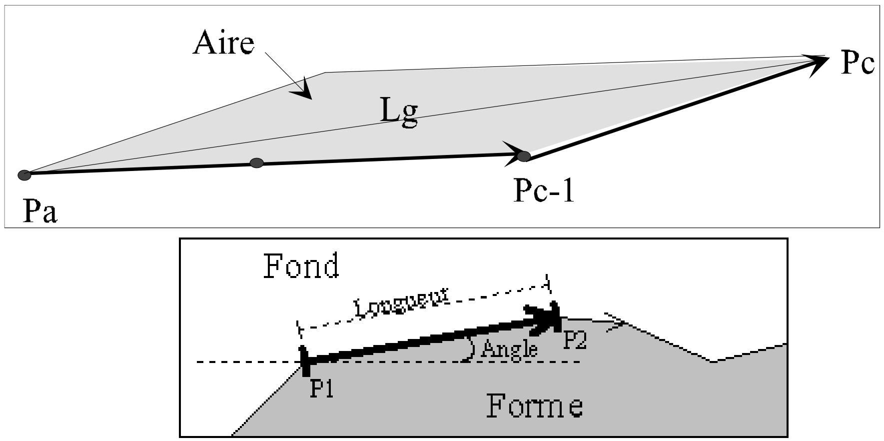
- From pixels to vectors

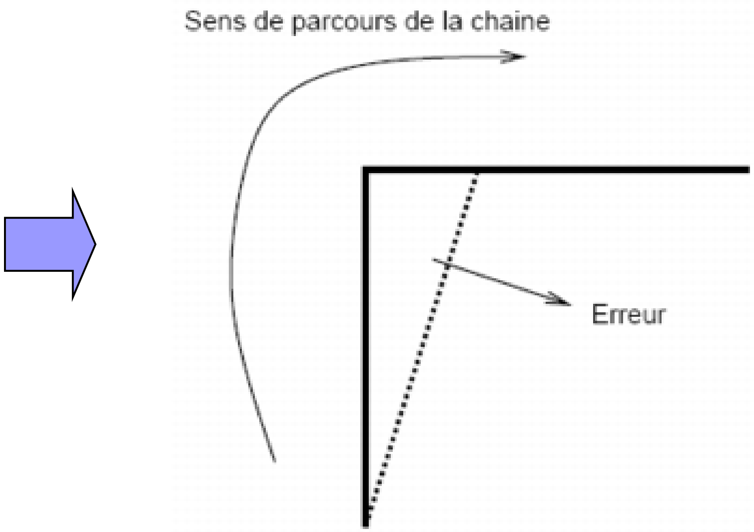


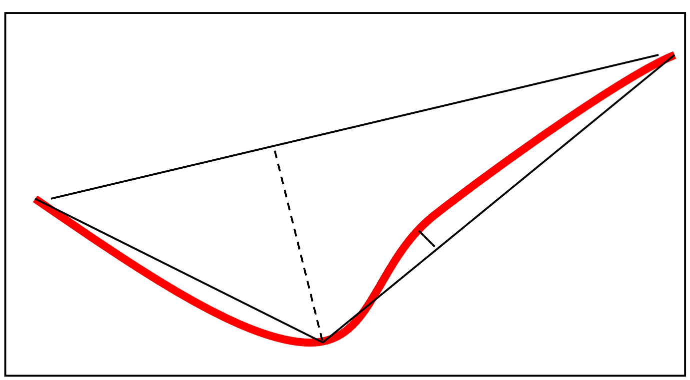
# Polygonal approximation→ Vectorisation

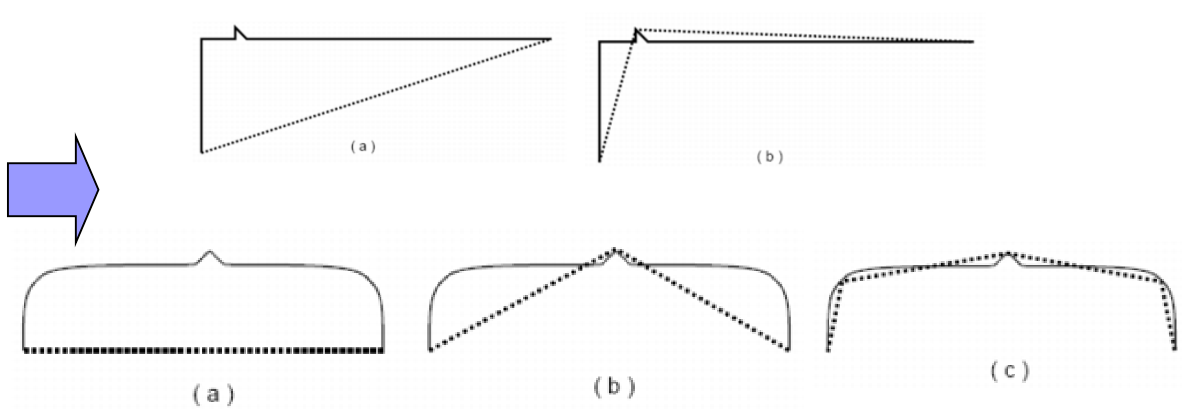
- Iterative Methods [Wall 84]



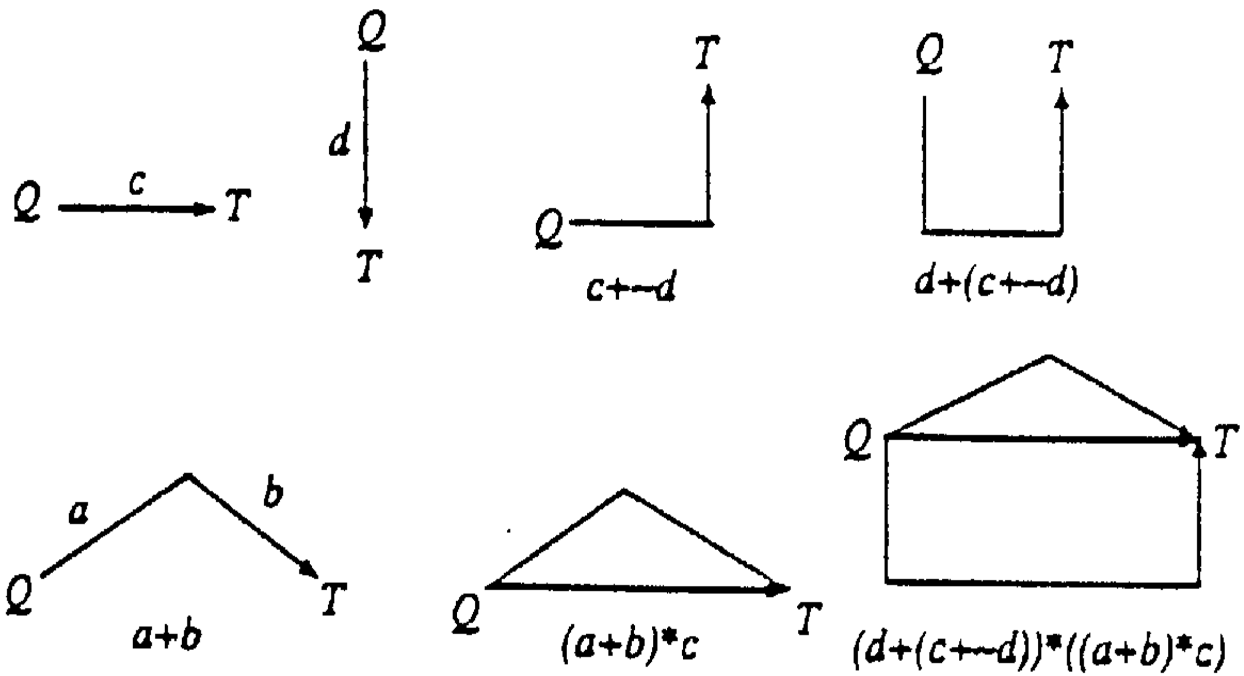

- Recursive Methods





# Relationships between primitives

- Syntactic methods ➔ Grammars
  - Adapted to 1D sequences ➔ Sequence of elements




- **Not adapted to 2D or 3D images**

# Relationship between primitives

**Spatial Relationship**

- Bi dimensional Allen Algebra
- Egenhofer algebra

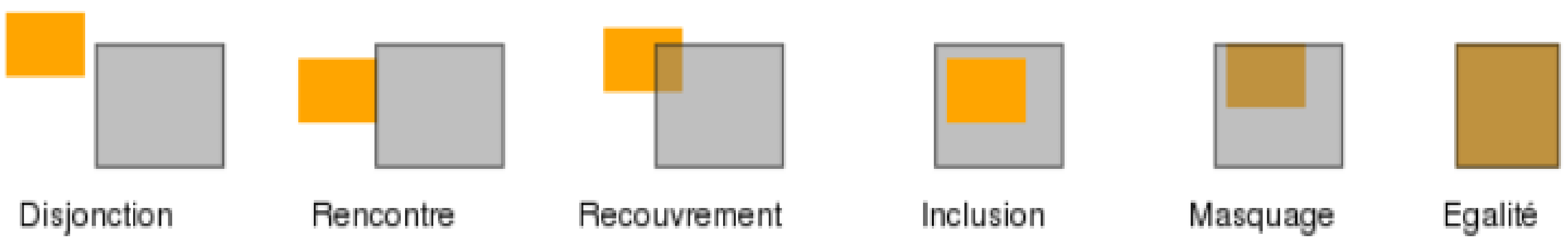
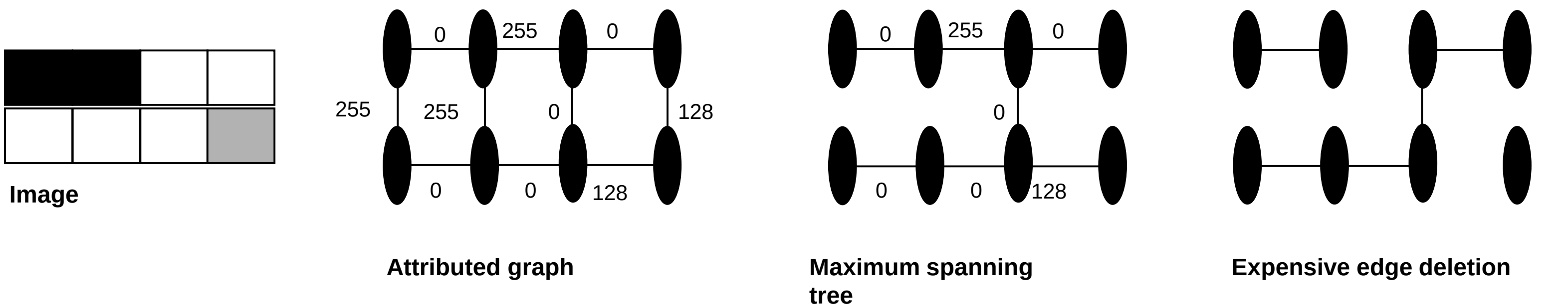


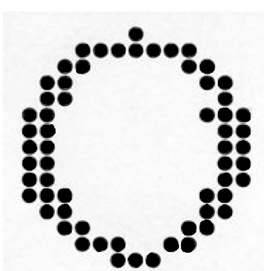

**Fig. 1.12.** Relations topoliques entre deux objets telles que définies par Egenhofer

# Graph of pixels

- Pixels ➔ The nodes of the graph
- Edges ➔The values (RGB, grey values)

[Morris, 1986]



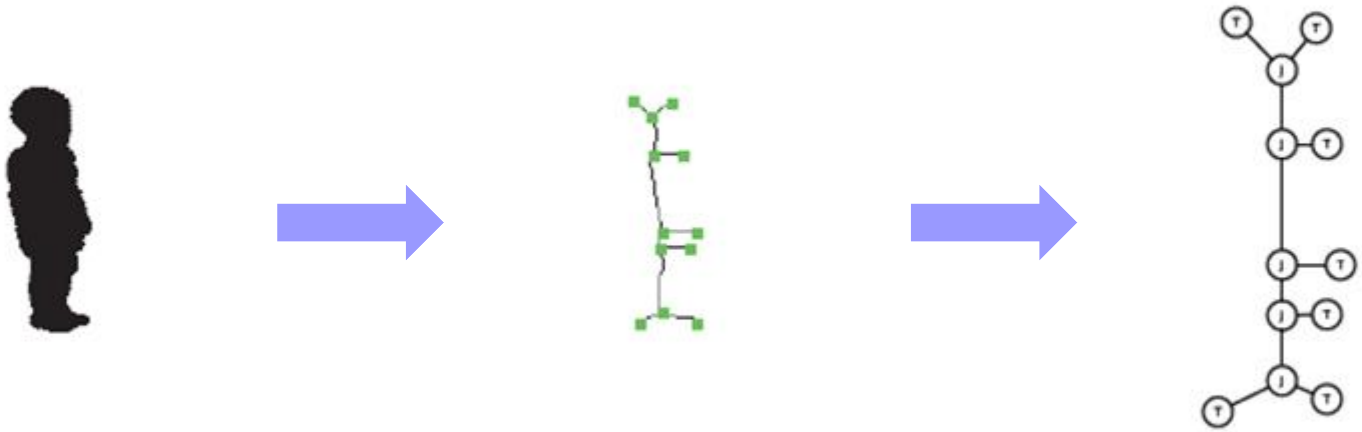



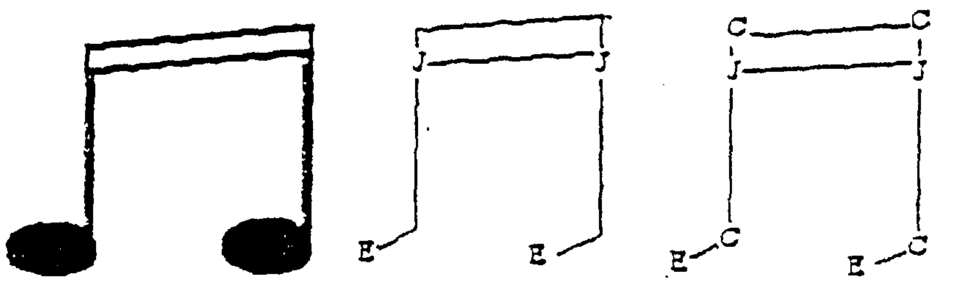
[Franco, 2003]

## Problem

Graphs made of pixels are often too big to be analysed

# Skeleton Graph



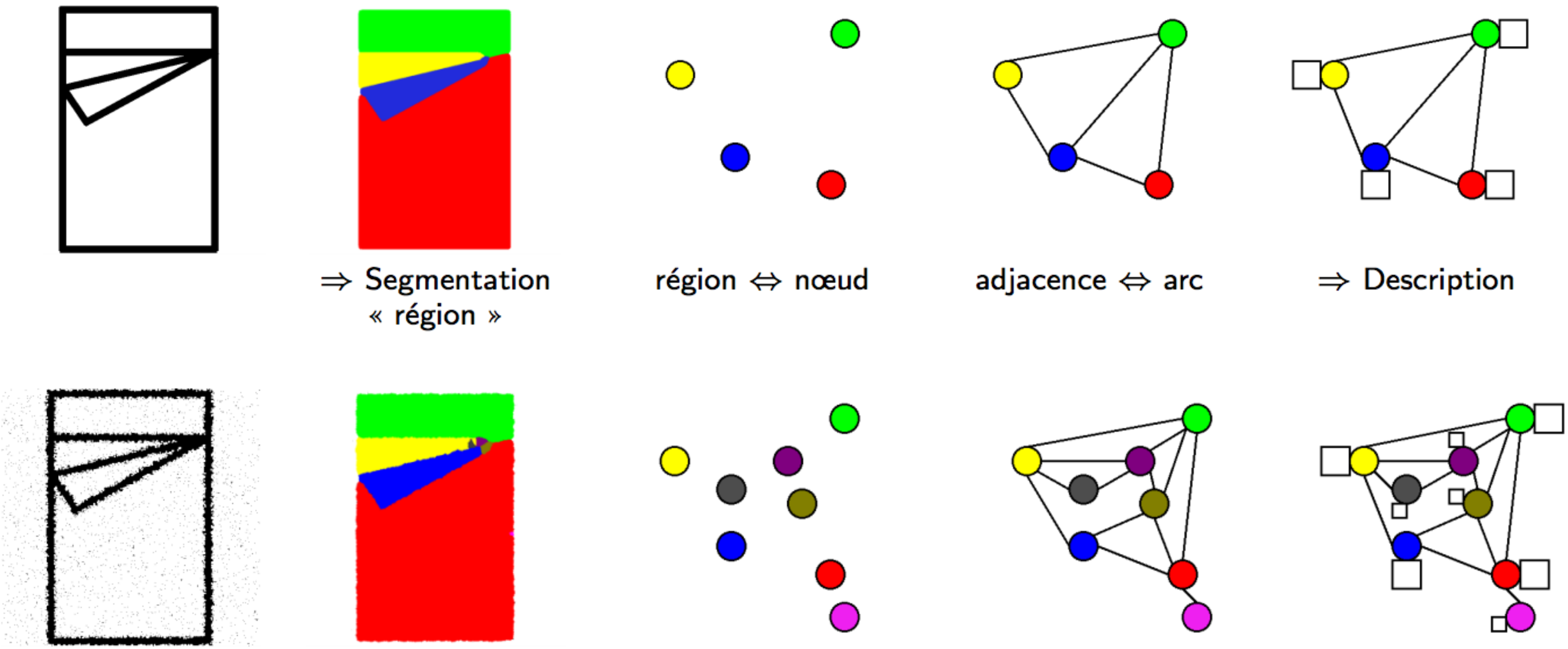



Objet : 2 doubles croches

Graphe du squelette avant découpage récursif :
4 segments
2 extrémités (E)
2 jonctions (J)

Graphe du squelette après découpage récursif :
8 segments
2 extrémités (E)
2 jonctions (J)
4 coupures (C)

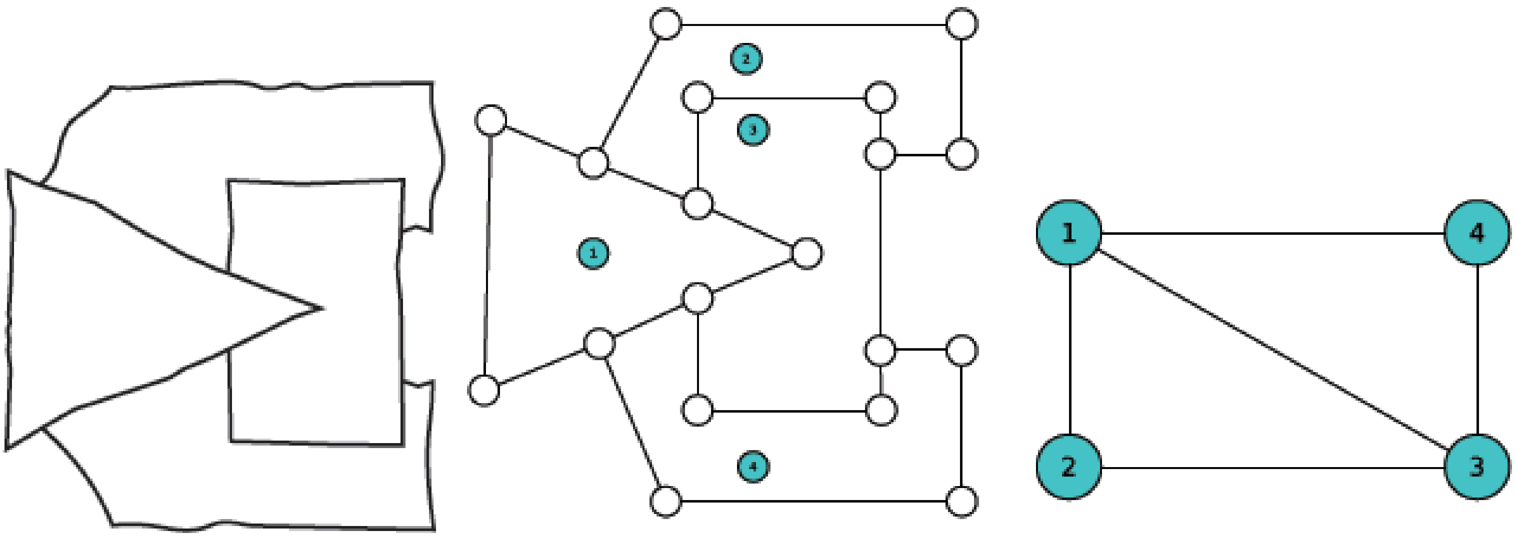
# Region Adjacency Graph




[Locteau 2008]

# Region Adjacency Graph

- Double representation

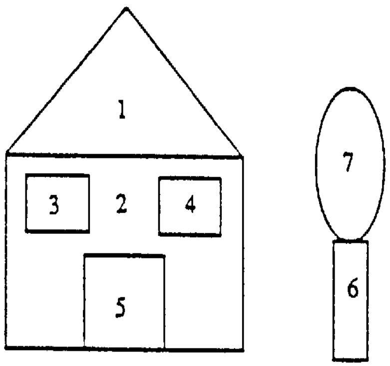


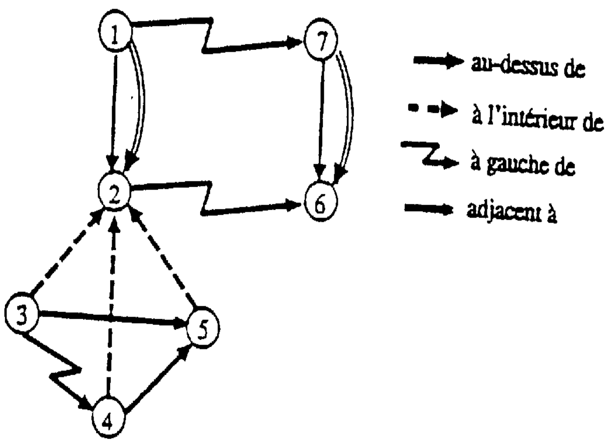

(a) Symbole original

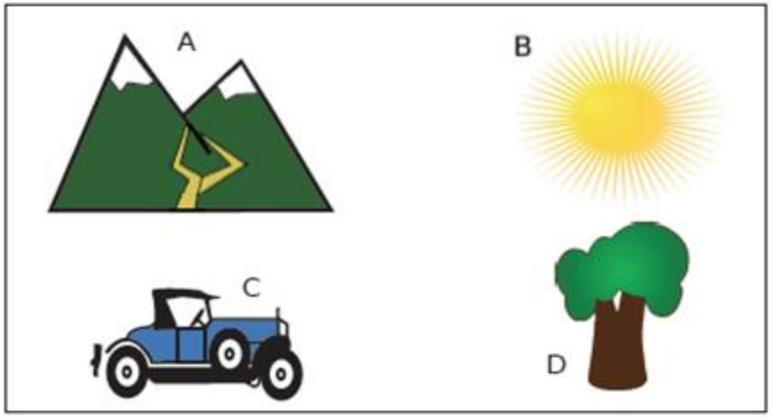
(b) Graphe correspondant

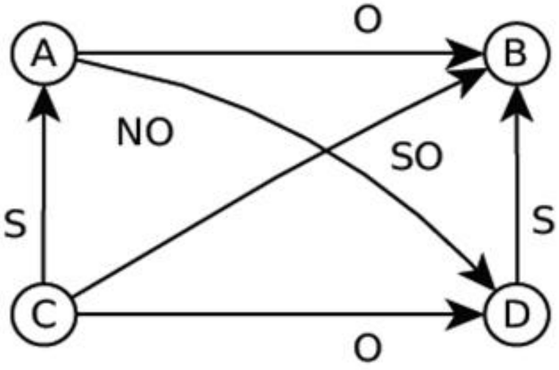
(c) Graphe d'adjacence des régions

# Spatial relationship graph



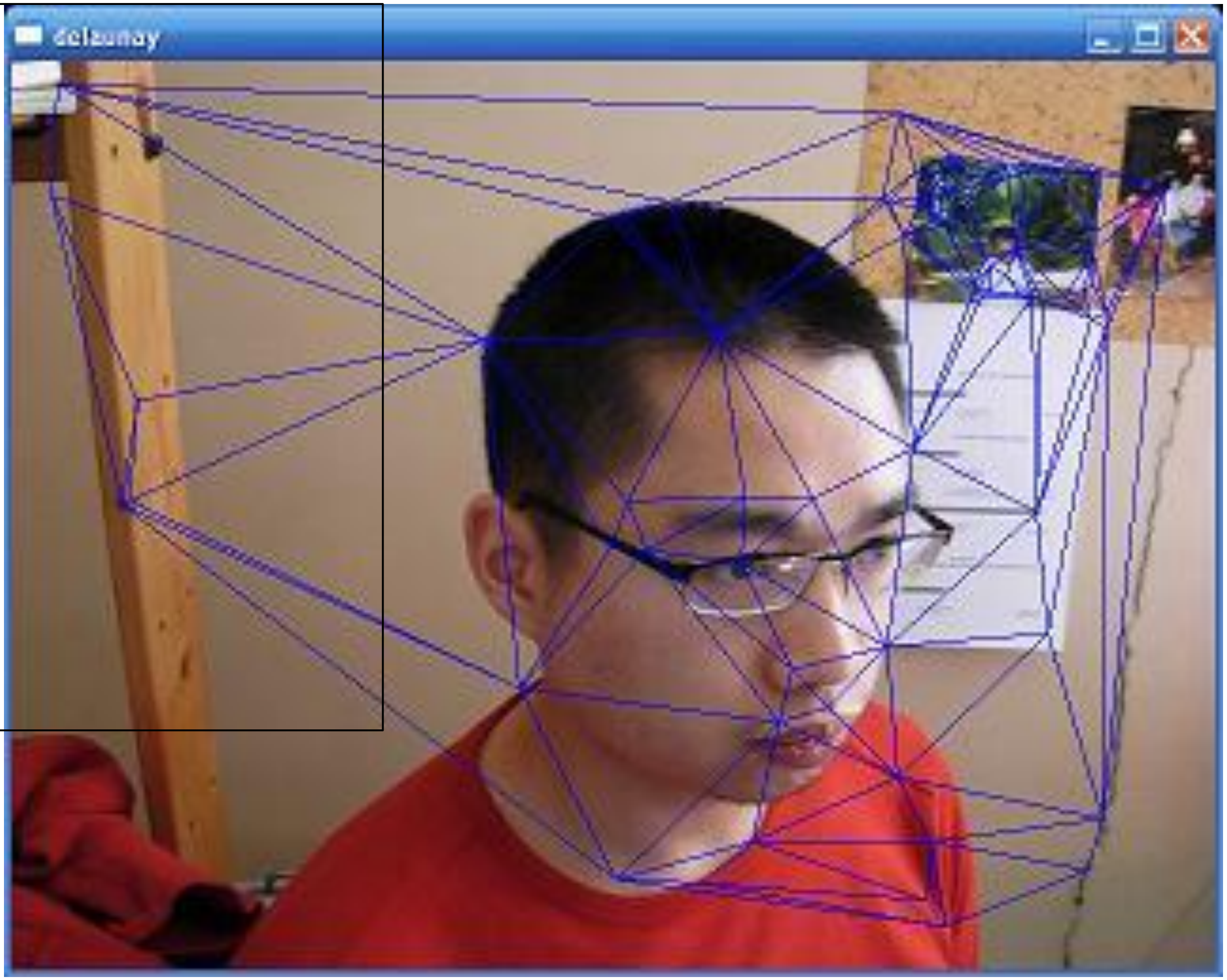



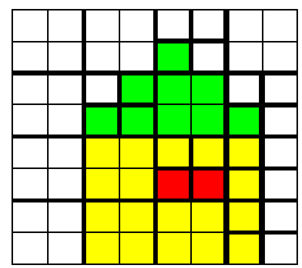
# Interest Point Graph

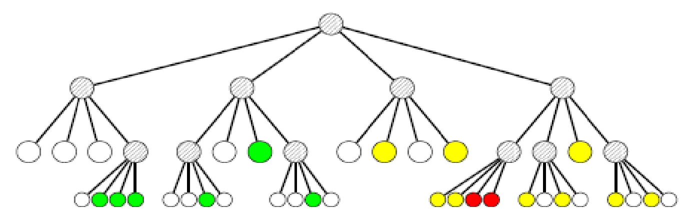
- Node ➔ Keypoints
- Edges ➔ distances between Keypoints
- Many other possibilities
  - Similarities
  - Angles
  - …

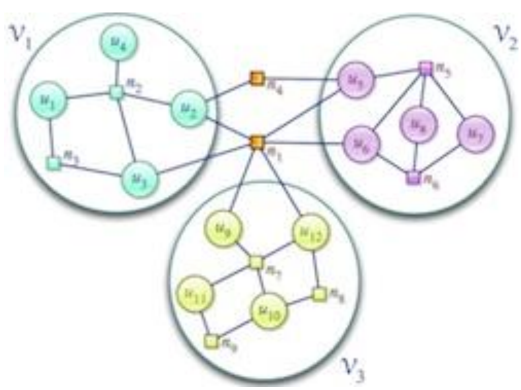


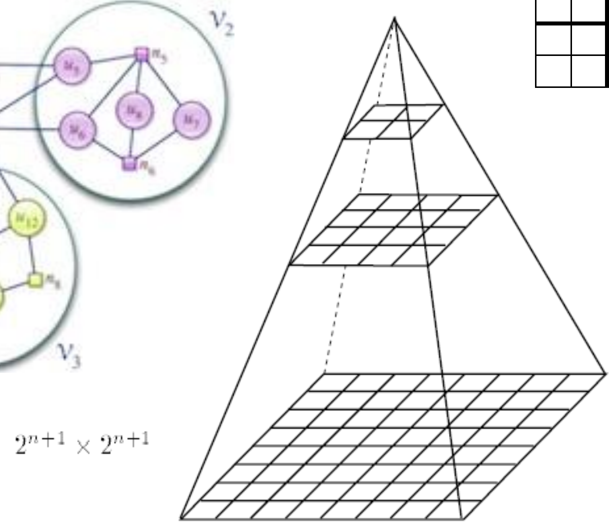
# Multi-level representations

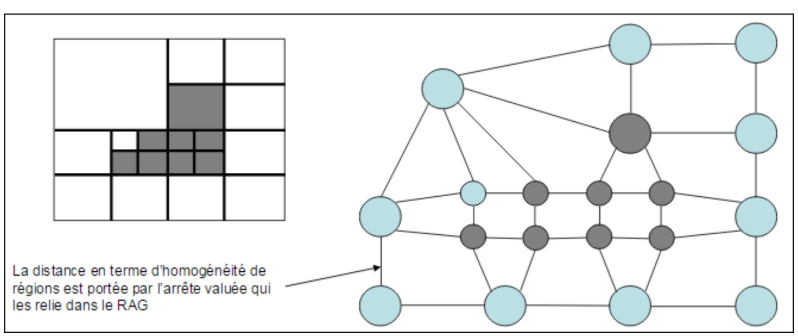
- **Quadtree**
  - Root = Full Image
  - Recursive splitting into 4 regions
  - Split : each node has 0 or 4 children
  - Merge : possible merging of nodes
- **Hyper-graphs, pyramids, …**





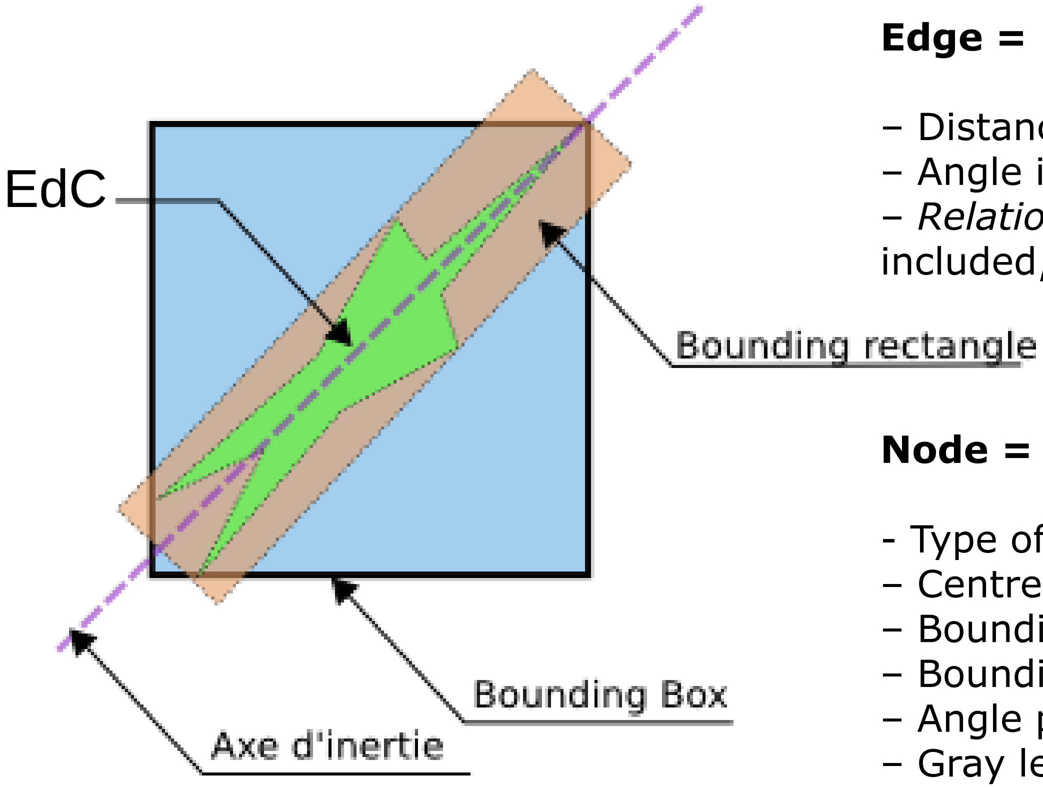


# A detailed example

## Image → Regions/primitives → Graph of EoC




**Edge = Relation between EoC**

– Distance minimal between 2 EdC
– Angle inter EdC
– *Relation* : Left, Right, above, below, inside, partially included, L , T ,P , X , S ,…

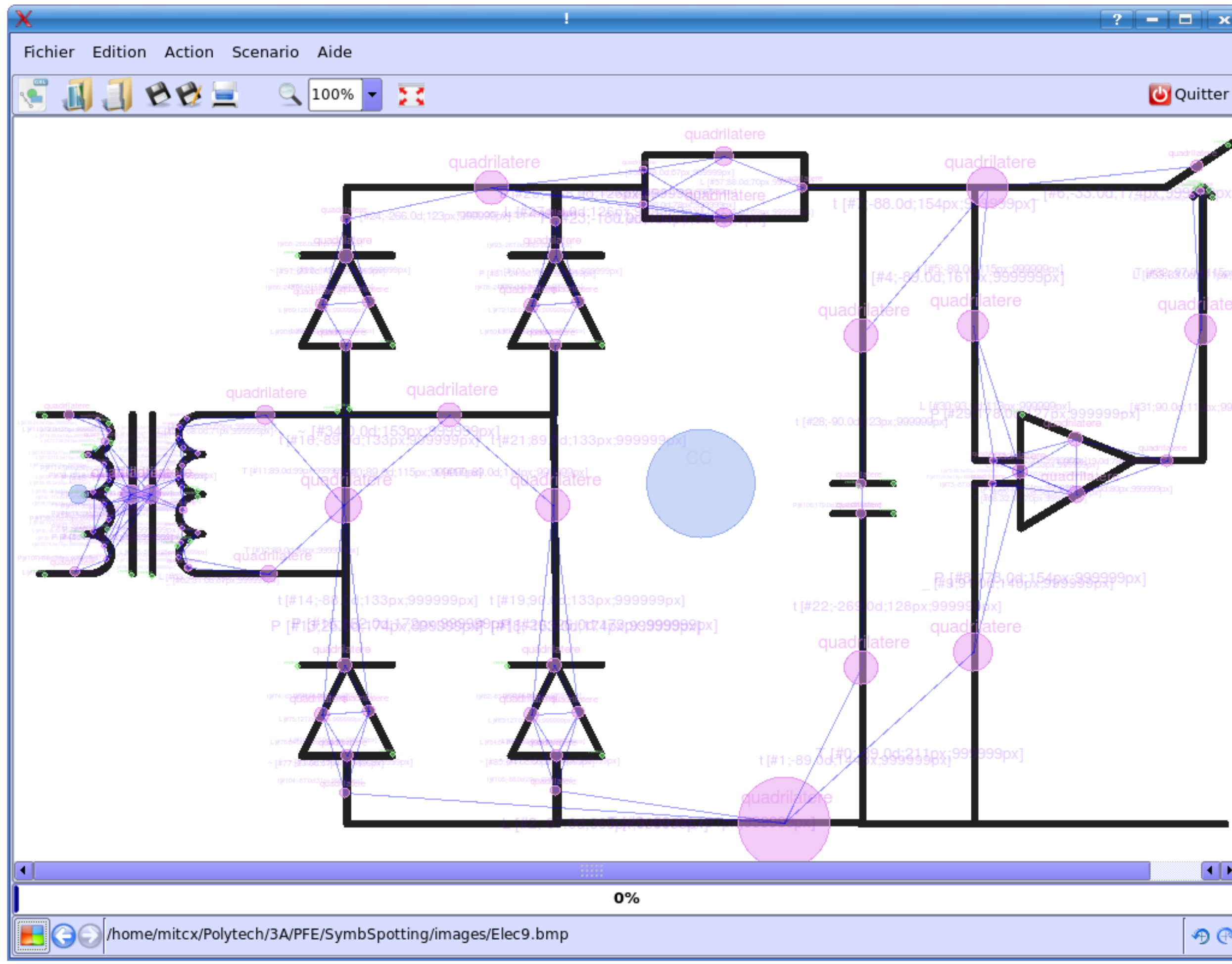
**Node = EoC**

- Type of EdC
– Centre c :(X,Y) de la Bounding Box
– Bounding Box
– Bounding Rectangle BR :(P1,P2,P3,P4)
– Angle principal = angle du rectangle englobant
– Gray level inside BB
– Color : average inside EdC
– Confidence rate
…

UNIVERSITÉ FRANÇOIS - RABELAIS TOURS

# A detailed example

Graph of EoC



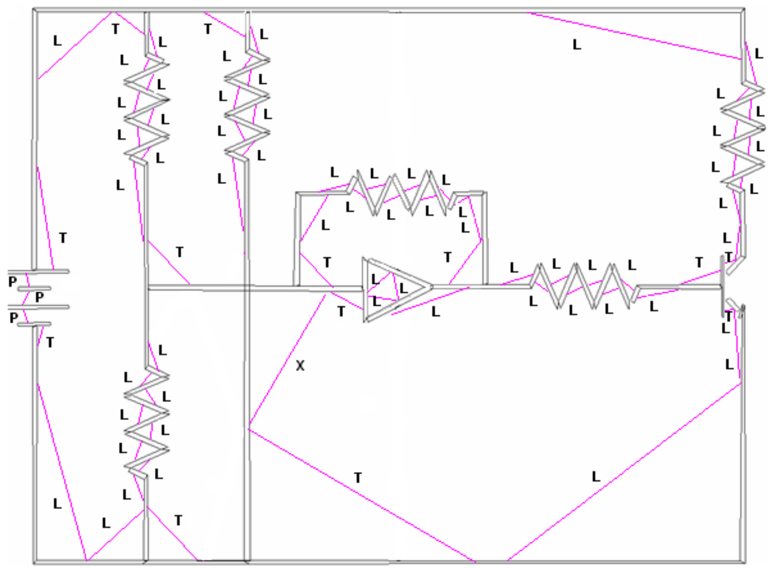

Initial graph

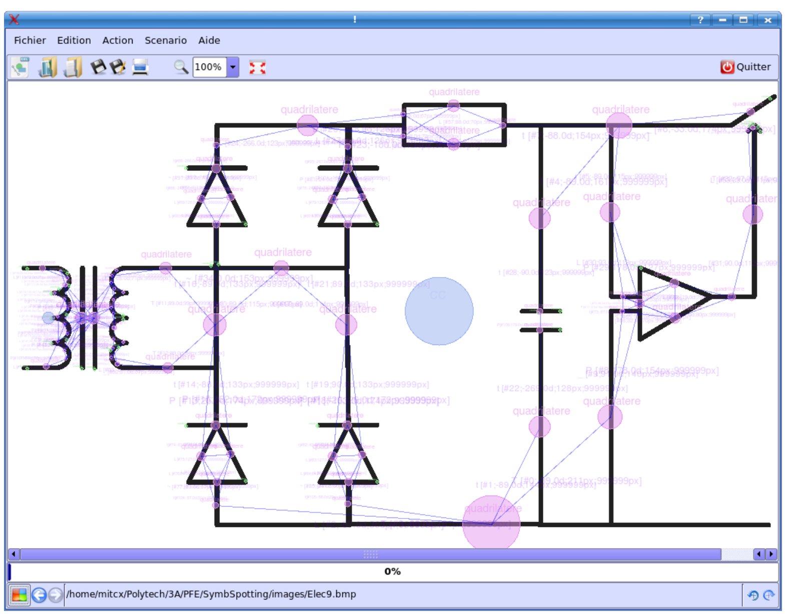
- Connected Components
- Vectors
- Quadrilaterals

Can evolve into:

- Characters
- Words
- Triangles
- Diodes
- …

# A detailed example





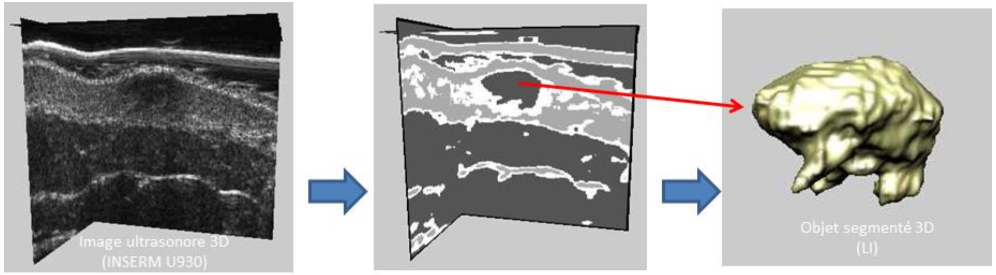


```
<Graph id="Symbole4">
  <node id="node0">
    <attr name="forme"><string>Quad</string></attr>
    <attr name="x1i"><string>232</string></attr>
    <attr name="y1i"><string>497</string></attr>
    <attr name="x1f"><string>231</string></attr>
    <attr name="y1f"><string>417</string></attr>
    <attr name="x2i"><string>227</string></attr>
    <attr name="y2i"><string>418</string></attr>
    <attr name="x2f"><string>229</string></attr>
    <attr name="y2f"><string>498</string></attr>
    <attr name="angle1"><string>90</string></attr>
    <attr name="angle2"><string>91</string></attr>
    <attr name="thickness1"><string>8</string></attr>
    <attr name="thickness2"><string>7</string></attr>
    <attr name="length"><string>81</string></attr>
    <attr name="score"><string>0.703</string></attr>
  </node>
    ...
    ...
    ...
<edge id="edge1" from="node0" to="node1">
    <attr name="angle"><string>89</string></attr>
    <attr name="type"><string>T</string></attr>
    <attr name="score"><string>0.409</string></attr>
</edge>
<edge id="edge2" from="node1" to="node2">
    <attr name="angle"><string>46</string></attr>
    <attr name="type"><string>L</string></attr>
    <attr name="score"><string>0.388</string></attr>
</edge>
...
...
```

# USING GRAPH
# FOR IMAGE SEGMENTATION

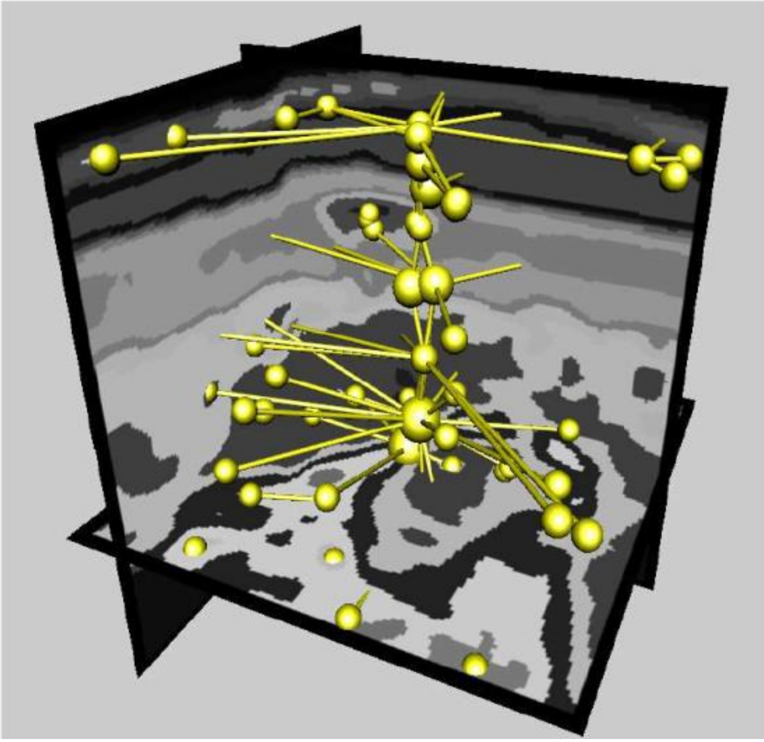
# Interactive image segmentation

- **RAG for an incremental segmentation of 3D images**
  - Improving the quality of the segmentation by interactive transformations
  - Addition and deletion of nodes & edges
  - Splitting or Merging nodes
  - Annotation of nodes and edges




  - Interactive Features selection

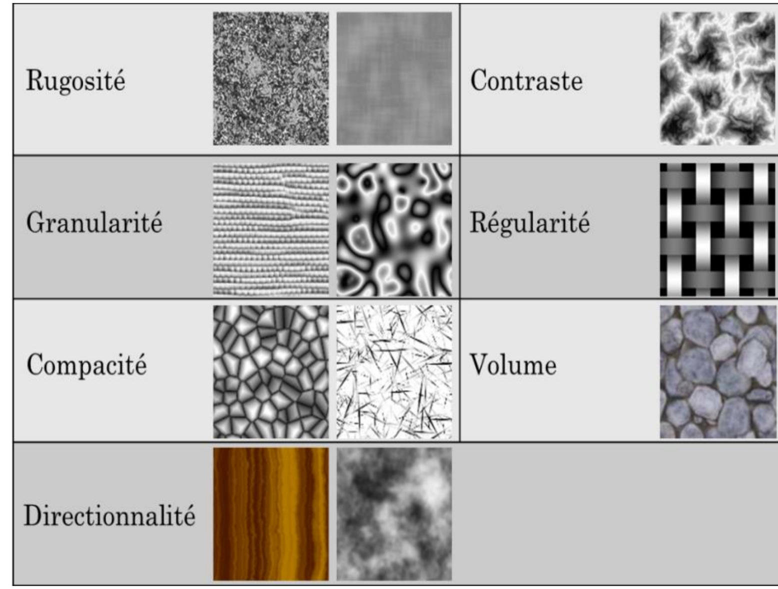
**What is this graph?**



# Interactive image segmentation

## Voxel characterization by understandable features

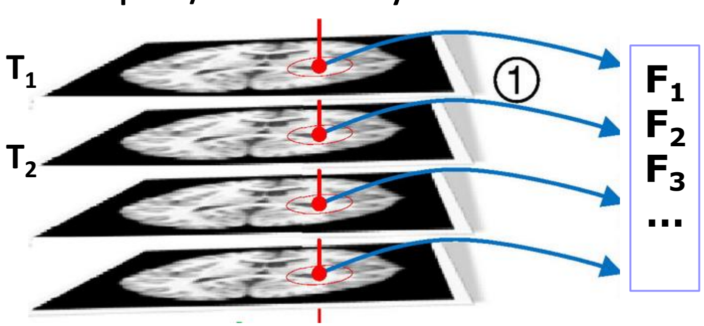
- Registration of the different modalities ( T1, T2, …)
- 1 voxel = 1 list of features $R_j$ instead of a color ➔ $Voxel_i = [F_1\ F_2\ F_3\ \ldots]$
- 1 voxel = 1 individual




## Region characterization

- Average of the features computed on all the voxels of the region

1 pixel / Voxel → Many modalities



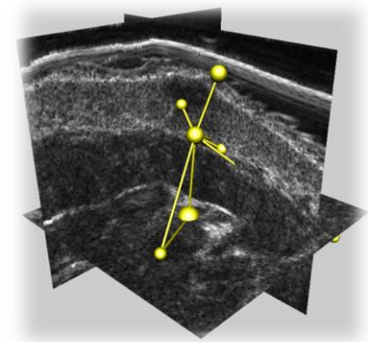

1 pixel / Voxel → Many understandable features

$Voxel_1 = [F_1\ F_2\ F_3\ \ldots]$
$Voxel_2 = [F_1\ F_2\ F_3\ \ldots]$
$Voxel_3 = [F_1\ F_2\ F_3\ \ldots]$
$Voxel_4 = [F_1\ F_2\ F_3\ \ldots]$
$Voxel_5 = [F_1\ F_2\ F_3\ \ldots]$
.....
$Voxel_N = [F_1\ F_2\ F_3\ \ldots]$

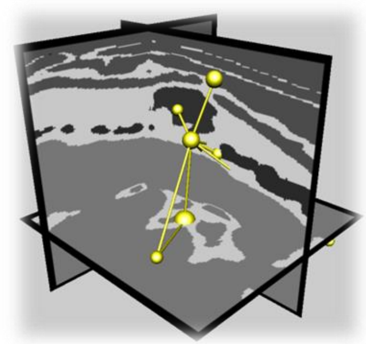
$R_1 = [\overline{F_1}\ \overline{F_2}\ \overline{F_3}\ \ldots]$

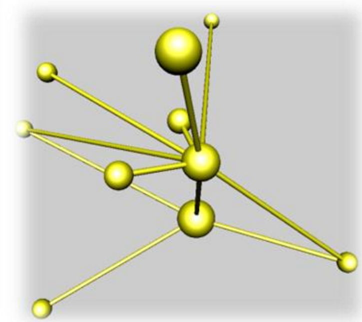
# Interactive image segmentation

- Structuration of the data with a RAG
- **Region = Node**
  - Attributes = The understandable features
    - Averages of the features inside the region F
    - Centre of gravity G(x,y,z)



- **Relationship = Edge**
  - Attributes = Relation descriptors
    - The 2 linked nodes
    - Intersection surface



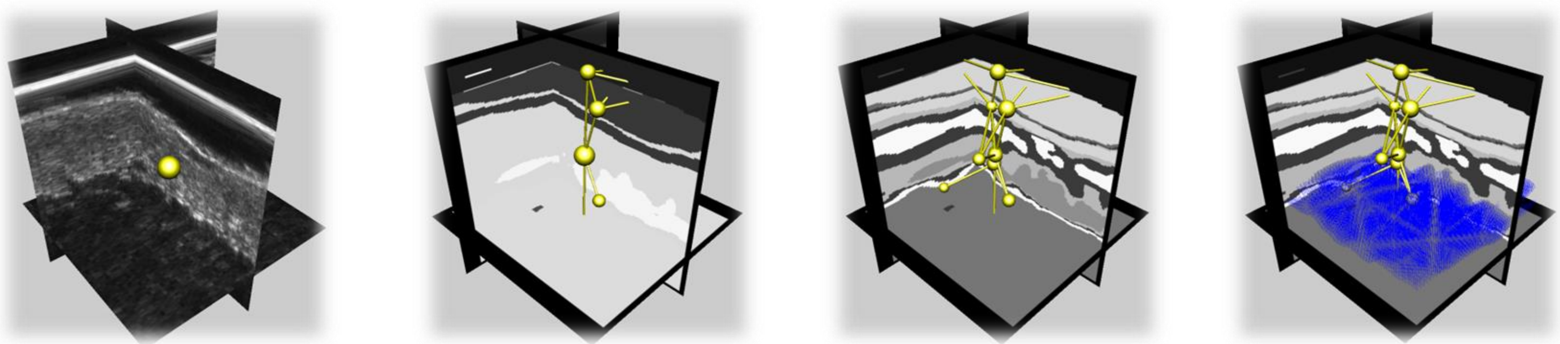


# Interactive image segmentation

## Initialisation

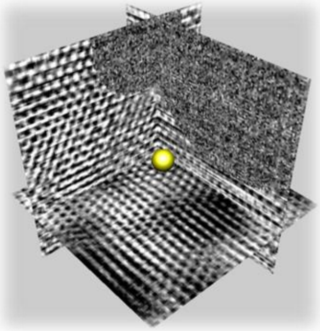
- RAG with 1 node = full image
- 1st step of the process : division
- After, what the user decides…

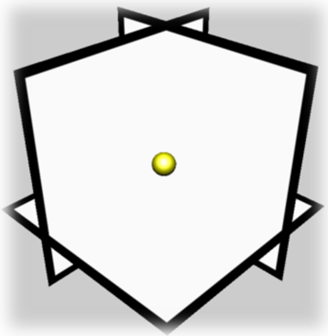
$$Seg: F, U_{Seg}, G_k, U_{op} \rightarrow G_{k+1}$$



# Interactive image segmentation

- **Division operator**
  - Fonction *Seg* corresponding to *K-means*
    - Fast and low memory
    - Robust Partitioning





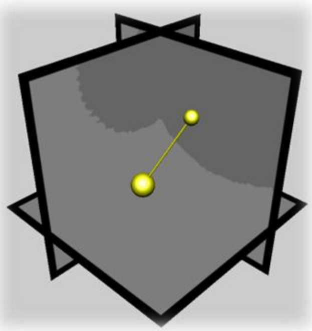
  - New graph $G_{k+1}$ :

$$G_{k+1} = Kmeans(U_{op}.V, G_k, F, U_{Seg})$$

$U_{Seg}$ : desired nb of regions

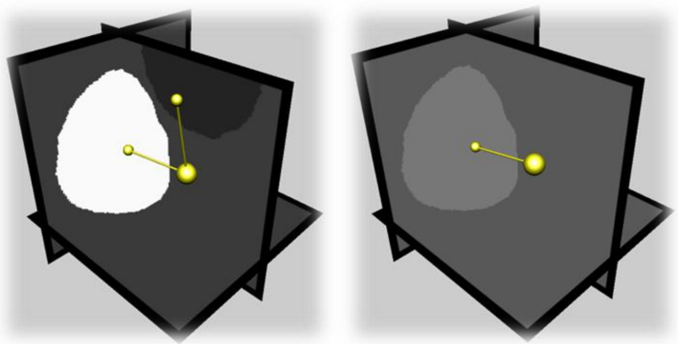
$F$ : selected features

$U_{op}.V$ : node to divide



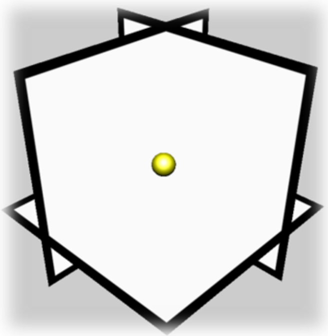
# Interactive image segmentation

- **Merging operator**
  - Union of the 2 selected regions
  - New graph $G_{k+1}$ :



$$G_{k+1} = Merge(U_{op}.E, G_k)$$

  - Updating of the node attributes $V_{new}$ :

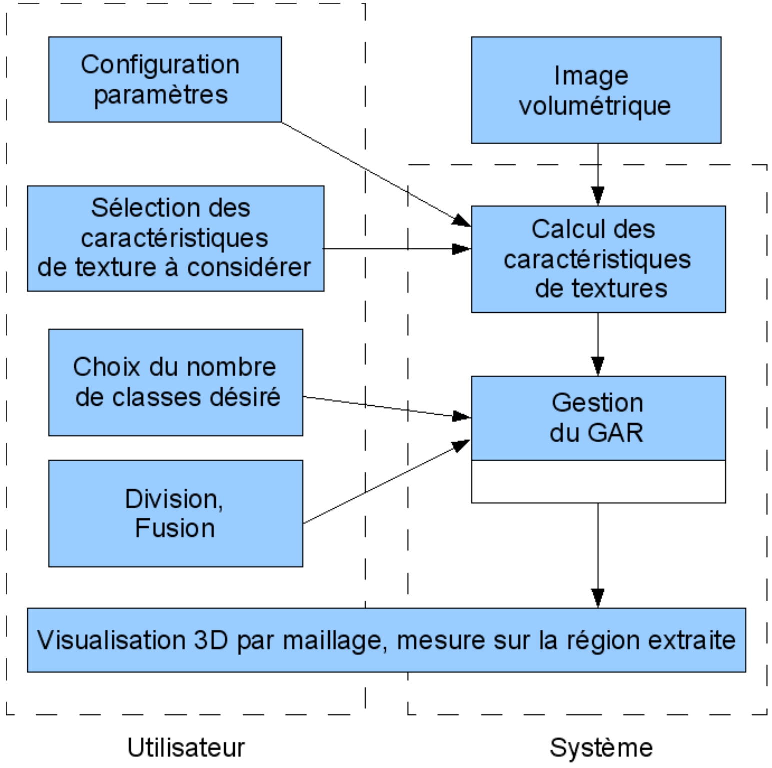


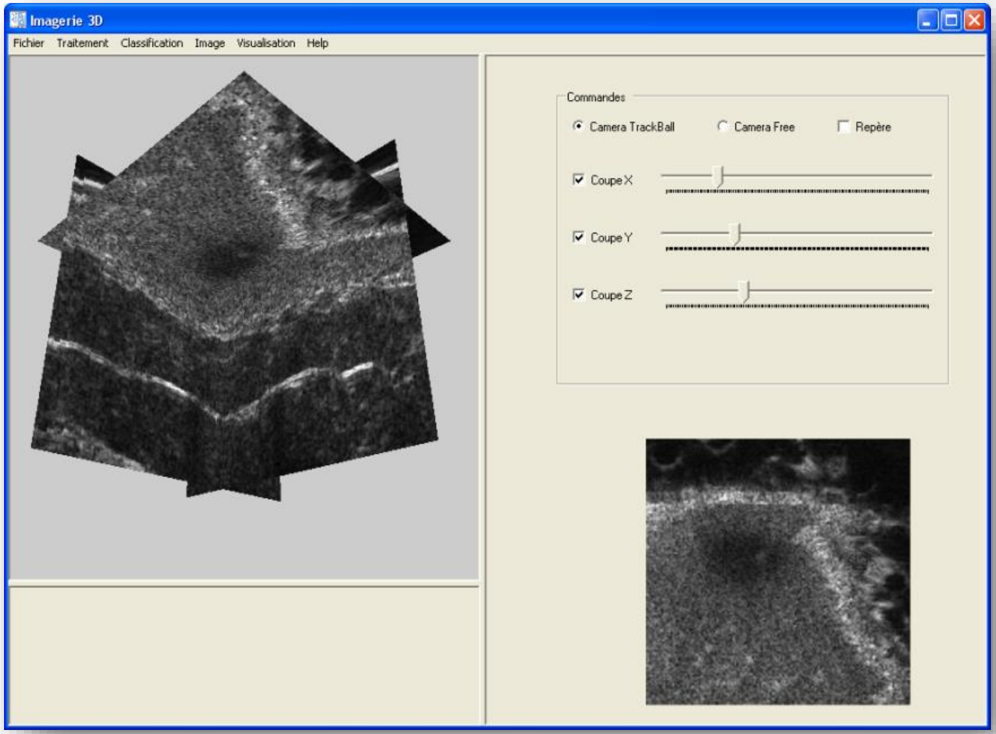
$$V_{new}.T_i = \frac{(V_1.T_i)(V_1.NV) + (V_2.T_i)(V_2.NV)}{V_1.NV + V_2.NV}$$

With $NV$ the number of voxels associated to a node and

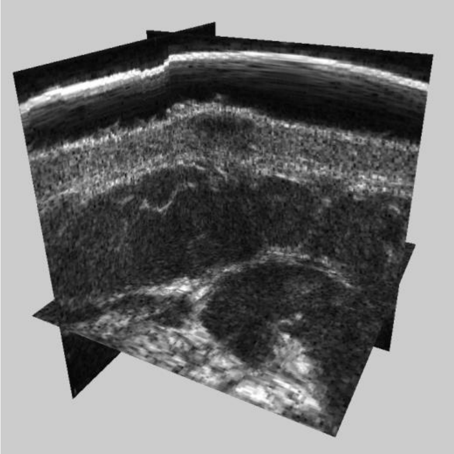
$T = \{\bar{F}, \bar{G}\}$

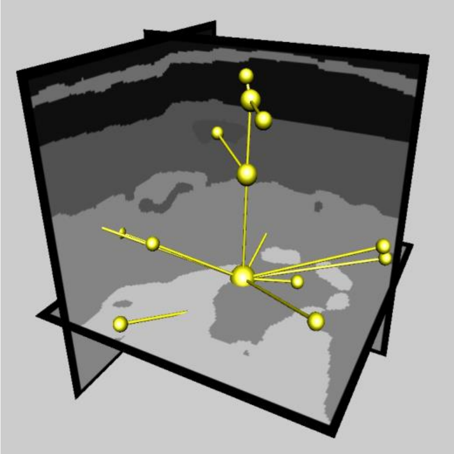
# Incremental segmentation (GUI)



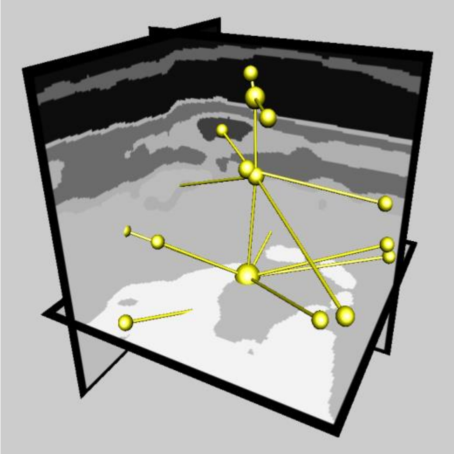



# Incremental segmentation – Explanations?

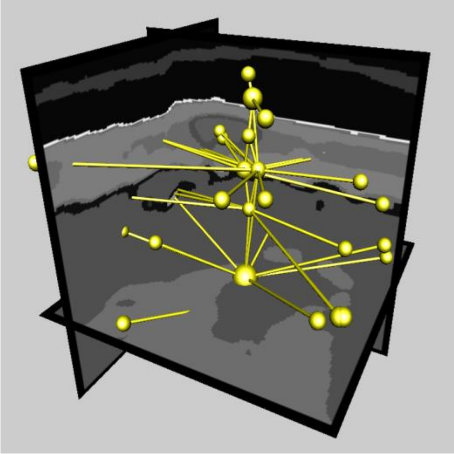




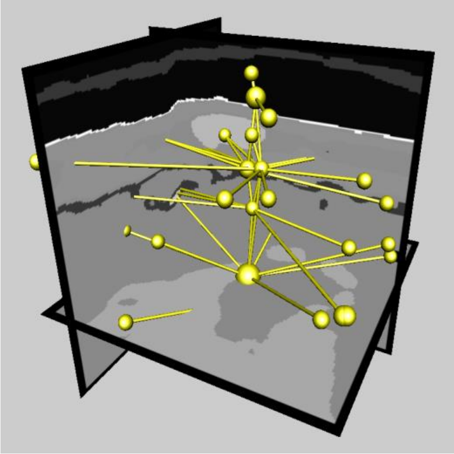
Division, K=2



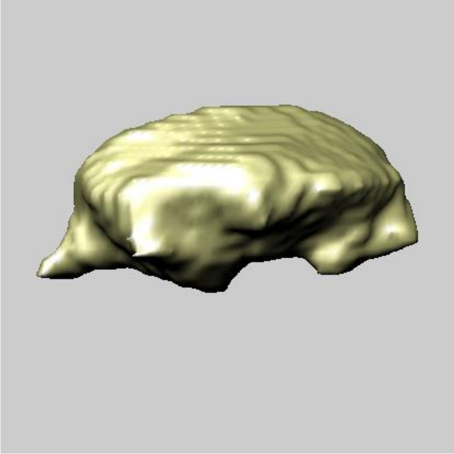
Division, K=2



Division, K=2

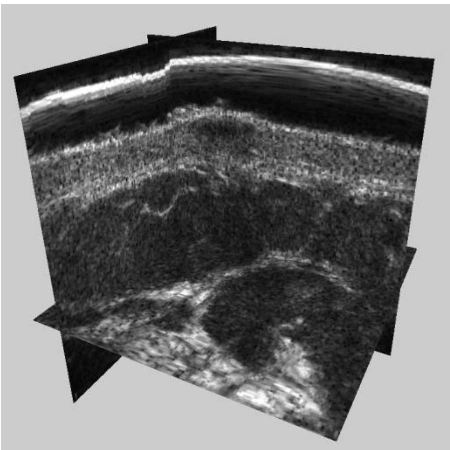


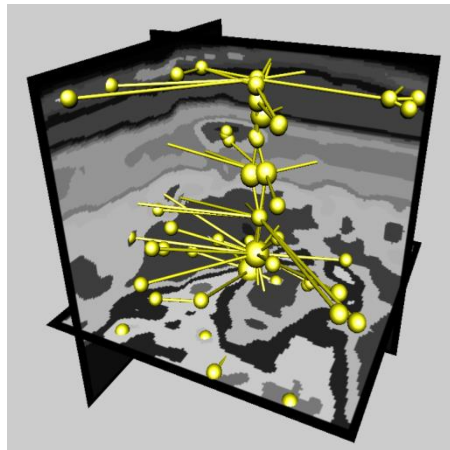
Fusion



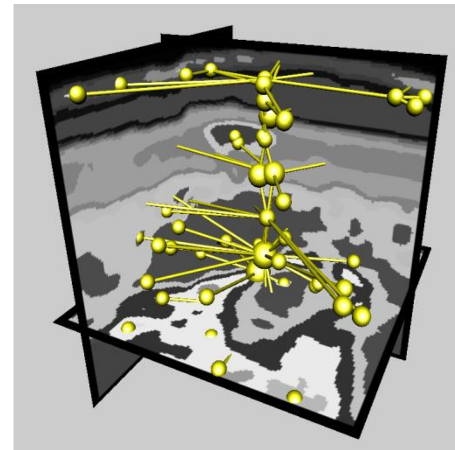
Result scénario 1

# Incremental segmentation

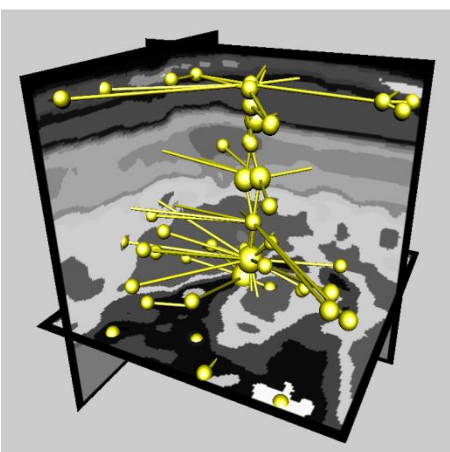




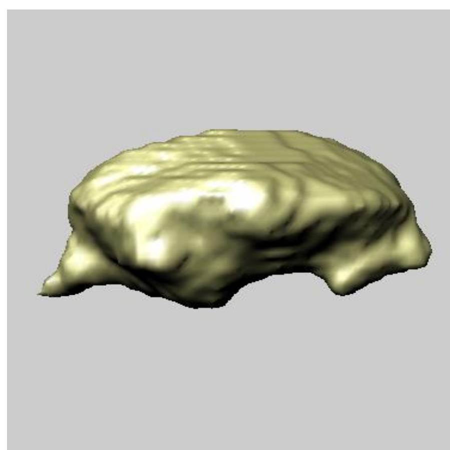
Division, K=6



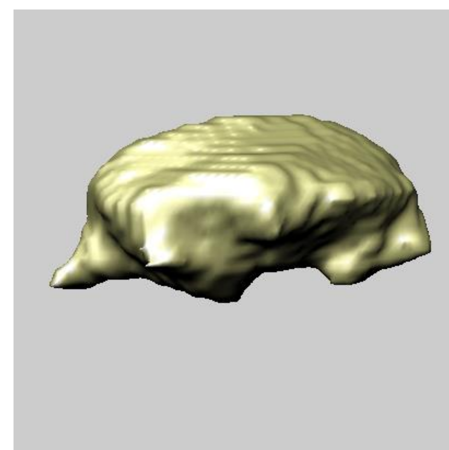
Fusion



Fusion



Result scenario 2

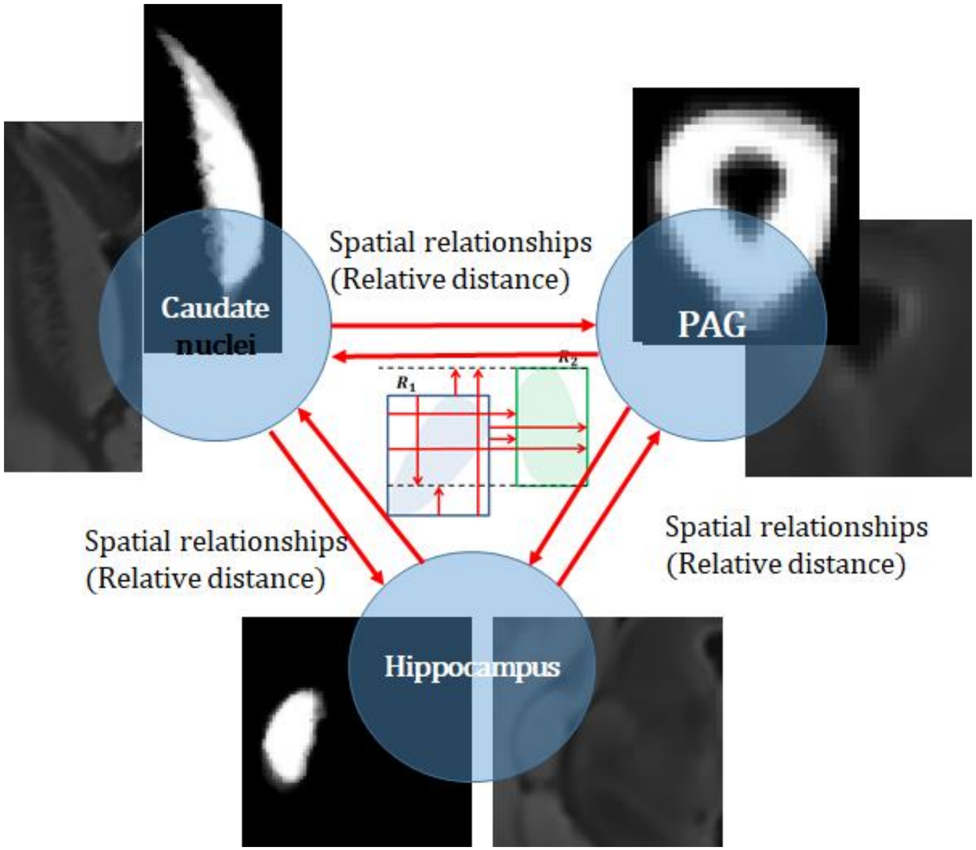


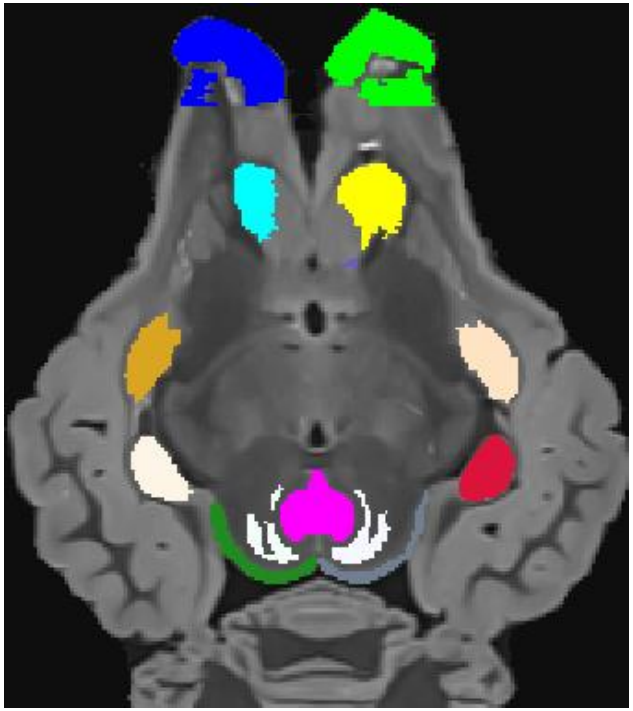
Result scenario 1

# Encoding the *a priori* information with graphs

**Segmentation of sheep brain images using local probabilistic atlases coupled with topological information (NeuroGeo) [Galisot]**

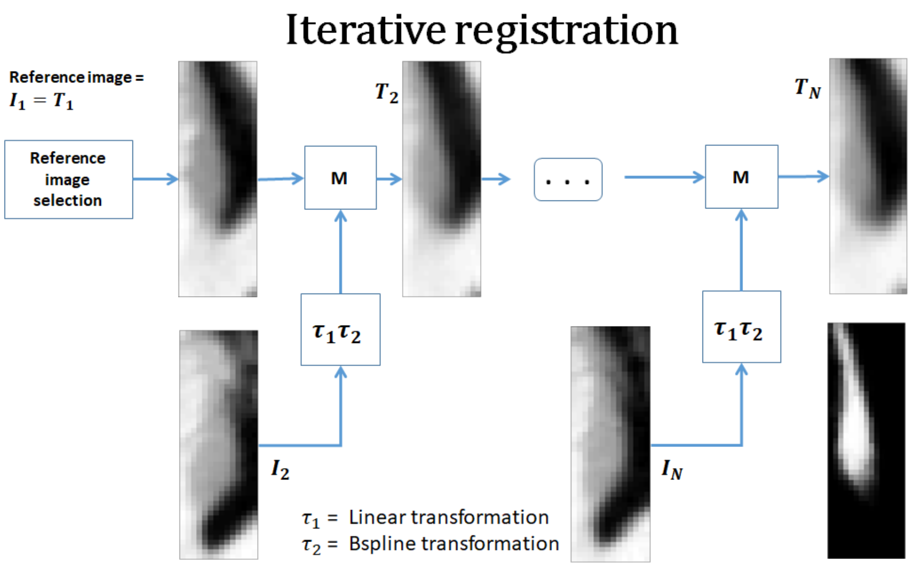
- Learning a topological graph
- Using it for better segmentation



# How to encode the ‘*a priori* information’?

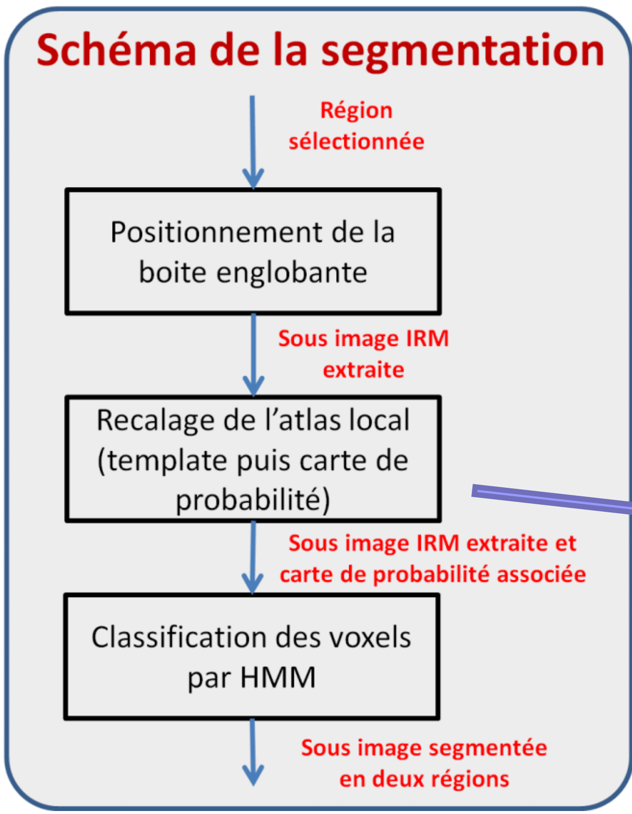
- Learning of the topological graph
  - Based on labeled images
  - Local iterative registration
  - Getting probability map
  - Getting template image

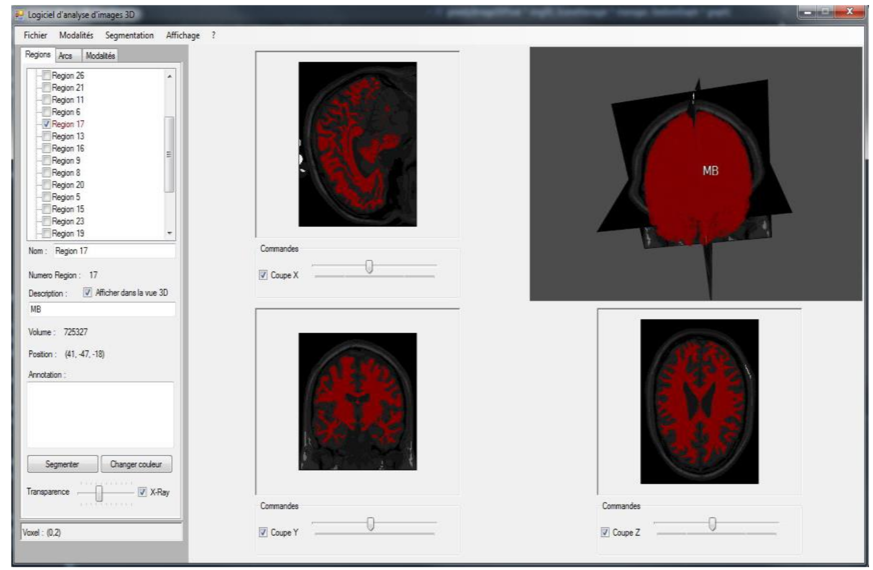


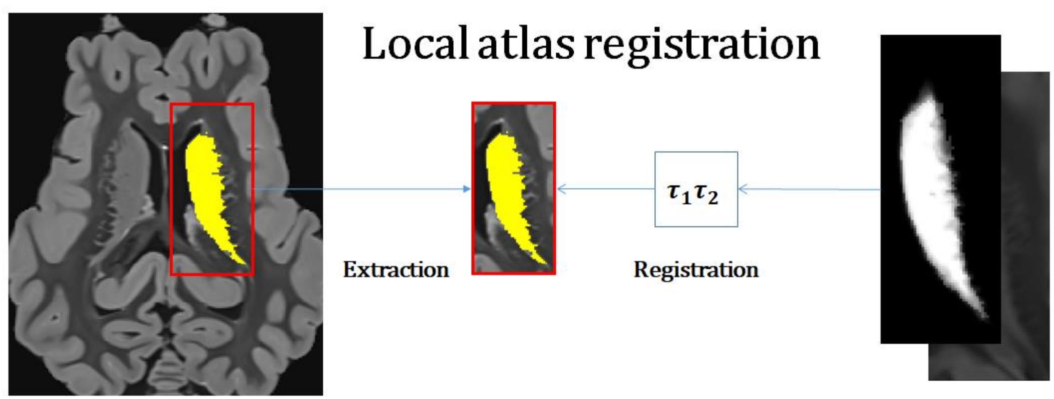

- Protocol
  - 6 images MRI, T2, 7T, ex vivo (NeuroSpin)
  - 16 regions have been manually segmented :
  - Olfactory bulbs, Caudate nucleus, PAG, Amygdalae, Superior Colliculus optic, Superior Colliculus moteur, Inferior Colliculus moteur, Septum,Hippocampus

# Using a priori information with graphs

- Incremental segmentation



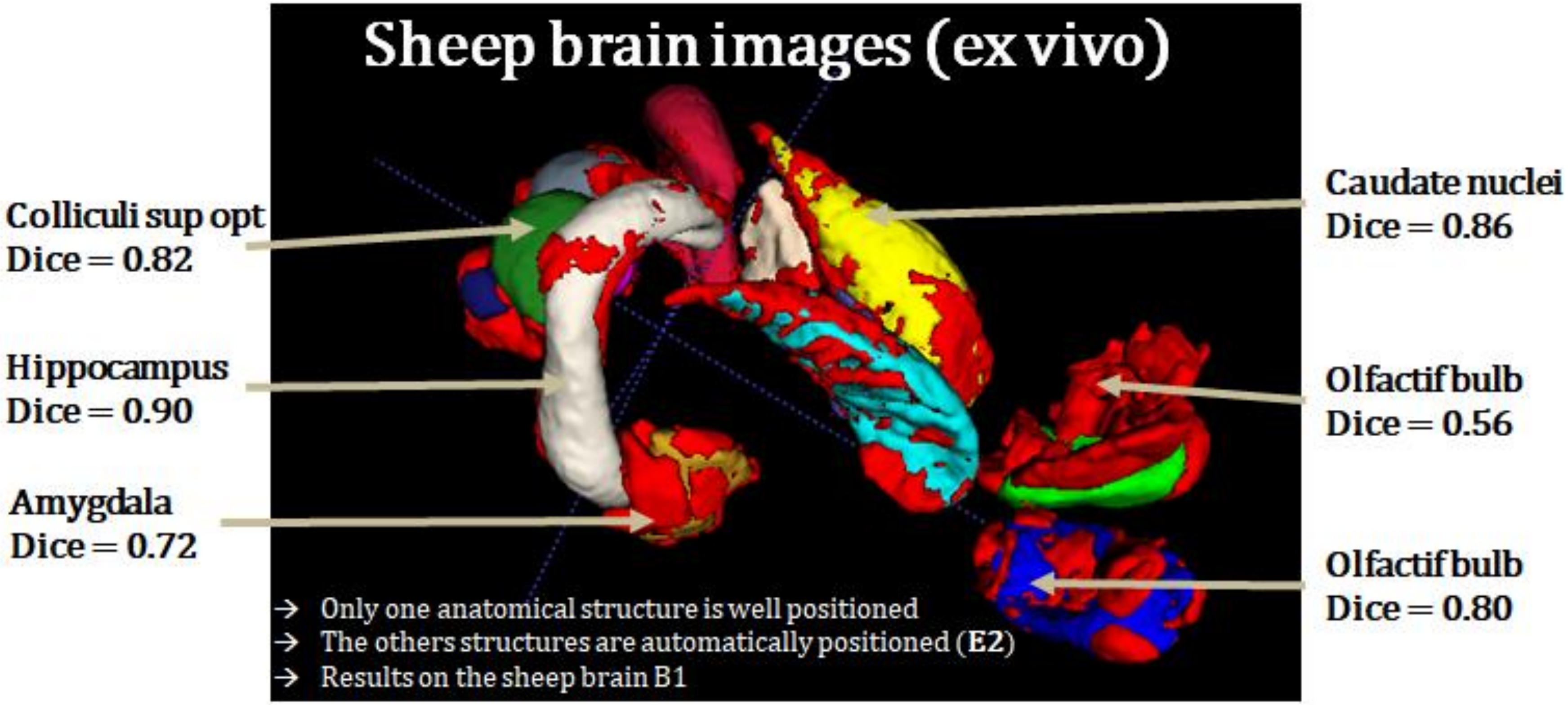




MRI image to be segmented

# Using a priori information with graphs

- Some results

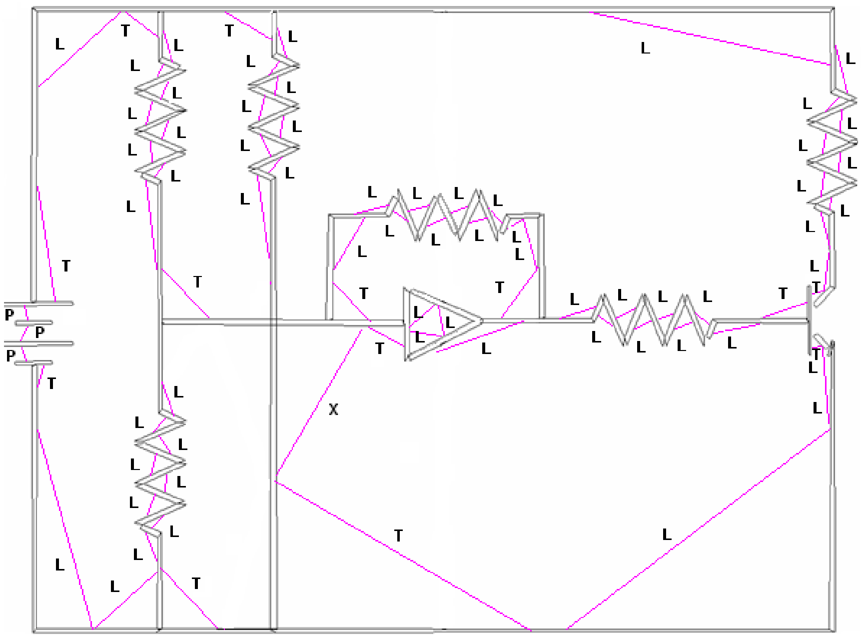


# Region Spotting with graphs

## Score computation

### A first draft

- Using heuristics to associate score to the nodes and edges
- Using Machine Learning



### Used Heuristics

H1 – Symbols are composed of small segments compared to the other parts

H2 – Segments inside a symbol are of similar length

H3 – Symbols can correspond to loops

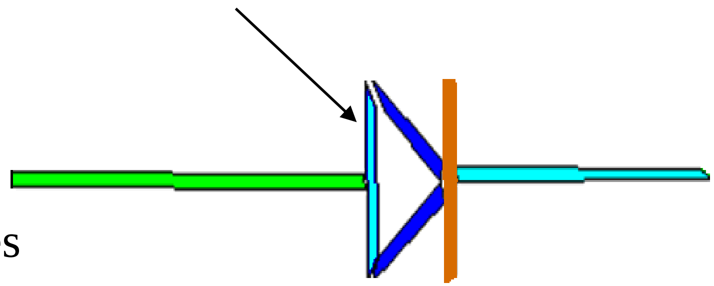
H4 – Symbols can correspond to parallel segments

H5 – Segments inside symbols are connected to maximum 3 other segment

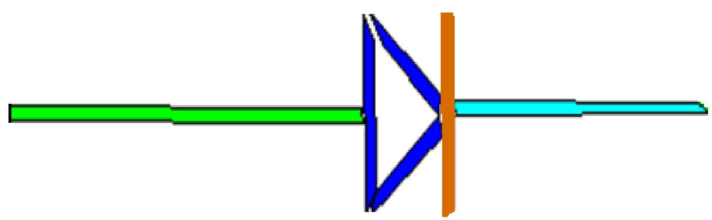
H6 – Two segments with 90° usually correspond to a symbol

# Region Spotting with graphs

## Score Propagation



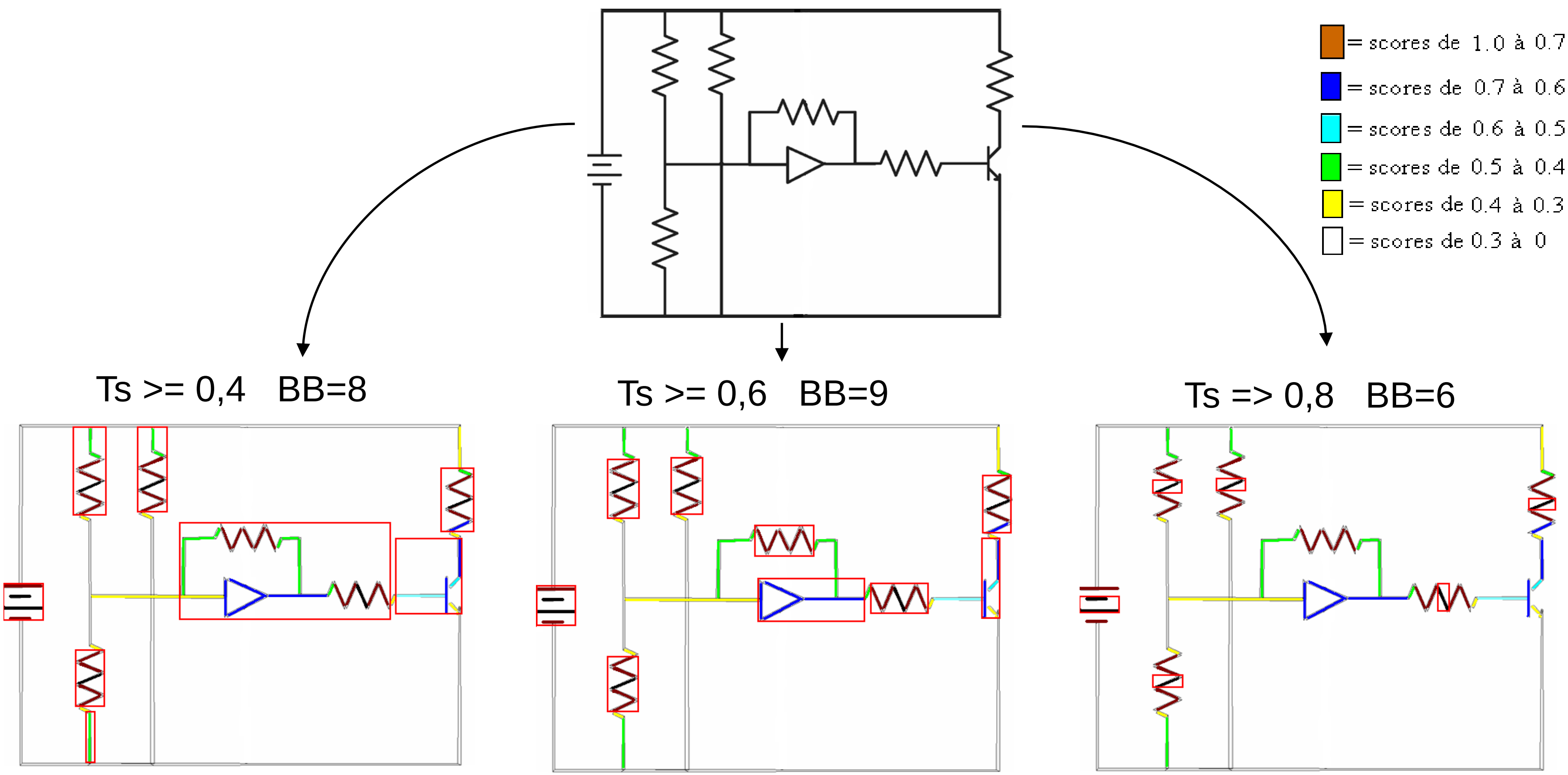
(a) Initial scores



(b) score propagation inside loops ➔ 0.7 has been propagated

# Region Spotting with graphs

## Extraction of RoI / sub-graphs using the scores



# Region Spotting with graphs

## Experimentations on different types of documents

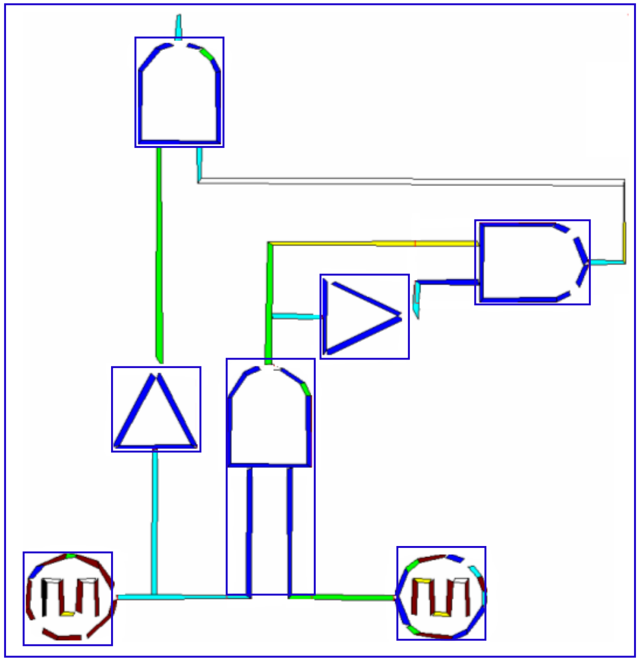
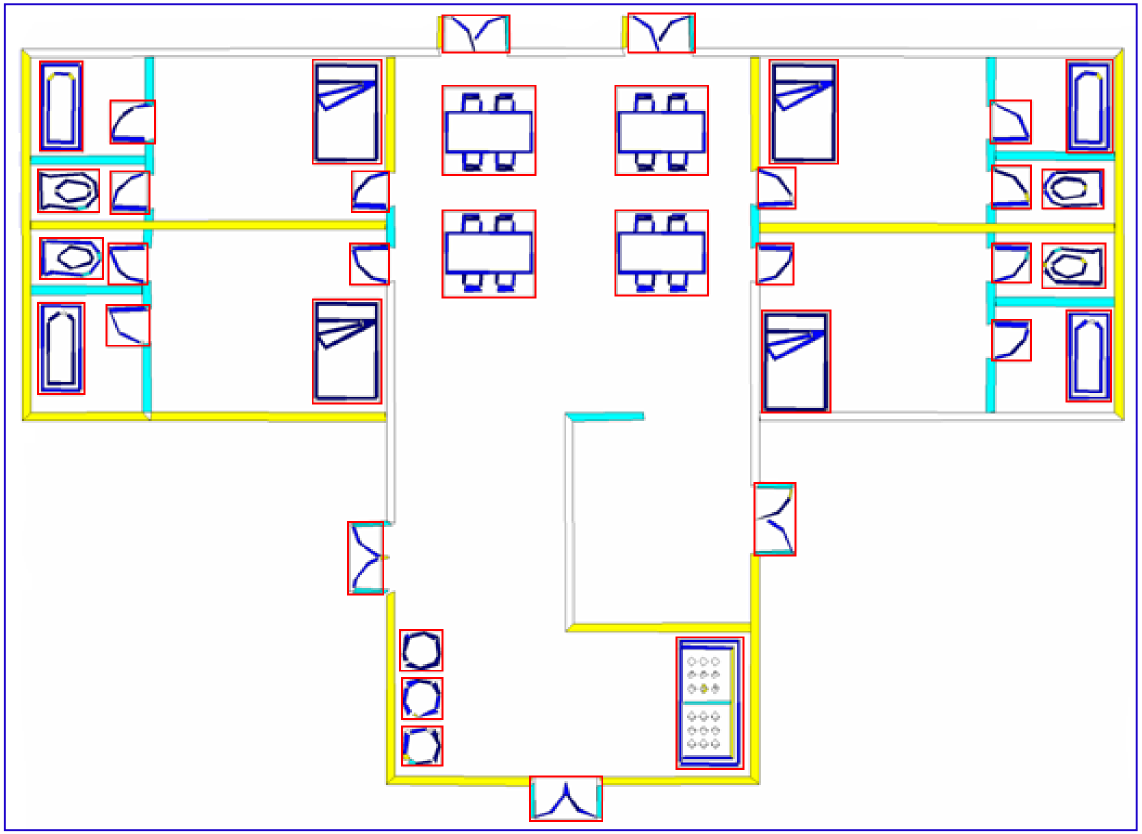


Diagramme logique Ts >= 0,6



Plan d'architecture Ts >= 0,5

# Using Graphs for Object Recognition

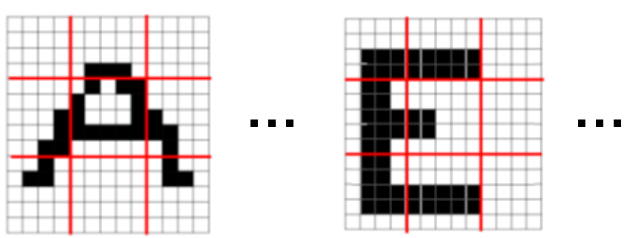
# A recall about PR mechanisms ?

**Pattern / EoC recognition** (toward Machine Learning)

**How computers can recognize objects?**

- We need a large set of (labelled) examples similar to the patterns to be recognized ➔ **a training set**
- We need a list of stable and discriminative **features** (shape, color, size,…) used to describe the patterns (labelled ones and unknown one)

V=(0,3,0,4,12,4,3,0,3) V=(6,10,0,12,4,0,10,10,0)

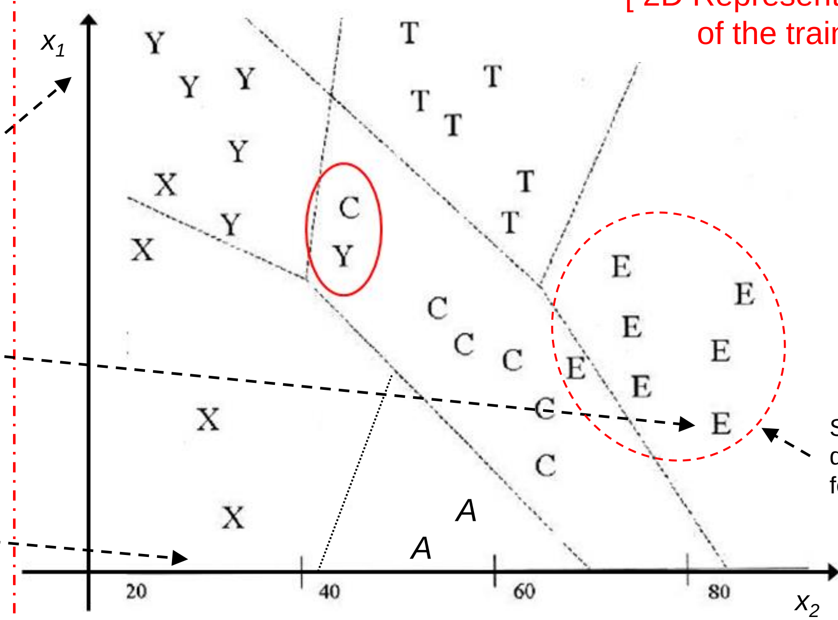



A E

[ Training set ]

1 EoC ➔ 1 Vecteur $\vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \in IR^n$

[ 2D Representation of the training set]



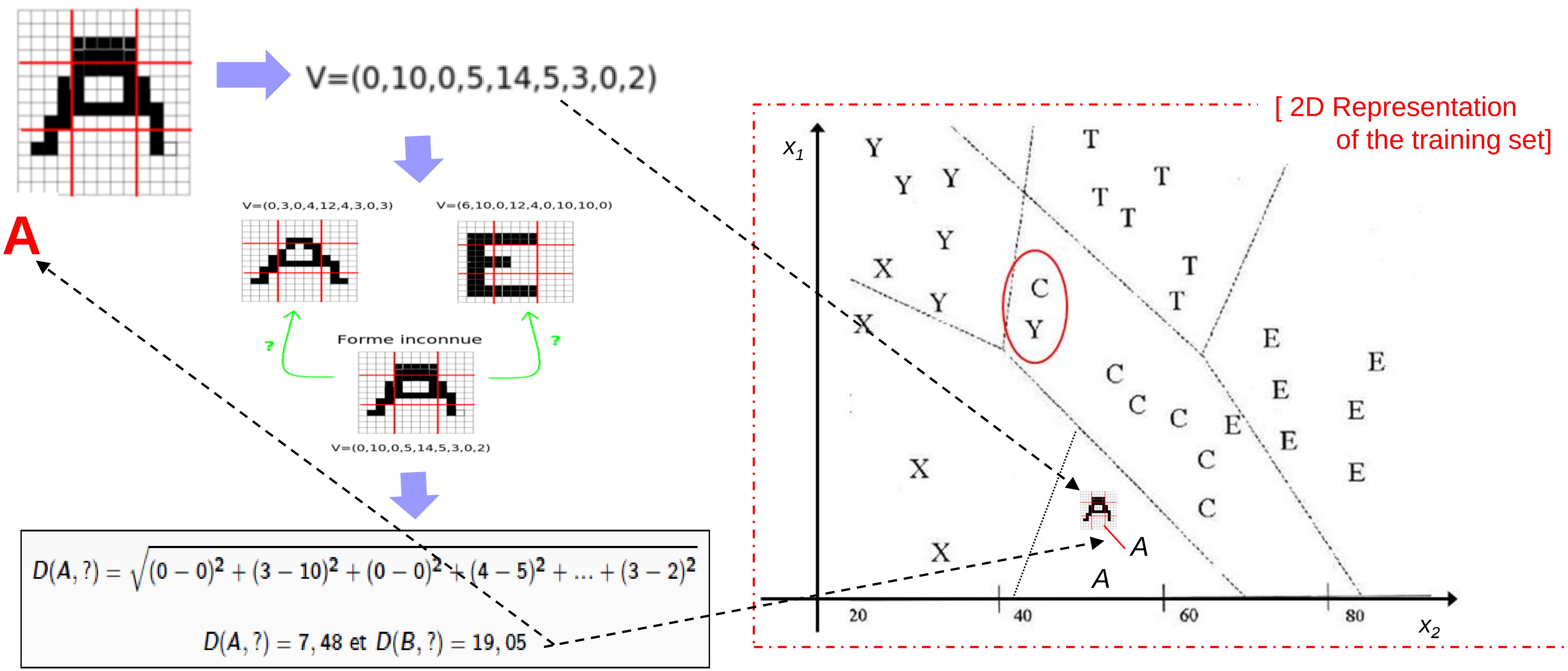

Stable and discriminative features

# A recall about PR mechanisms

## **Pattern / EoC recognition** (toward Machine Learning)

### How computers can recognize objects?

- When an unknown EoC arrives, we compute its features and compare it with the content of the training set (associated built models)

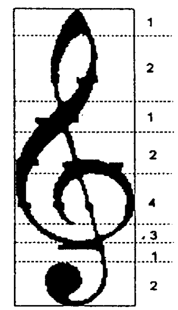


# A recall about PR mechanisms

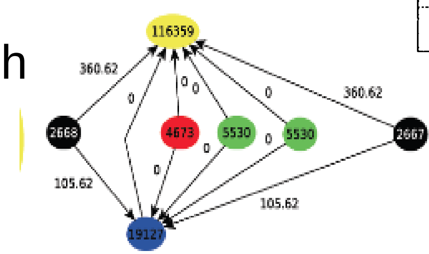
## Many possible choices and techniques

- For selection of discriminative features

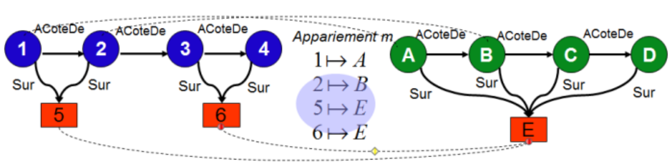
1 EoC ➔ 1 Vector $\vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ .. \\ x_n \end{bmatrix} \in IR^n$

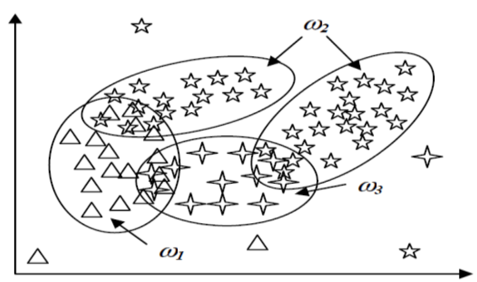


1 EoC ➔ 1 Graph

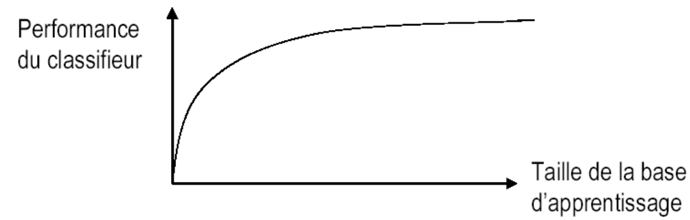



- Many Machine Learning models and tools

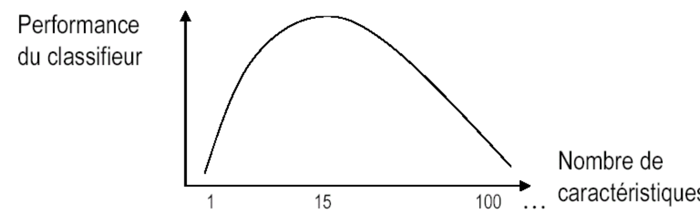


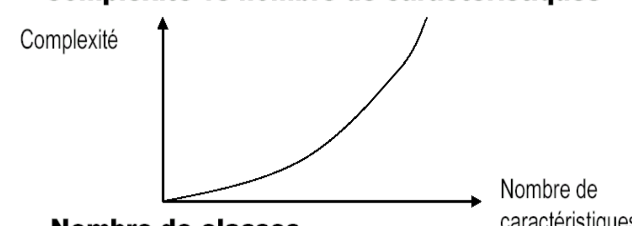

**Statistique suffisante**



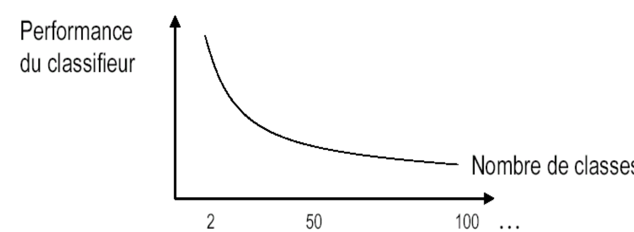

**Malédiction de la dimensionalité**




**Complexité vs nombre de caractéristiques**

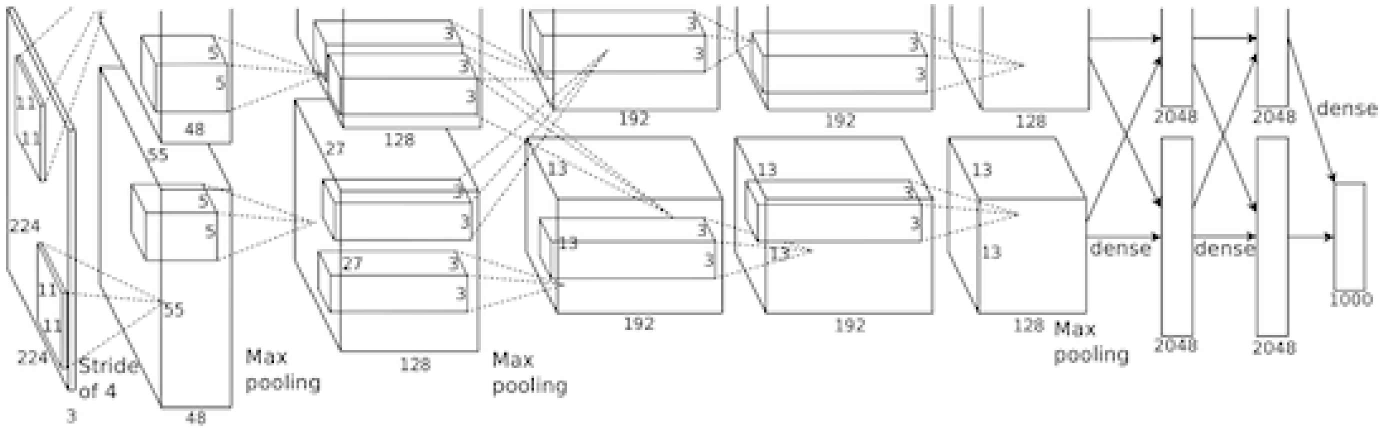


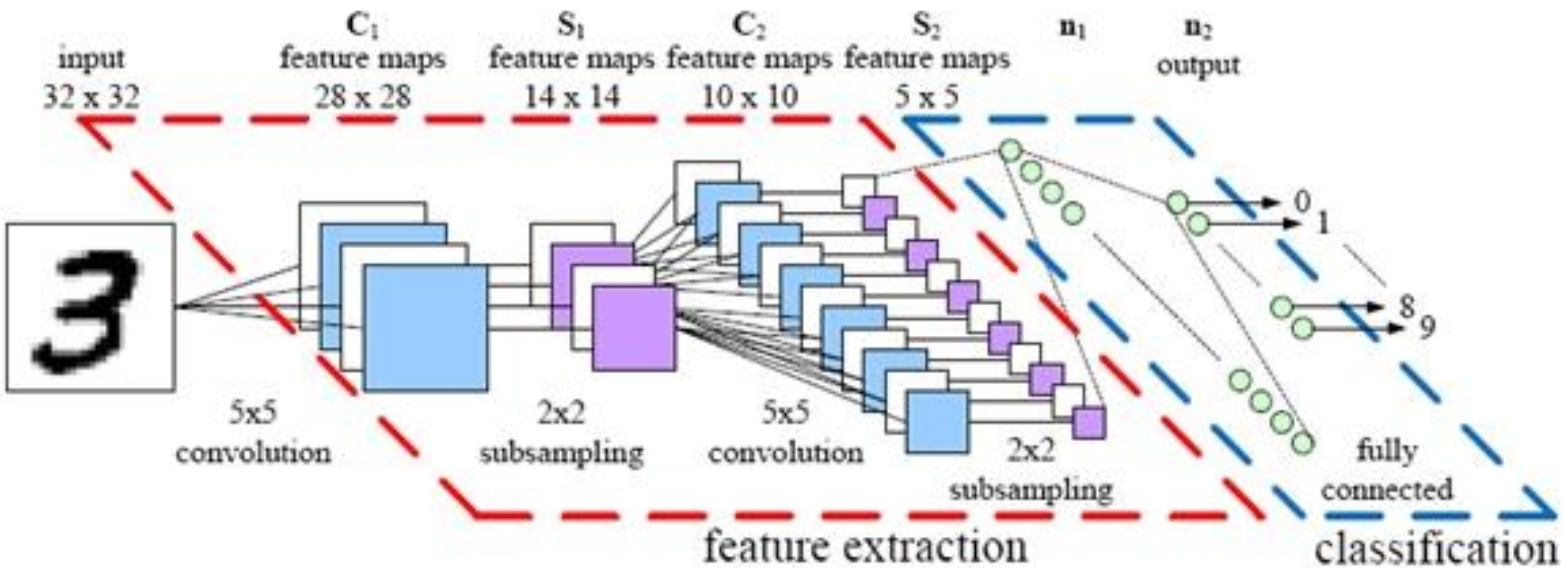

**Nombre de classes**



# A recall about PR mechanisms

## Deep Learning (Conv. Neural Net)



# Why using structural methods?

- **Statistical Methods**
  - Classes and frontiers
  - Existing statistical tools for evaluation of the quality of the chosen feature space
  - So many models and toolbox

- **Structural Methods**
  - **Taking into account the context**
  - **A matching between sub-parts as results in addition to the decision**
  - **Partial or incremental recognition**
  - **Adaptive dimensionality of the models**
  - **Multimodal Features**

  - **Computational limitations?**
  - **Learning?**

# Structural PR → Graph based matching

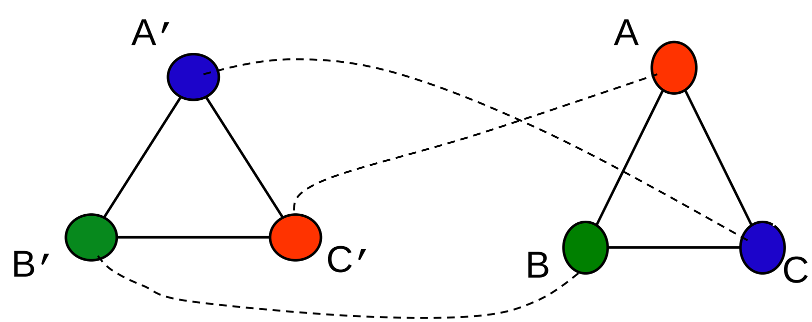
- Definition (Matching)
  - A matching between G1 = (V1;E1) and G2 = (V2;E2)
  - = a relation m $\subseteq$ V1 x V2 (u1; u2) $\in$ m
  - $\Rightarrow$ The vertex u1 is matched with the vertex u2

- Different types of matching
  - Bijective matching : cardinality = (1; 1)
  - Injective matching : cardinality = (1; 0..1)
  - Univoque matching : cardinality = (0..1; 0..1)
  - ...
  - Multivoque matching : cardinality = (0..|V2|; 0..|V1|)

**What does it mean?**

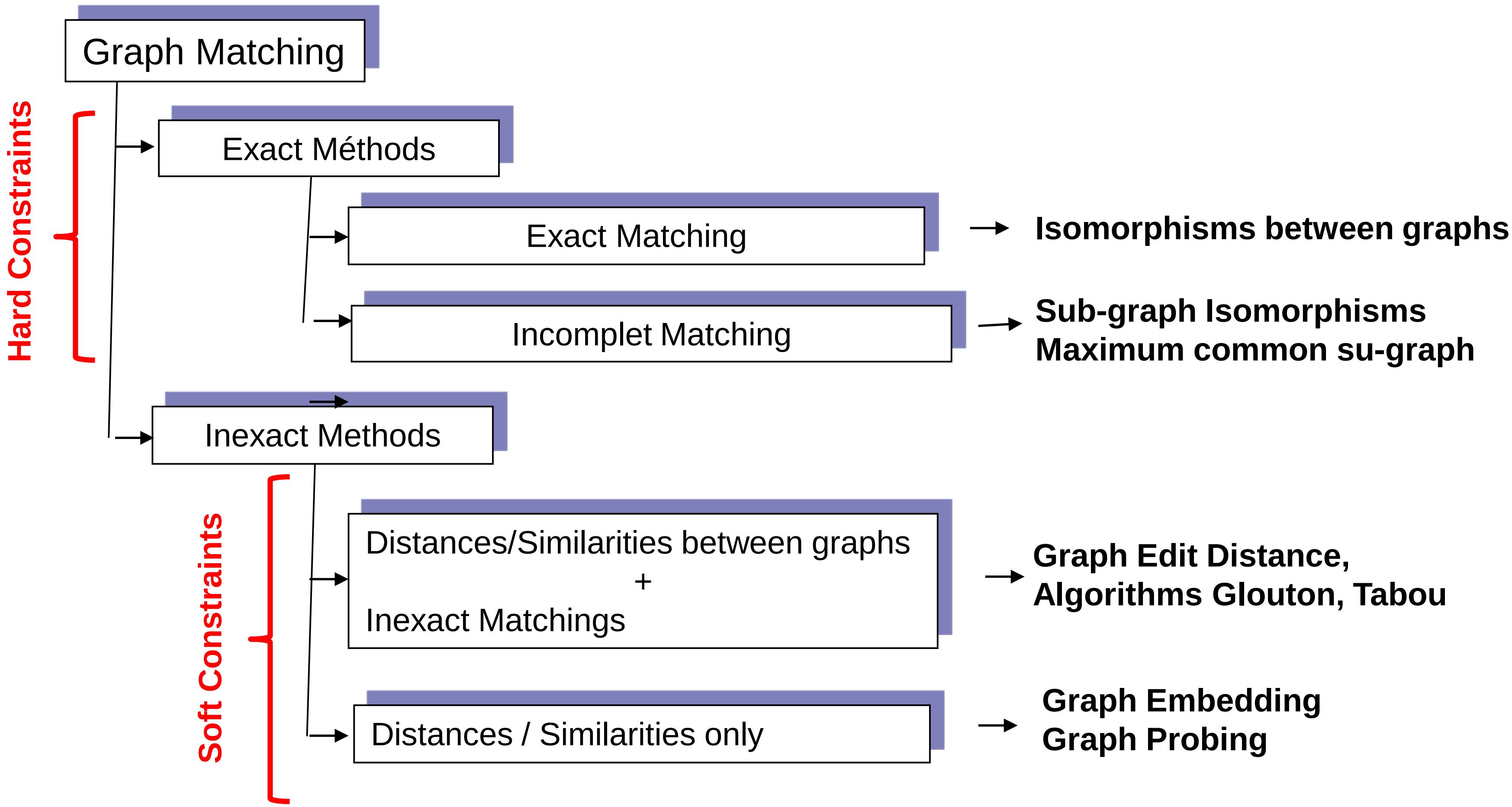
- High Complexity → Toward approximative methods !

[Solnon, 2007]



# Structural PR → Graph based matching

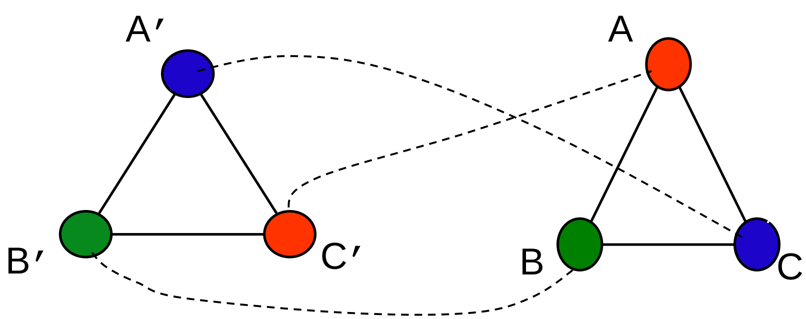
## Taking care of the attributes in addition to the graph topology



# Structural PR ➔ Graph based matching

## Univoques Matching – Hard Constraints

### Graph isomorphism problem




Model Graph $G_M$ Test Graph $G_D$

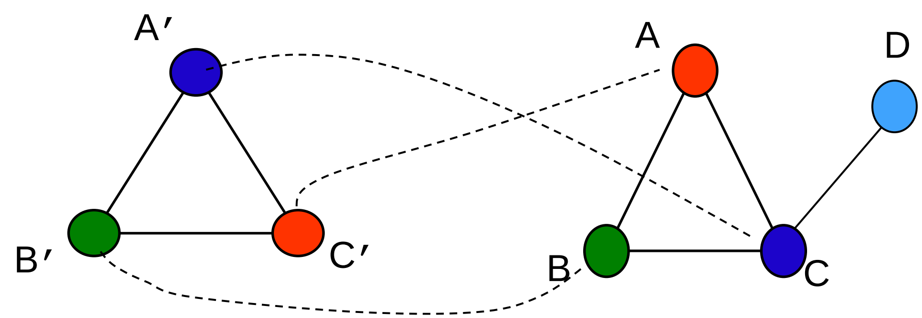
**Objective**

Bijective Matching
Hard Constraints
Possible on huge graphs

**Problem**

Not robust to noise and distorsions

### Sub-gtaph isomorphism




Test Graph $G_D$ Model Graph $G_M$

**Objective**

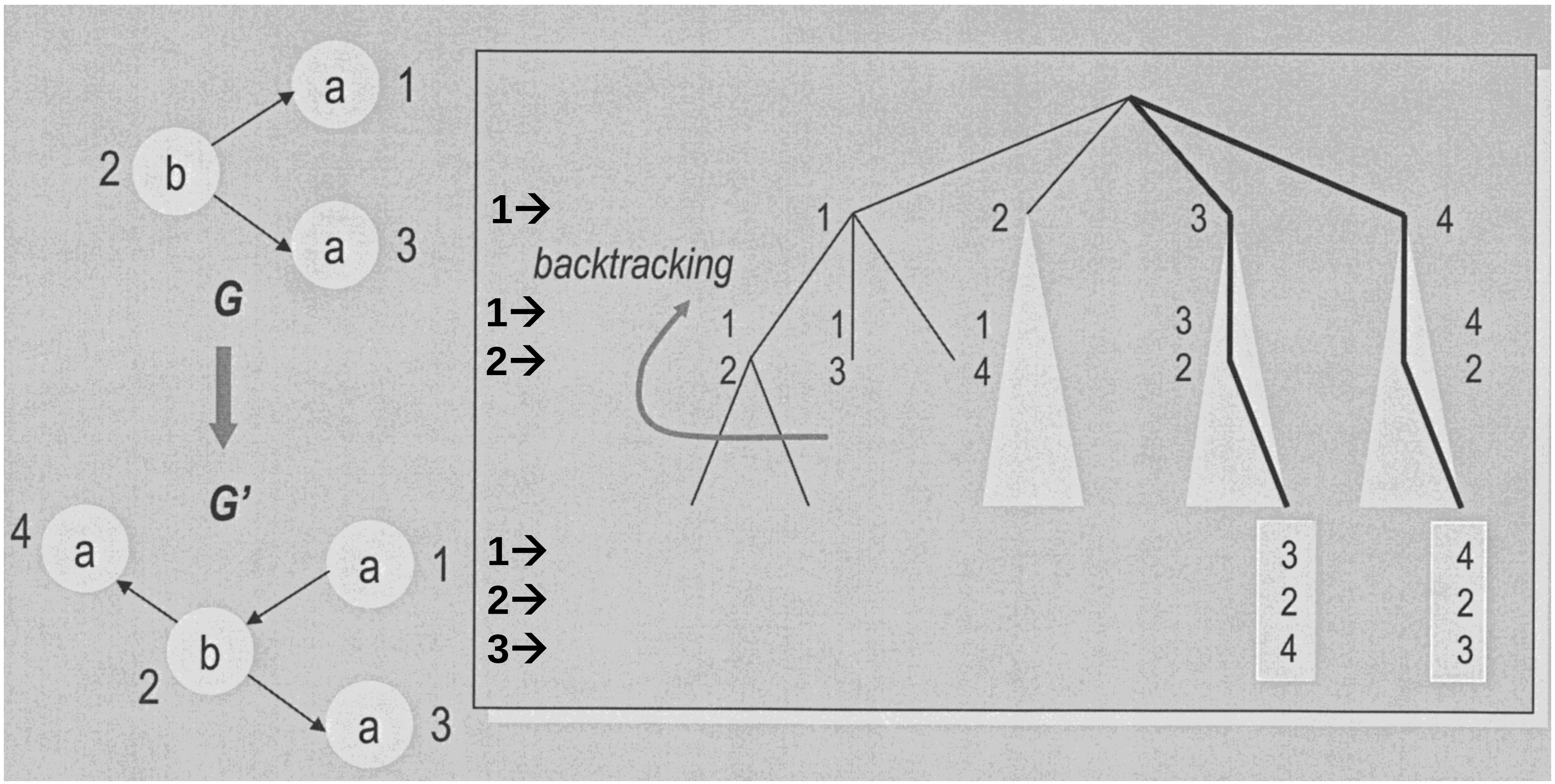
Injective Matching
Hard Constraints
NP-complete

**Problem**

Possible on medium size graphs

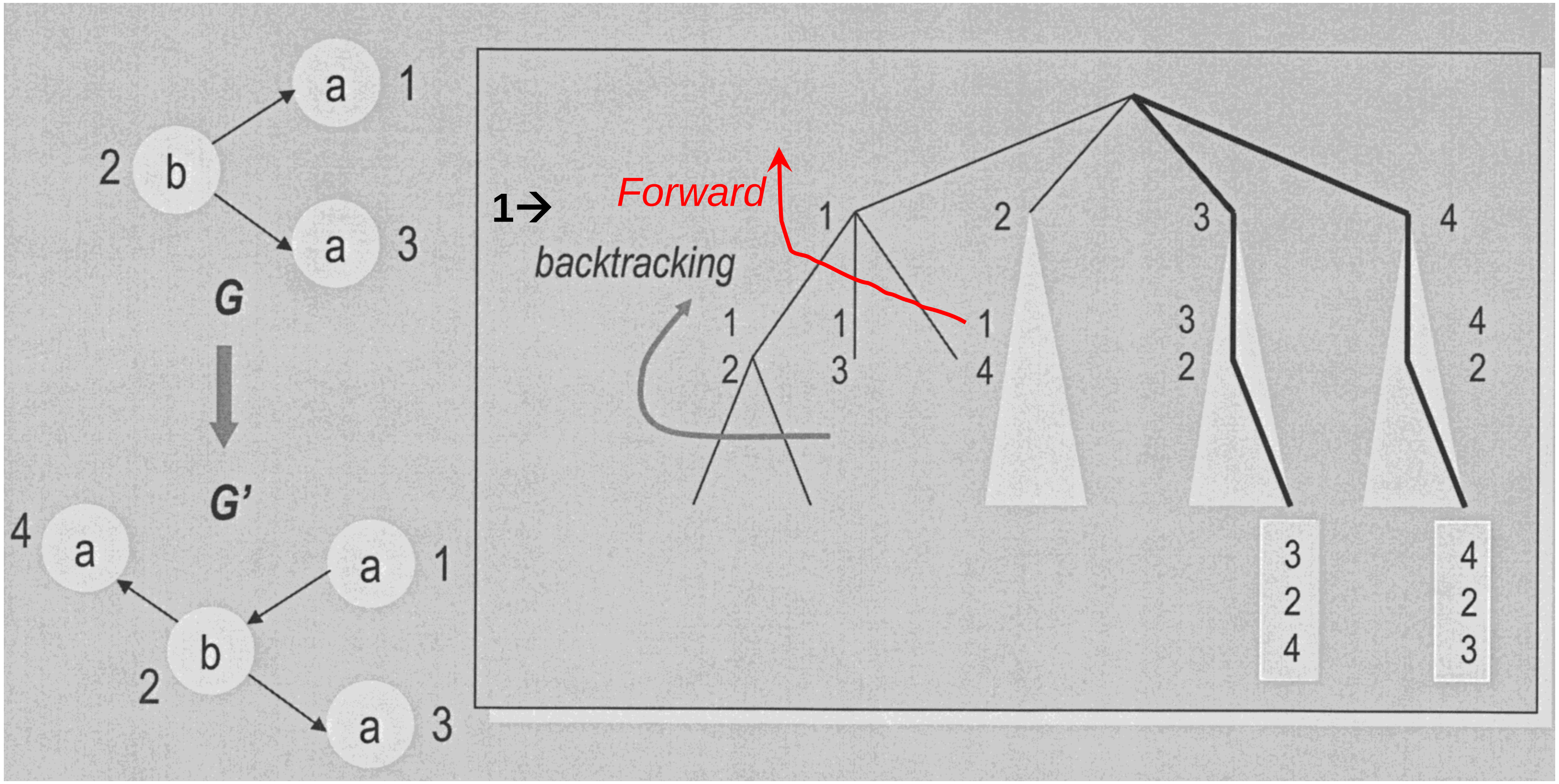
# Structural PR → Graph based matching

## Tree search Algorithms (with backtrack)



# Structural PR → Graph based matching

## Tree search Algorithms (with forward checking)



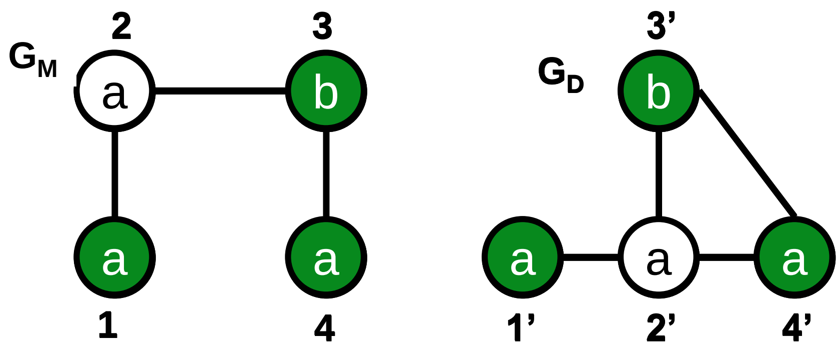

A look-ahead() checks before each association, the existence of a possible matching at the next step

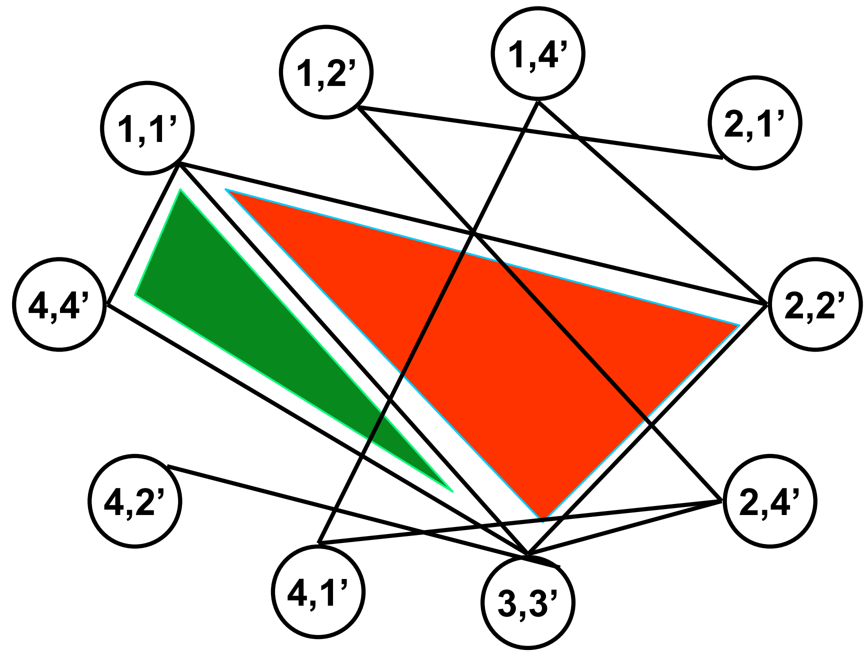
# Structural PR → Graph based matching

## Maximun commun sub-graph algorithm

### Resolution using the association graphs

(inside the association graph edges represent the topological compatibility of the matching)




Search for cliques of maximum size

C1 [(1, 1'), (2, 2'), (3, 3')]

C2 [(1, 1'), (3, 3'), (4, 4')]

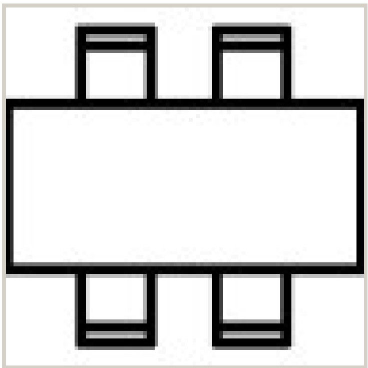
**Problem**

- Time complexity …
- Possible only for small graphs with symbolic attributes

# Structural PR ➔ Graph based matching

## Univoque Matching – Hard Constraints

## TOO HARD…



ArchitecturalH.BMP

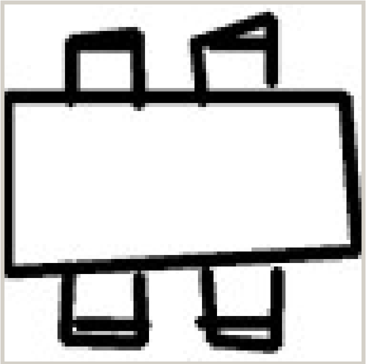
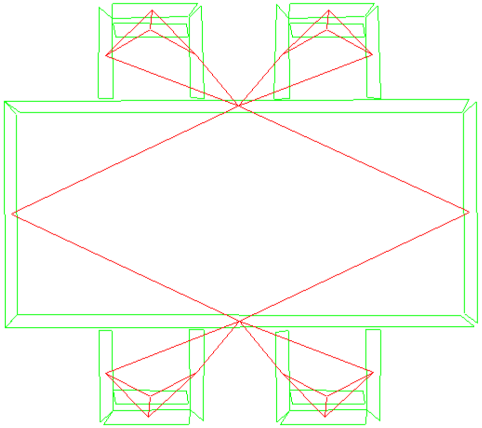
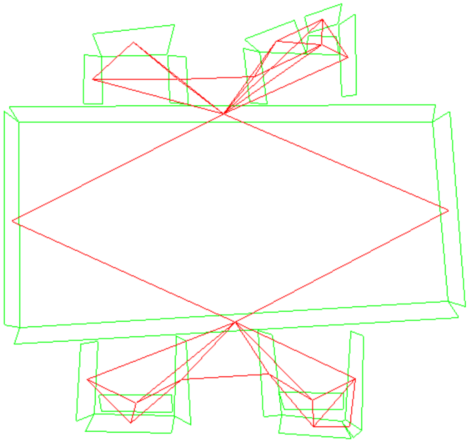


Image98.bmp

?
=





# Structural PR ➔ Graph based matching

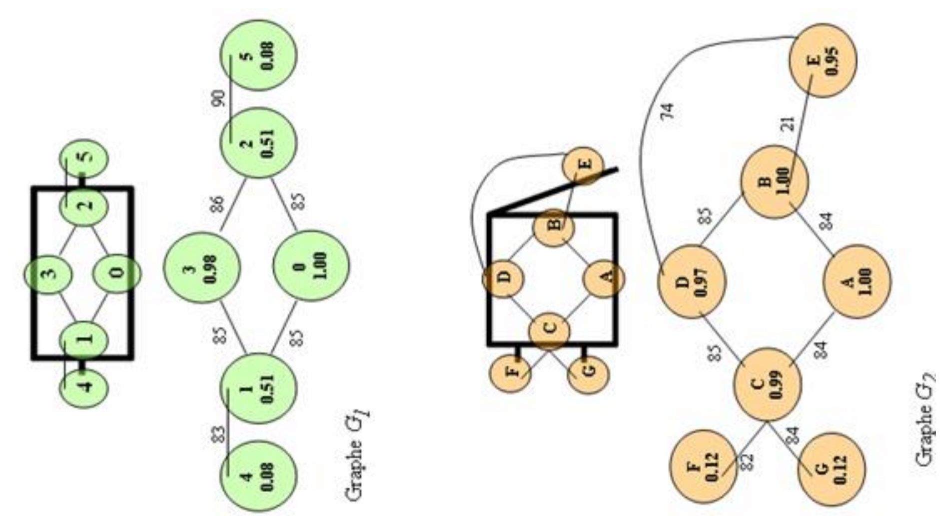
## Univoque Matching – SOFT Constraints

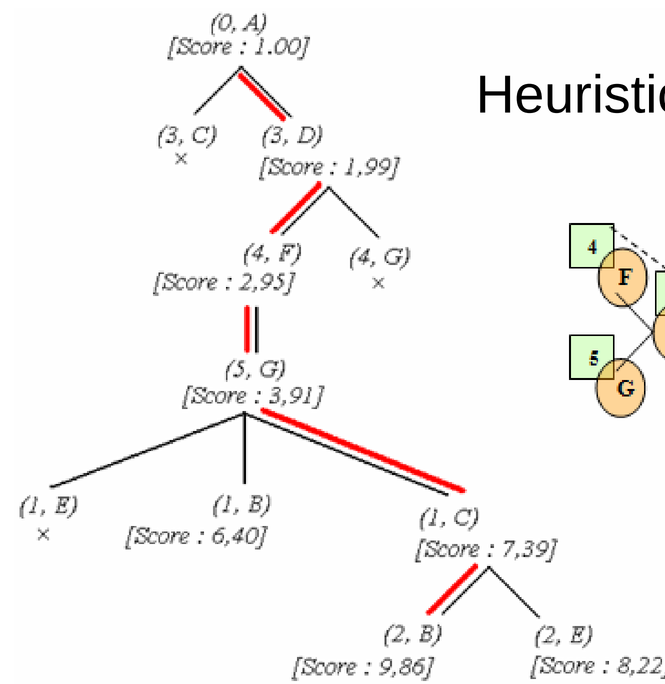
- **Soft Constraints**
  - Notion of similarity ≠ Exact matching
  - Exploration of possibilities…
    Very time consuming…
  - Similarity Matrix between nodes

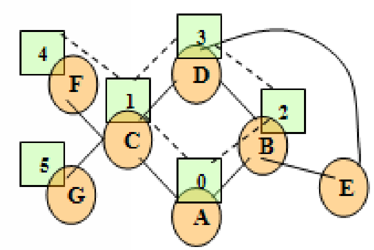
Etape-1 Etape-6 Etape-5 Etape-2 Etape-3 Etape-4

| | | A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|---|---|
| | | 1,00 | 1,00 | 0,99 | 0,97 | 0,95 | 0,12 | 0,12 |
| 0 | 1,00 | 1,00 | 1,00 | 0,99 | 0,97 | 0,95 | 0,12 | 0,12 |
| 1 | 0,51 | 0,51 | 0,51 | 0,52 | 0,54 | 0,56 | 0,61 | 0,61 |
| 2 | 0,51 | 0,51 | 0,51 | 0,52 | 0,54 | 0,56 | 0,61 | 0,61 |
| 3 | 0,98 | 0,98 | 0,98 | 0,99 | 0,99 | 0,97 | 0,14 | 0,14 |
| 4 | 0,08 | 0,08 | 0,08 | 0,09 | 0,11 | 0,13 | 0,96 | 0,96 |
| 5 | 0,08 | 0,08 | 0,08 | 0,09 | 0,11 | 0,13 | 0,96 | 0,96 |

Heuristics ☹



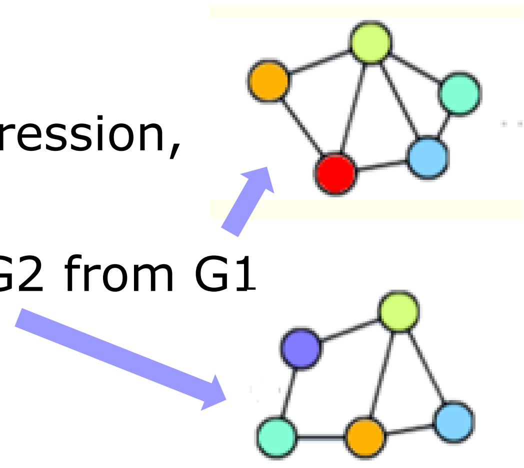



# Structural PR ➔ Graph based matching

## Univoque Matching – Soft constrainsts

### Graph Edit distance [Bunke 99]



**Cost** : associated to transformations (Insertion, suppression, substitution – edges and nodes)

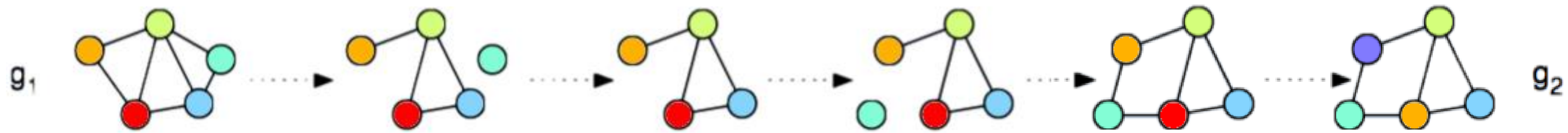
**Edit Path** : set of needed transformations to obtain G2 from G1

**Global Error** : Sum of all the elementary costs

**Objective** : Search for the minimal cost edit path

Let $G_1 = (V_1, E_1, L_{V_1}, L_{E_1}, \mu_1, \zeta_1)$ and $G_2 = (V_2, E_2, L_{V_2}, L_{E_2}, \mu_2, \zeta_2)$ be two graphs, the graph edit distance between $G_1$ and $G_2$ is defined as:

$$d_{plain}(g_m, g_t) = \min_{e_1, \cdots, e_k \in \gamma(g_1, g_2)} \sum_{i=1}^{k} c(e_i)$$



# Structural PR ➔ Graph based matching

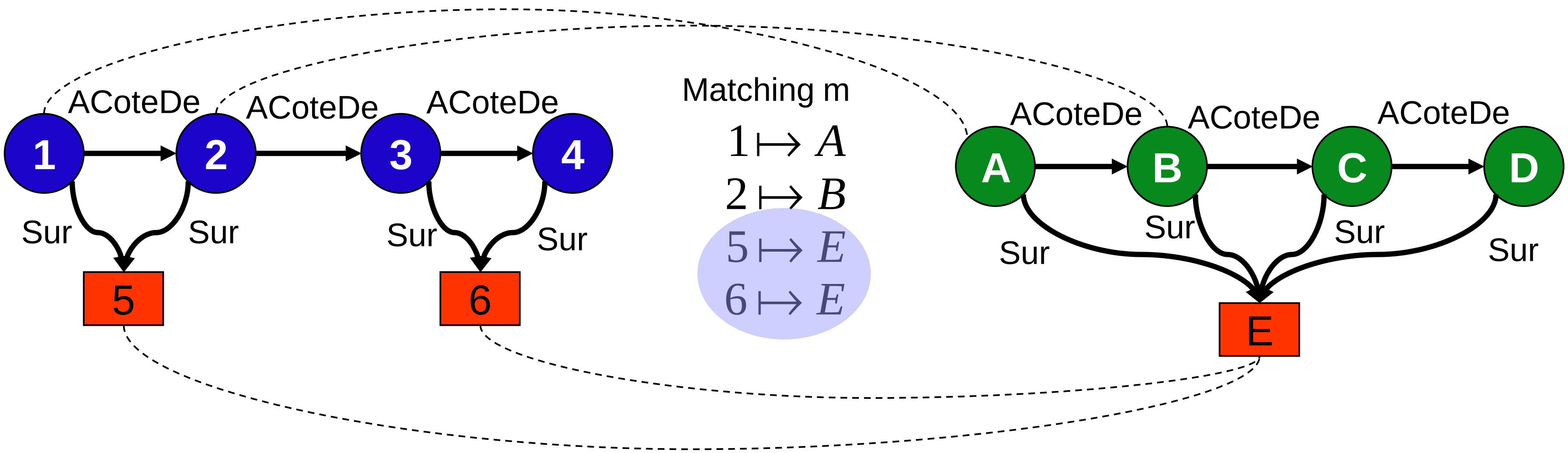
## Multivoque Matching – Soft constrainsts

## Evoluted version of the Graph Edit distance

Univoque Matching ➔ each node of G1 can be matched with only one node of G2

**New version of GED with additional possible transformations**

**MERGE & SPLIT → Multivoque Matching …**




**[Champin / Solnon] (2003-2005)**

# Structural PR → Graph based matching

## Multivoque Matching – Soft constrainsts

### Problem : Definition of the similarity measure and edit costs [Qureshi03]

$$Sc_{Mp} = \left[ \underbrace{\sum_{i=1}^{m}\left(1-\Delta V_i\right)}_{\text{Node to Node similarity}} + \underbrace{\sum_{j=1}^{n}\left(1-\Delta E_j\right)}_{\text{Edge to Edge similarity}} - \underbrace{\left(\sum_{i=1}^{k}\omega_i + \sum_{j=1}^{\ell}\omega'_j\right)}_{\text{Penalties for multiple matchings (splits)}} \right]$$

**Matching Exploration → a very combinatory problem !**

Goal = Finding $m \subseteq V_1 \times V_2$ maximising

$score(m) = f\,(G_1 \cap_m G_2) - g(splits(m))$

Problem NP-difficile → $2^{|V1|.|V2|}$ combinaisons

**Résolution by a complete search ?**

Structuring the search space with lattices...

...but the score function is not monotonous….

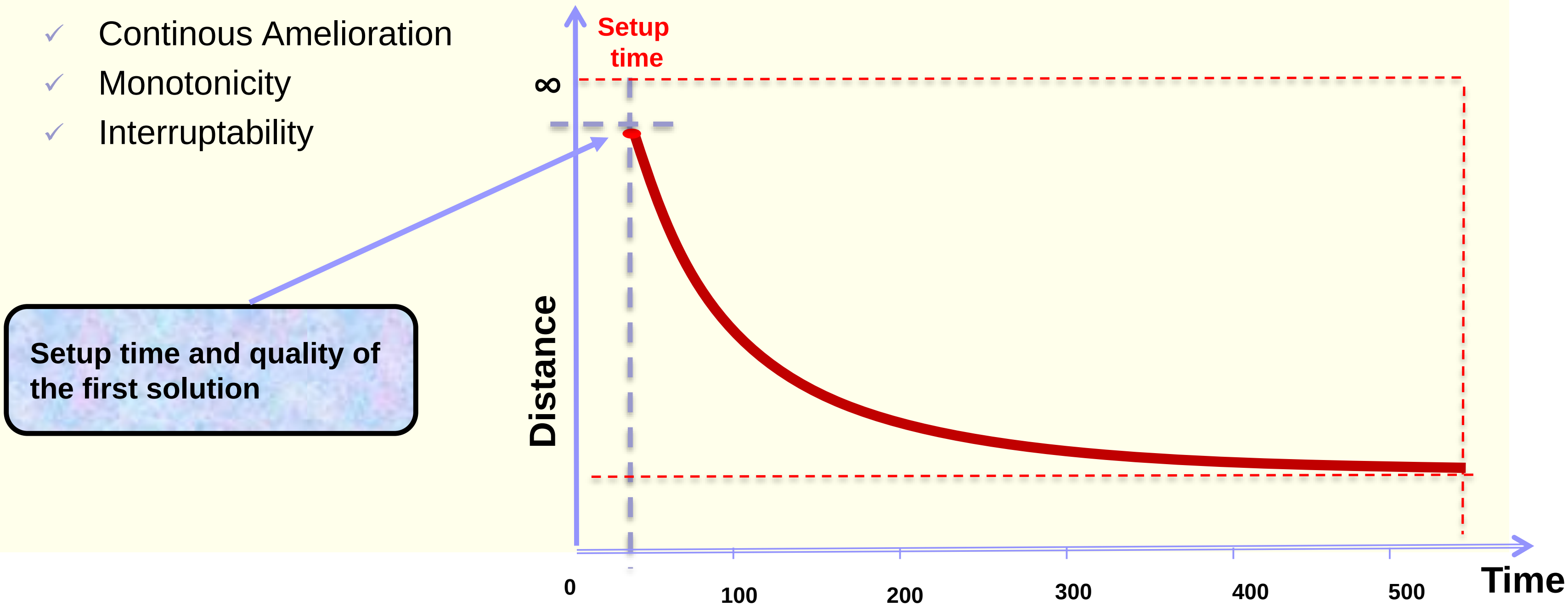
**Limited to very small graphs (10 nodes)**

**Using heuristics approaches (not exact)**

# Structural PR → Graph based matching

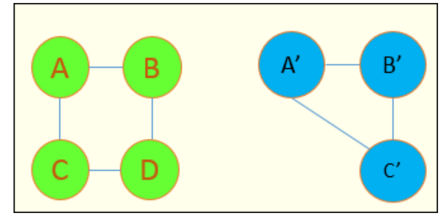
## Scalability of GM Algorithms: Anytime GM [Abu-Aisheh16]

- **Idea : Production of a succession of solutions with improved quality along time**
- **Anytime algorithms** [Zilberstein 1996]
  - ✓ Continous Amelioration
  - ✓ Monotonicity
  - ✓ Interruptability



## Anytime Depht First GED

- Tree search algorithm (A*)
- Low memory consomption (DF)
- Optimisation of the computation (with preprocessing)
- Selection of the first solution (UB)




**Cost Matrices computation**

❖ **To prepare the data**

$$C_e \text{ and } C_V = \left[\begin{array}{cccc||cc} c_{1,1} & \cdots & \cdots & c_{1,m} & c_{1\leftarrow\epsilon} & c_{1\rightarrow\epsilon} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ c_{n,1} & \cdots & \cdots & c_{n,m} & c_{n\leftarrow\epsilon} & c_{n\rightarrow\epsilon} \\ \hline c_{\epsilon\rightarrow 1} & \cdots & \cdots & c_{\epsilon\rightarrow m} & \infty & \infty \\ c_{\epsilon\leftarrow 1} & \cdots & \cdots & c_{\epsilon\leftarrow m} & \infty & \infty \end{array}\right]$$

**Preprocessing** → **Branch-and-Bound**

**Bipartite Graph Matching**

❖ **To prune the search tree**

$$d_{UB} = d_{p_{min}}$$

$$p_{min} = \{\underbrace{u_1 \rightarrow v_{i_1}}_{\omega_1}, \underbrace{u_2 \rightarrow v_{i_2}}_{\omega_2}, \ldots, \underbrace{u_k \rightarrow v_{i_k}}_{\omega_k}\}$$

❖ **To start with the promising vertices of G1**

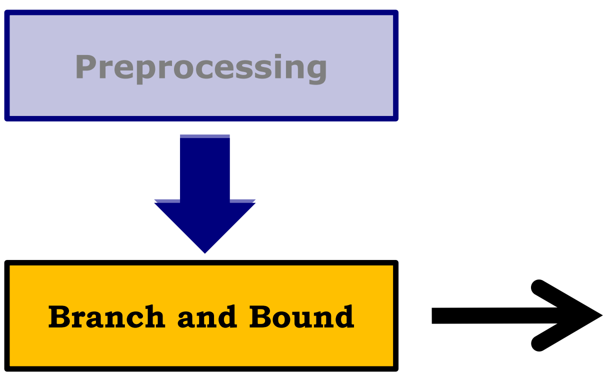
Sort $V_1$ in ascending order according to $\omega$'s

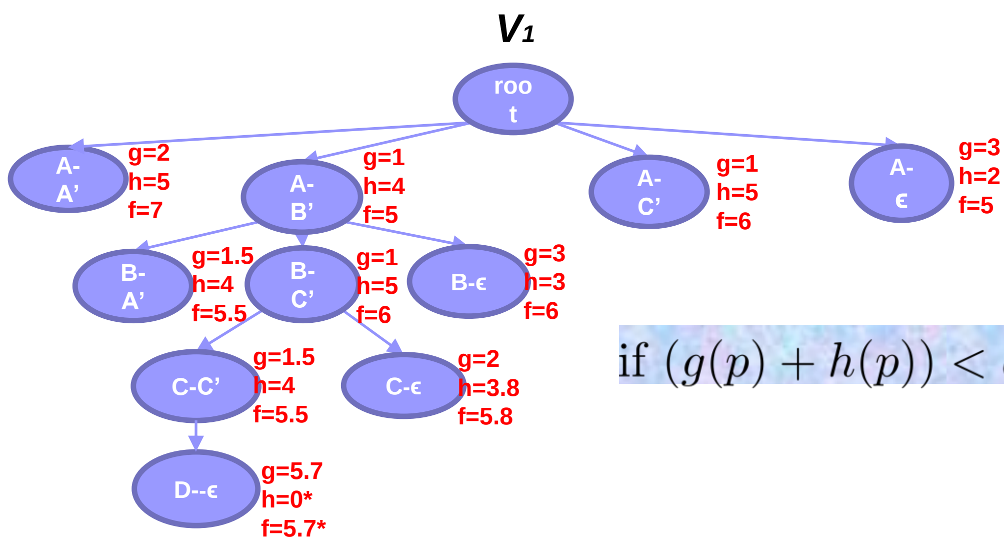
$$V_1 = \{u_k, u_2, u_1, \ldots.\}$$

**Anytime Depht First GED**

- Selection of the 1st solution (UB) ➔ Setup time ➔
- Speed of the evolution
- Stop criteria ?
- …

$$\begin{cases} d_{UB} = +oo \\ \text{or} \\ d_{UB} = d_{p_{min}} \end{cases}$$



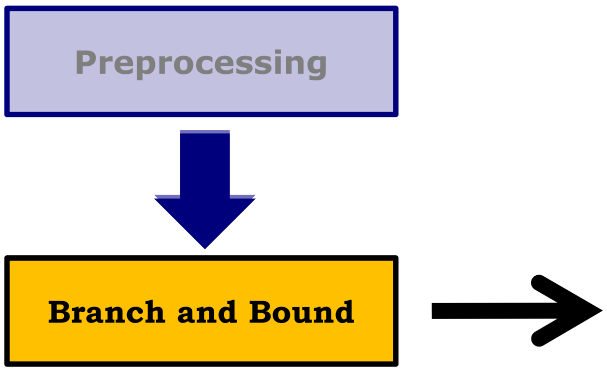



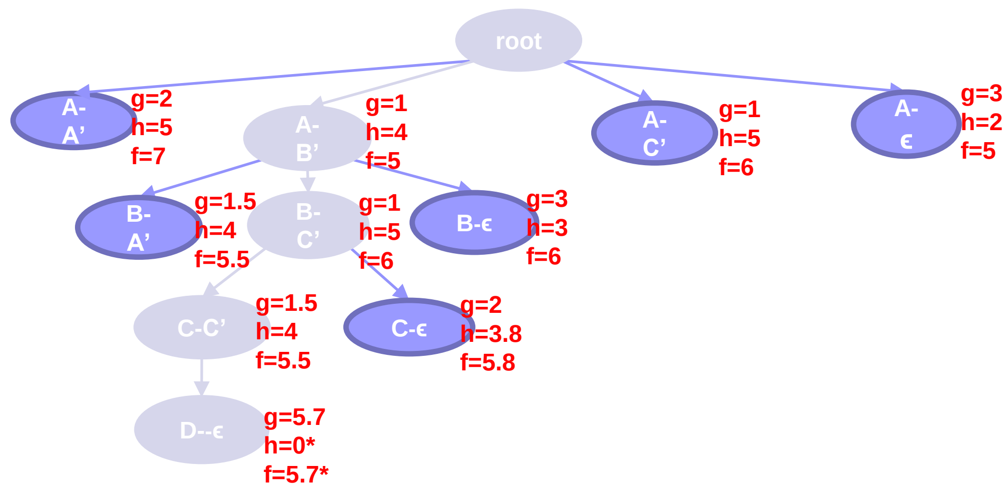

$$\text{if } (g(p) + h(p)) < d_{UB}$$

**Anytime Depht First GED**

- Selection of the 1st solution (UB) ➔ Setup time ➔
- Speed of the evolution
- Stop criteria ?
- …

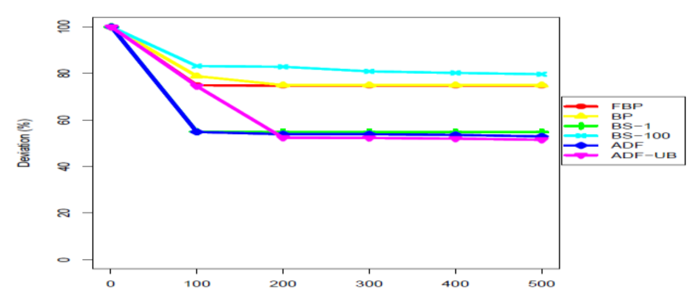
$$\begin{cases} d_{UB} = +oo \\ \text{ou} \\ d_{UB} = d_{p_{min}} \end{cases}$$




**Results**

- Trade-off between time and quality
- Shape of the curve (setup, …)
- Depening of the graph types
- ….



# Structural PR ➔ Graph based matching

## Graph Probing and Embedding

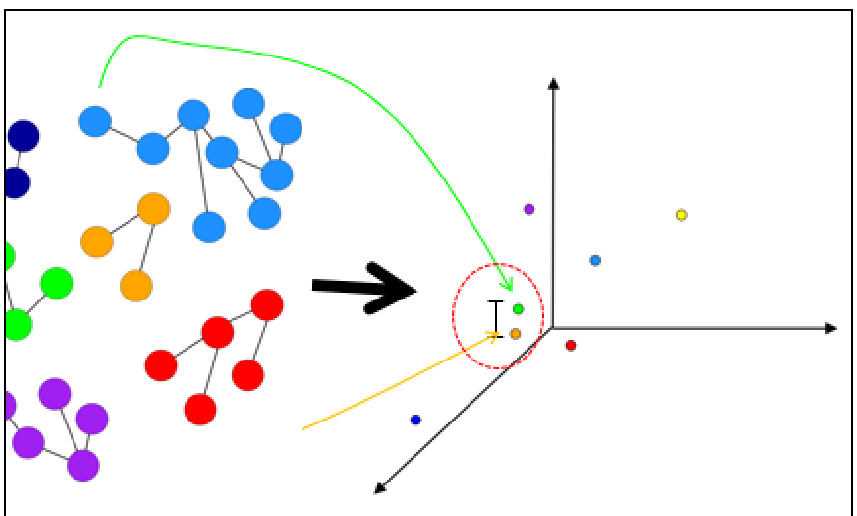
**From graph space back to Vector space…**

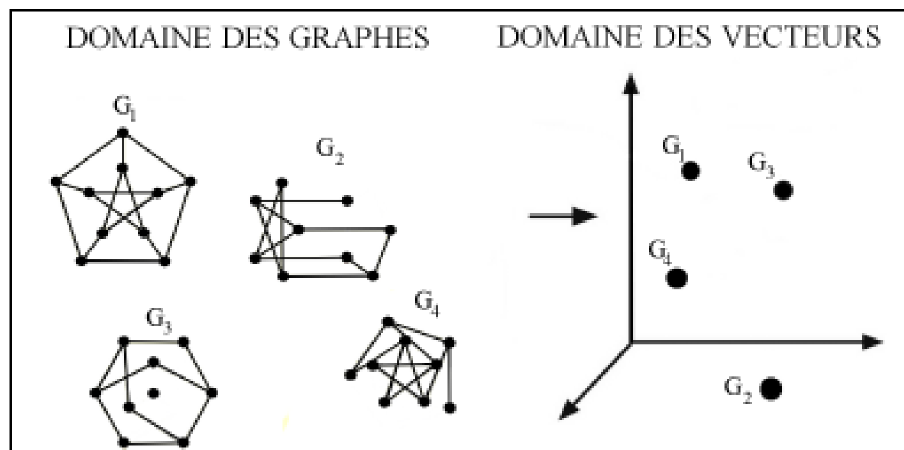
**➔The node to node matching is lost !**

Information extraction by feature selection → Construction of a feature vector:

$$\varphi : G \rightarrow R^n \quad => \quad \varphi(g) = (x_1, \ldots, x_n)$$

Combination of structural and statistical approaches



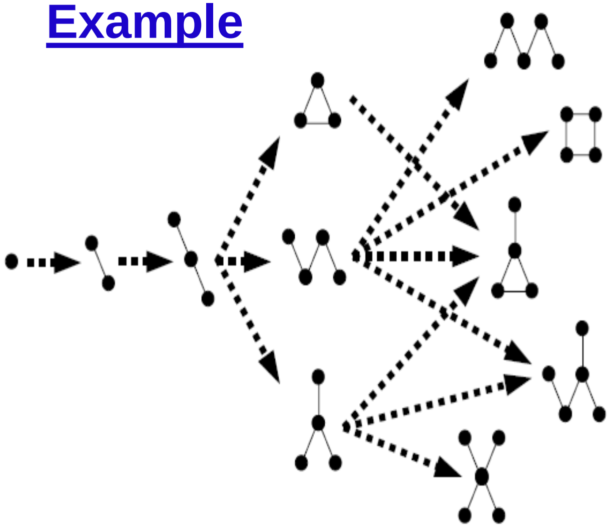



**What does it mean?**

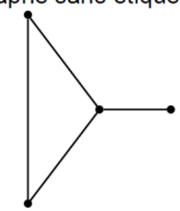
# Structural PR ➔ Graph based matching

## Embedding topological information [Sidère09]

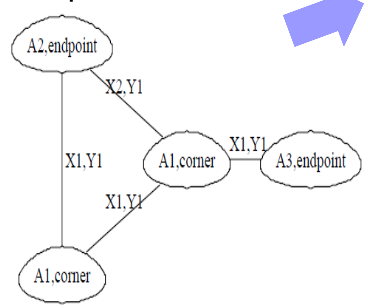
**Example**

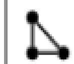
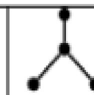


Graphe sans étiquette $G$



Graphe avec attributs




| Pattern |  |  |  |  |  |  |
|---|---|---|---|---|---|---|
| Freq. | 4 | 4 | 5 | 2 | 1 | 1 |
| A1, corner | 2 | 4 | 9 | 4 | 2 | 2 |
| A2, corner | 0 | 0 | 0 | 0 | 0 | 0 |
| A3, corner | 0 | 0 | 0 | 0 | 0 | 0 |
| A1, endpoint | 0 | 0 | 0 | 0 | 0 | 0 |
| A2, endpoint | 1 | 2 | 4 | 2 | 1 | 1 |
| A3, endpoint | 1 | 1 | 2 | 2 | 0 | 1 |
| X1, Y1 | 0 | 3 | 7 | 5 | 2 | 2 |
| X1, Y2 | 0 | 0 | 0 | 0 | 0 | 0 |
| X2, Y1 | 0 | 1 | 3 | 1 | 1 | 1 |
| X2, Y2 | 0 | 0 | 0 | 0 | 0 | 0 |

$$\varphi(g) = (x_1, \ldots, x_n) = (4,4,5,2,1,1)$$

**Lexicon of Topological patterns**

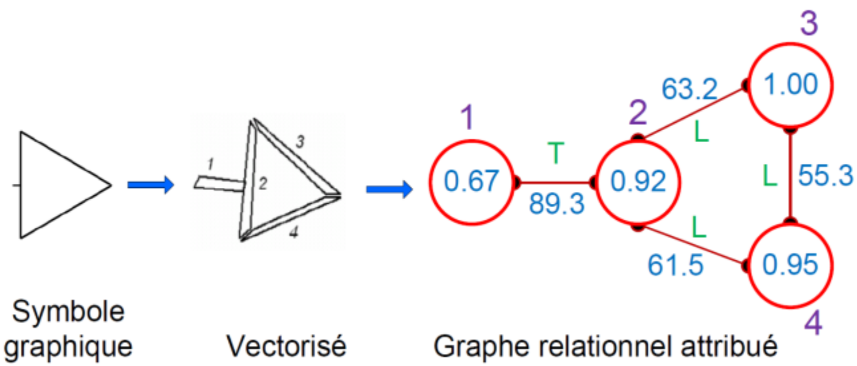
Frequency of the patterns ➔ Construction of a vector

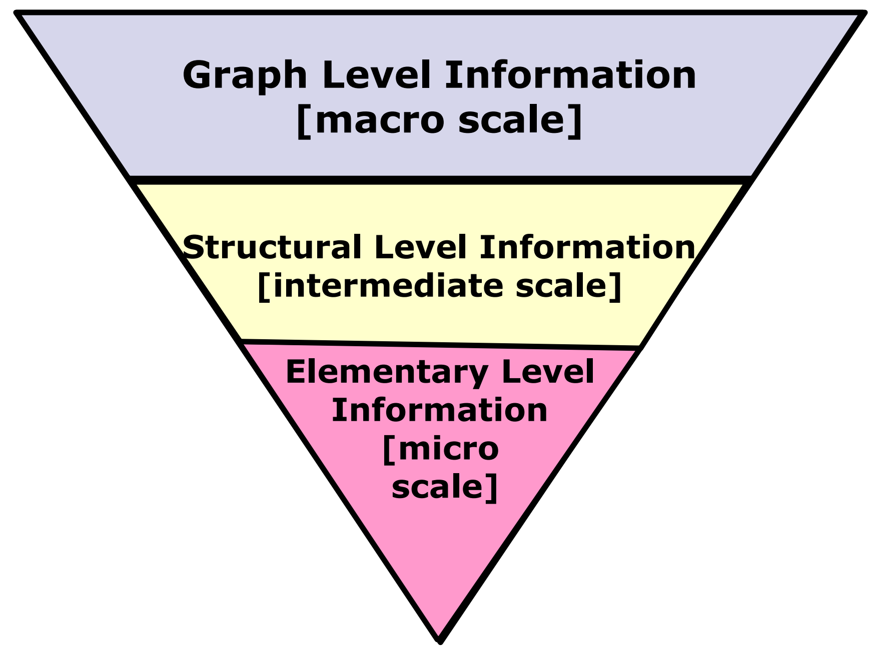
Trying to take care of attributes ➔ Construction of a Matrix ➔ discrétisation

# Structural PR ➔ Graph based matching

## Fuzzy multi-level Graph Embedding [Luqman13]

**Trying to embed topological and statistical information**



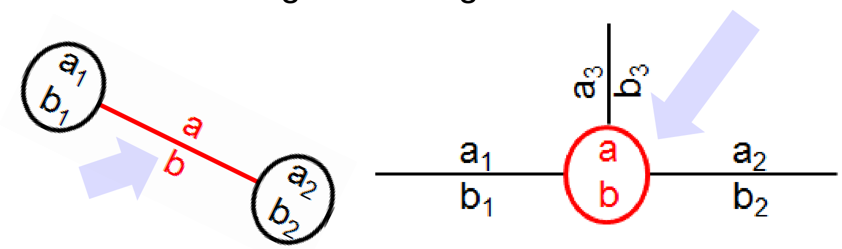

$$\varphi(g) = (x_1, \dots, x_n)$$

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Graph order | Graph size | Fuzzy histogram of node degrees | Fuzzy histograms of numeric resemblance attributes | Crisp histograms of symbolic resemblance attributes | Fuzzy histograms of numeric node attributes | Crisp histograms of symbolic node attributes | Fuzzy histograms of numeric edge attributes | Crisp histograms of symbolic edge attributes |

# Structural PR ➔ Graph based matching

## Fuzzy multi-level Graph Embedding [Luqman13]

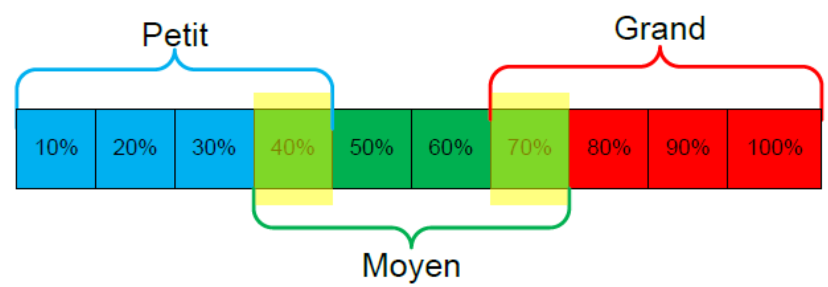
**Adding local topology information**
Ressemblance Degree on node attributes
Ressemblance Degree on edge attributes



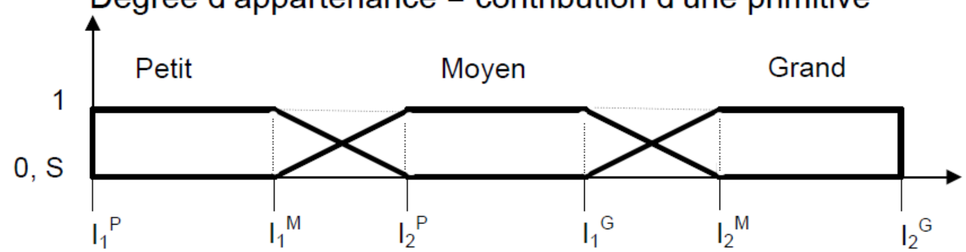

$$\textit{numeric resemblance} = \frac{\min(|a_1|,|a_2|)}{\max(|a_1|,|a_2|)}$$

$$\textit{symbolic resemblance} = \begin{cases} 1 & \textit{if } b_1 = b_2 \\ 0 & \textit{otherwise} \end{cases}$$

**Fuzzy embedding**
Frequency Histogram
Fuzzy transformation




Dégree d'appartenance = contribution d'une primitive




$\varphi(g) = (x_1, \ldots, x_n)$

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Graph order | Graph size | Fuzzy histogram of node degrees | Fuzzy histograms of numeric resemblance attributes | Crisp histograms of symbolic resemblance attributes | Fuzzy histograms of numeric node attributes | Crisp histograms of symbolic node attributes | Fuzzy histograms of numeric edge attributes | Crisp histograms of symbolic edge attributes |

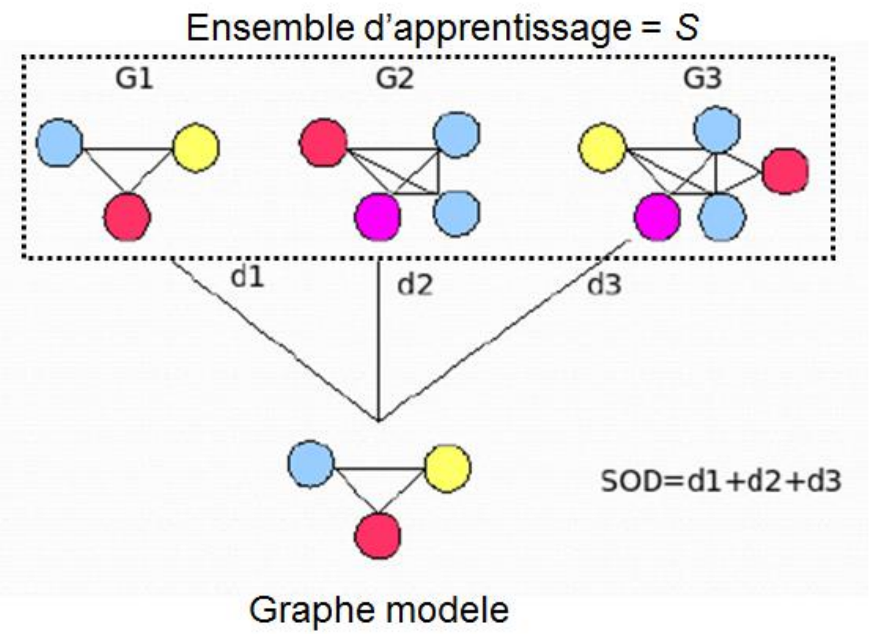
# Machine Learning with graphs?

## Median Graph

$$\overline{g} = arg_{g \in C}\, min\, SOD\,(g, S)$$

$$SOD\,(g, S) = \sum_{i=1}^{|S|} d\,(g, g_i)$$

*S = a set of graphs*
*C = set of possible graphs derived from S*
*D = an edit distance*




## Many other Problems…

How to define GED Costs?

How to define the embedding functions?

➔ **Learning to match Graphs is the actual crutial question…**

# References

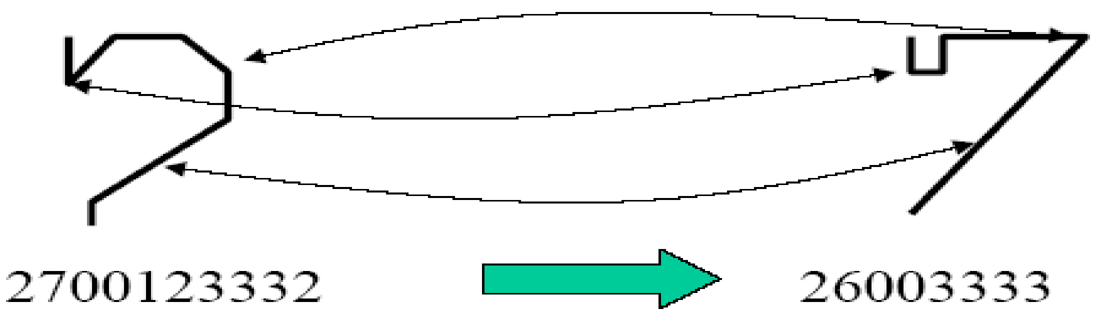

- Gauzere, B., Brun, L., and Villemin, D. (2011).Two new graph kernels and applications to chemoinformatics. Pattern Recognition Letters.
- Riesen, K., Neuhaus, M., and Bunke, H. (2007). Graph embedding in vector spaces by means of prototype selection. In Escolano, F. and Vento, M., editors, 6th IAPR-TC15 International Workshop GbRPR 2007, pages 383{393. IAPR TC15, Springer-Verlag.
- H. Bunke et X. Jiang : Graph matching and similarity, chapitre de “Intelligent systems and interfaces”, Kluwer, 2000
- Jaume Gibert, Ernest Valveny, Horst Bunke: Graph embedding in vector spaces by node attribute statistics. Pattern Recognition 45(9): 3072-3083 (2012)
- S. Sorlin, C. Solnon et J.-M. Jolion : A Generic Graph Distance Measure Based on Multivalent Matchings, chapitre de “Applied Graph Theory in PR”, Vol 52:151-182, Springer, 2007
- S. Sorlin et C. Solnon : Reactive Tabu Search for Measuring Graph Similarity, GbR, LNCS 3434:172-182, 2005
- Zeina Abu-Aisheh. Anytime and Distributed Approaches for Graph Matching. Université François Rabelais de Tours – France. May 18th, 2016.
- D. Conte, J.-Y. Ramel, N. Sidère, M.M. Luqman, B. Gauzère, J. Gibert,L. Brun, et M. Vento. “A Comparison of Explicit and Implicit GraphEmbedding Methods for Pattern Recognition”. In : Proceedings ofthe 9th IAPR-TC15 workshop on Graph-based Representation in PatternRecognition (GbR 2013). 2013, p. 81-90.

# References


- Zeina Abu-Aisheh, Romain Raveaux, P Martineau, Jean-Yves Ramel. Distributed Graph Matching and Graph Indexing Approaches: Applications to Pattern Recognition, ICPRAM 2015, Doctoral consortium
- Muhammad Muzzamil Luqman, Jean-Yves Ramel, Josep Lladós, Thierry Brouard: Fuzzy multilevel graph embedding. Pattern Recognition 46(2): 551-565 (2013)
- Muhammad Muzzamil Luqman, Jean-Yves Ramel, Josep Lladós, Thierry Brouard: Subgraph Spotting through Explicit Graph Embedding: An Application to Content Spotting in Graphic Document Images. ICDAR 2011: 870-874
- Romain Raveaux, Sébastien Adam, Pierre Héroux, Éric Trupin: Learning graph prototypes for shape recognition. Computer Vision and Image Understanding 115(7): 905-918 (2011)
- Romain Raveaux, Jean-Christophe Burie, Jean-Marc Ogier: A graph matching method and a graph matching distance based on subgraph assignments. Pattern Recognition Letters 31(5): 394-406 (2010)
- Nicolas Sidere, Pierre Héroux, Jean-Yves Ramel: Vector Representation of Graphs: Application to the Classification of Symbols and Letters. ICDAR 2009: 681-685
- Rashid Jalal Qureshi, Jean-Yves Ramel, Hubert Cardot: Graph Based Shapes Representation and Recognition. GbRPR 2007: 49-60
- Rashid Jalal Qureshi, Jean-Yves Ramel, Didier Barret, Hubert Cardot: Spotting Symbols in Line Drawing Images Using Graph Representations. GREC 2007: 91-103

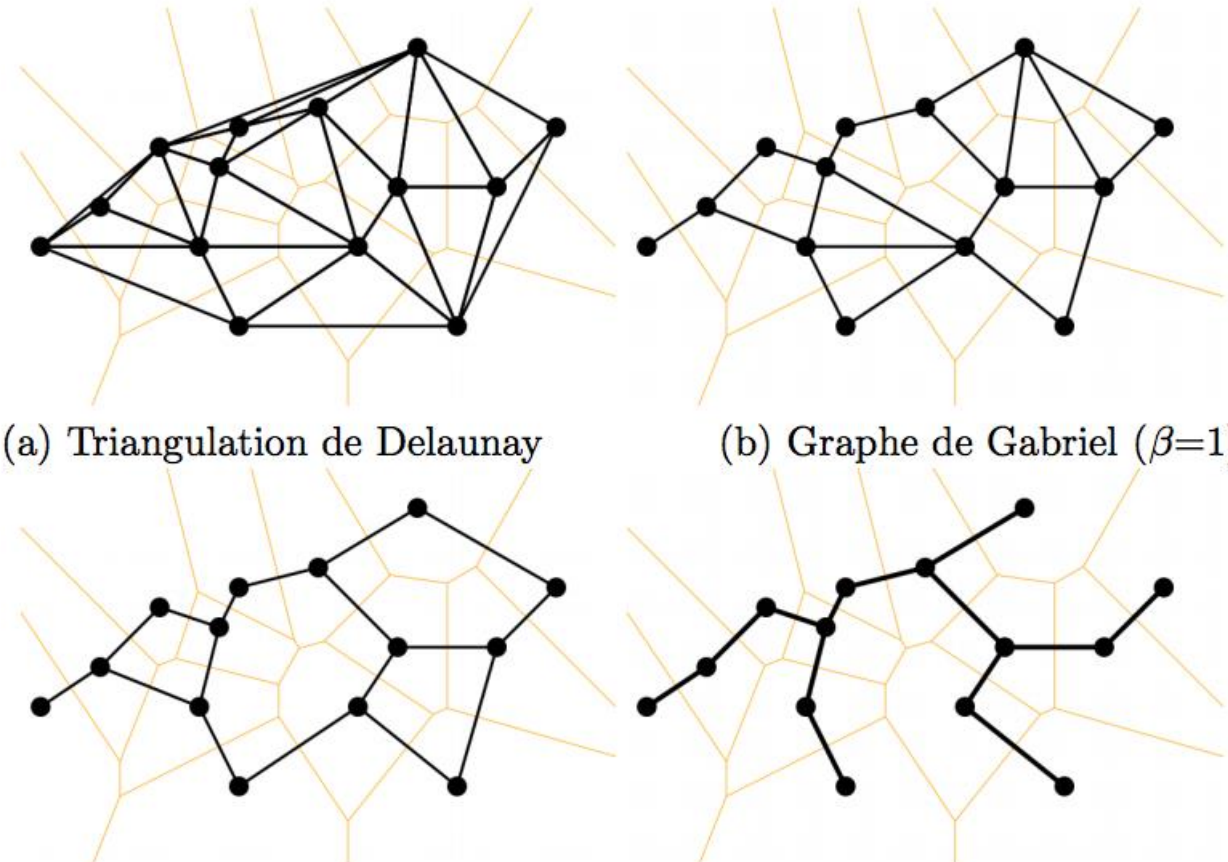
# String Comparison

- **String Edit distance** : cost to transform a sequence into an other
  - Possible transformations :
    - Insertion of an element
    - Suppression of an element
    - Substitution of an element by an other one




Distance = cost( 7→6 ) + cost( suppr1 ) + cost( 2→3 ) + cost( suppr2 )

# Spatial relationship graph

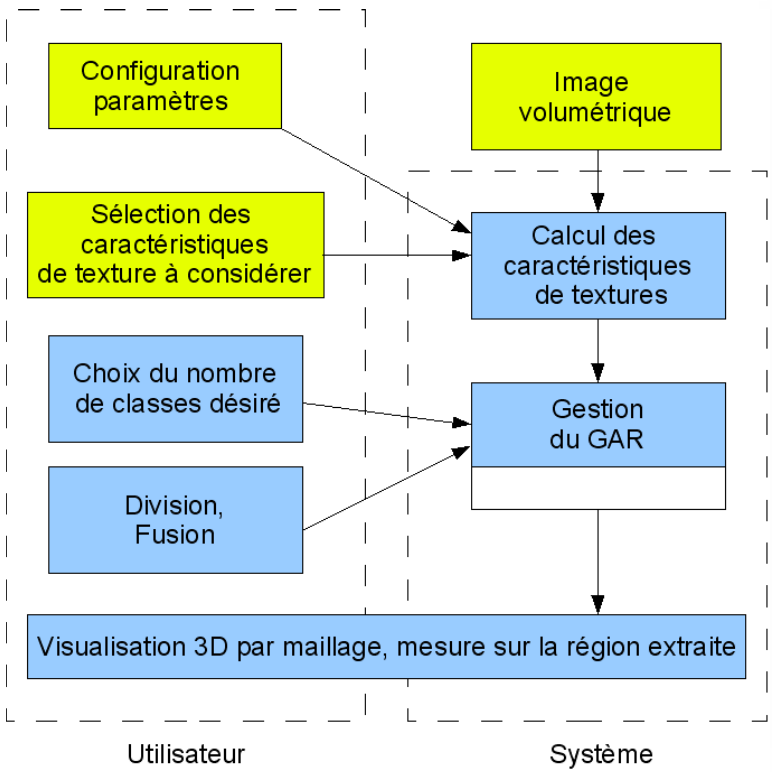


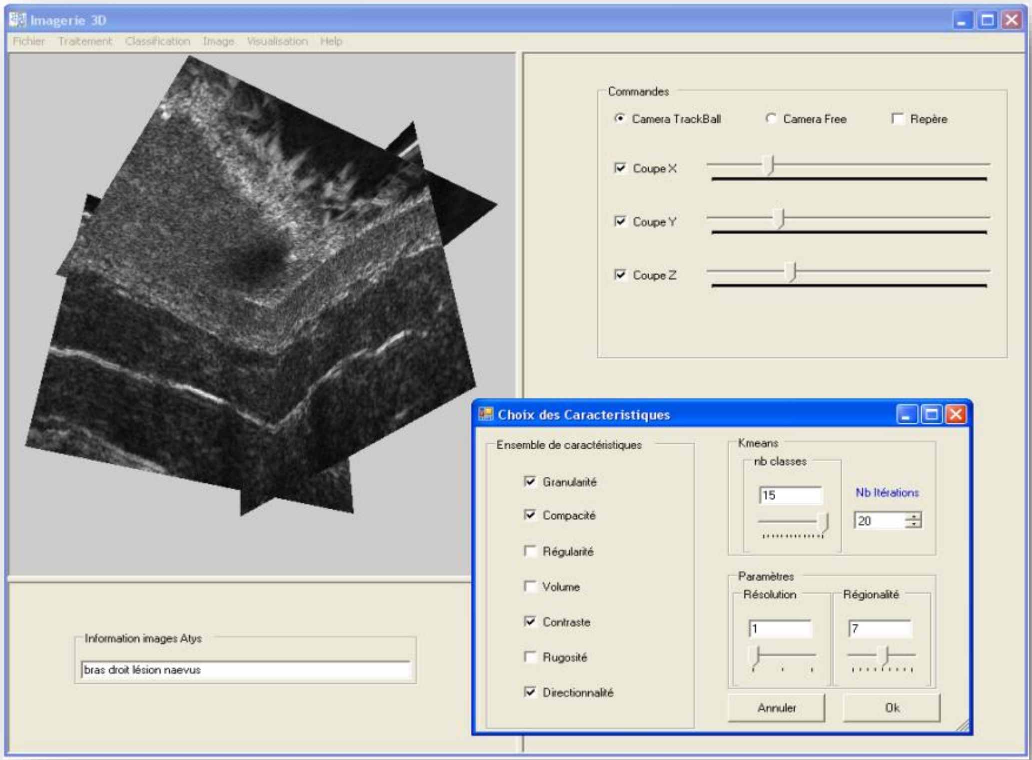
(a) Triangulation de Delaunay

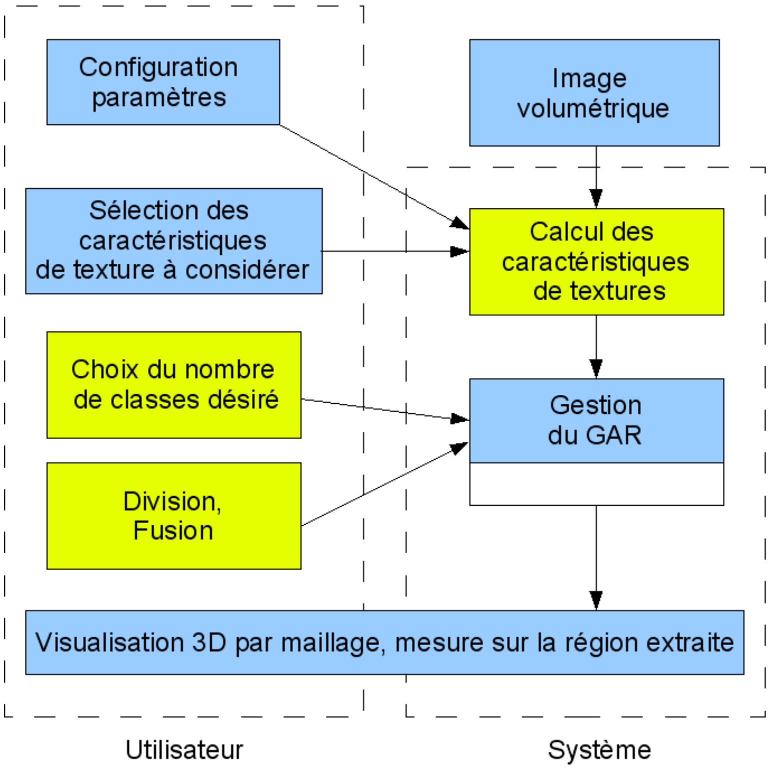
(b) Graphe de Gabriel ($\beta$=1)

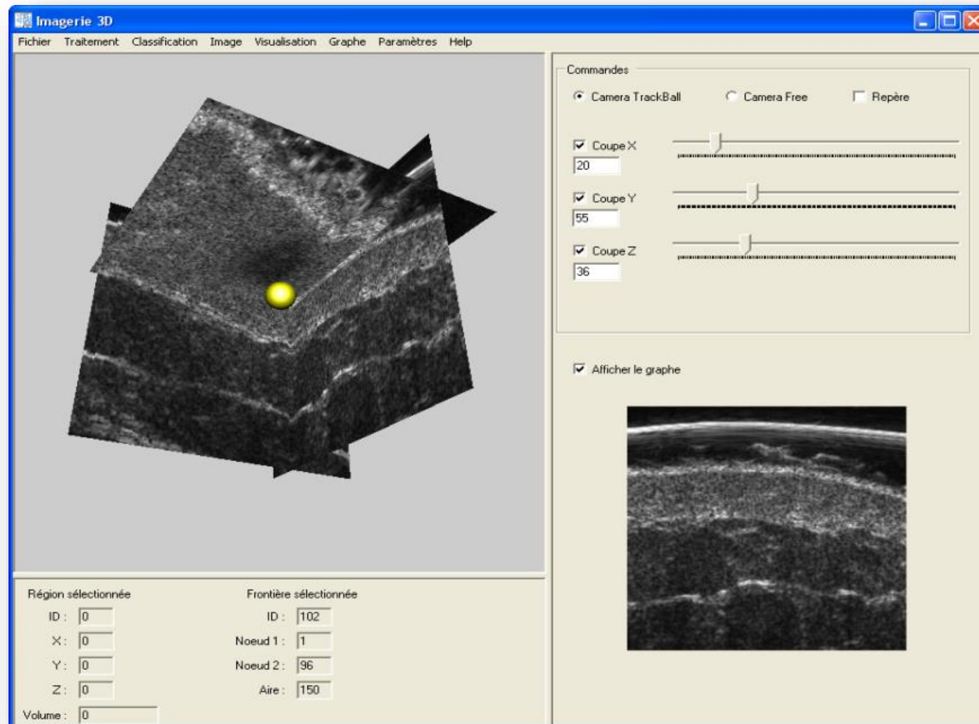
(c) Graphe de voisinage

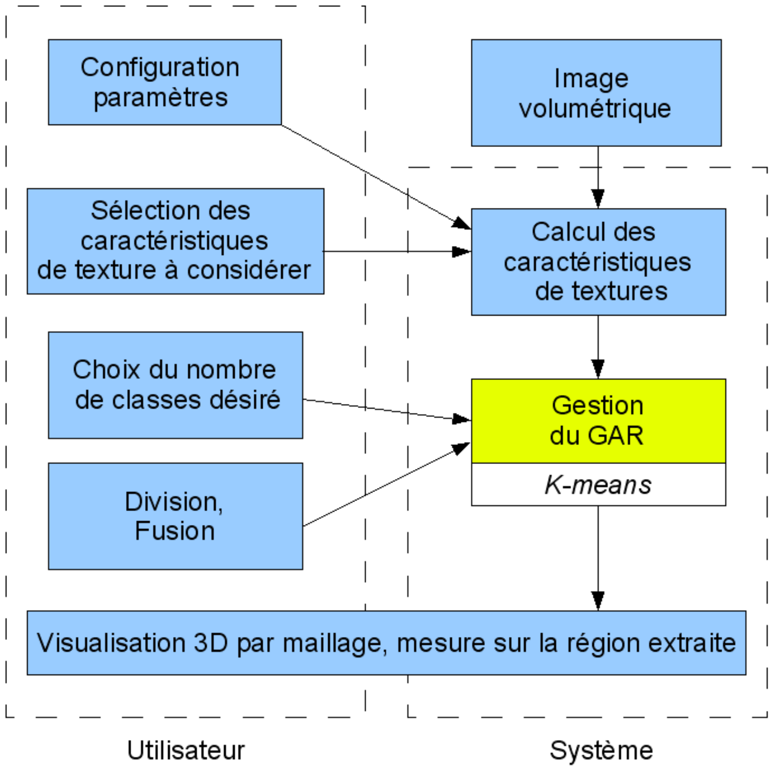
(d) Arbre de recouvrement minimal

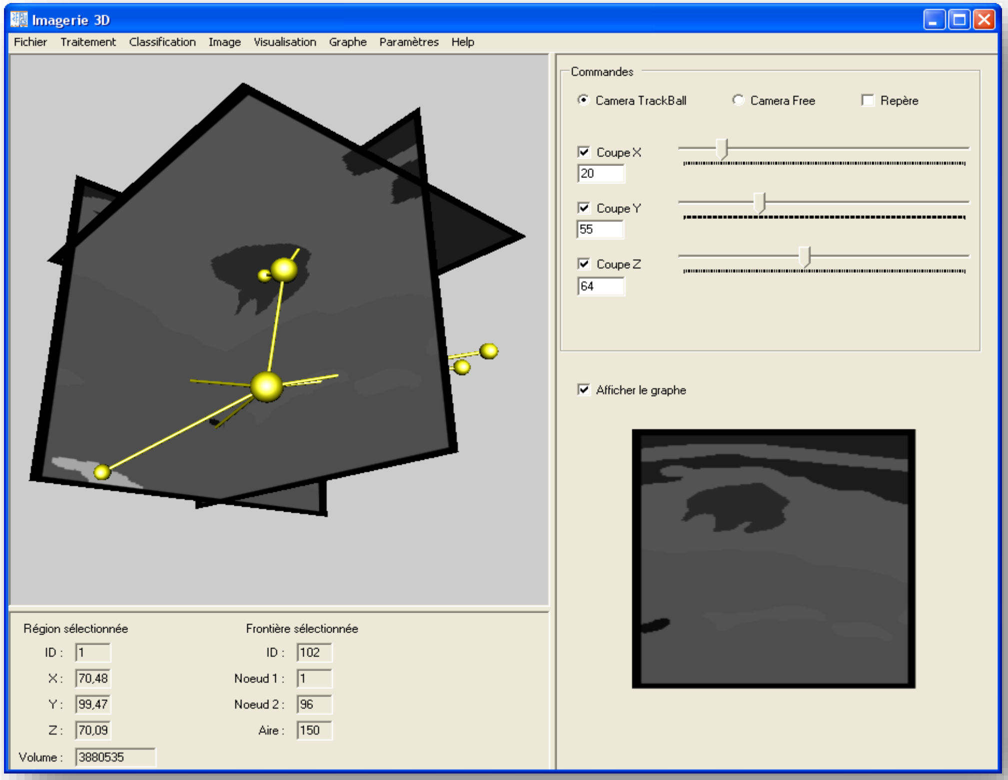

Configuration paramètres
Sélection des caractéristiques de texture à considérer
Choix du nombre de classes désiré
Division, Fusion
Image volumétrique
Calcul des caractéristiques de textures
Gestion du GAR
Visualisation 3D par maillage, mesure sur la région extraite
Utilisateur
Système

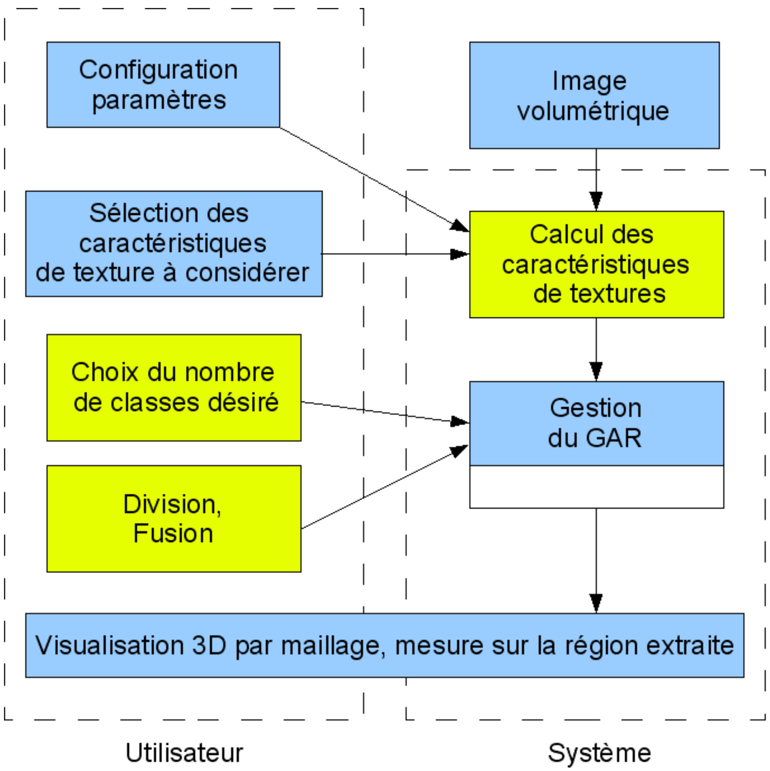

Imagerie 3D
Fichier Traitement Classification Image Visualisation Help
Commandes
Camera TrackBall
Camera Free
Repère
Coupe X
Coupe Y
Coupe Z
Information images Atys
bras droit lésion naevus
Choix des Caracteristiques
Ensemble de caractéristiques
Granularité
Compacité
Régularité
Volume
Contraste
Rugosité
Directionnalité
Kmeans
nb classes
15
Nb Itérations
20
Paramètres
Résolution
1
Régionalité
7
Annuler
Ok

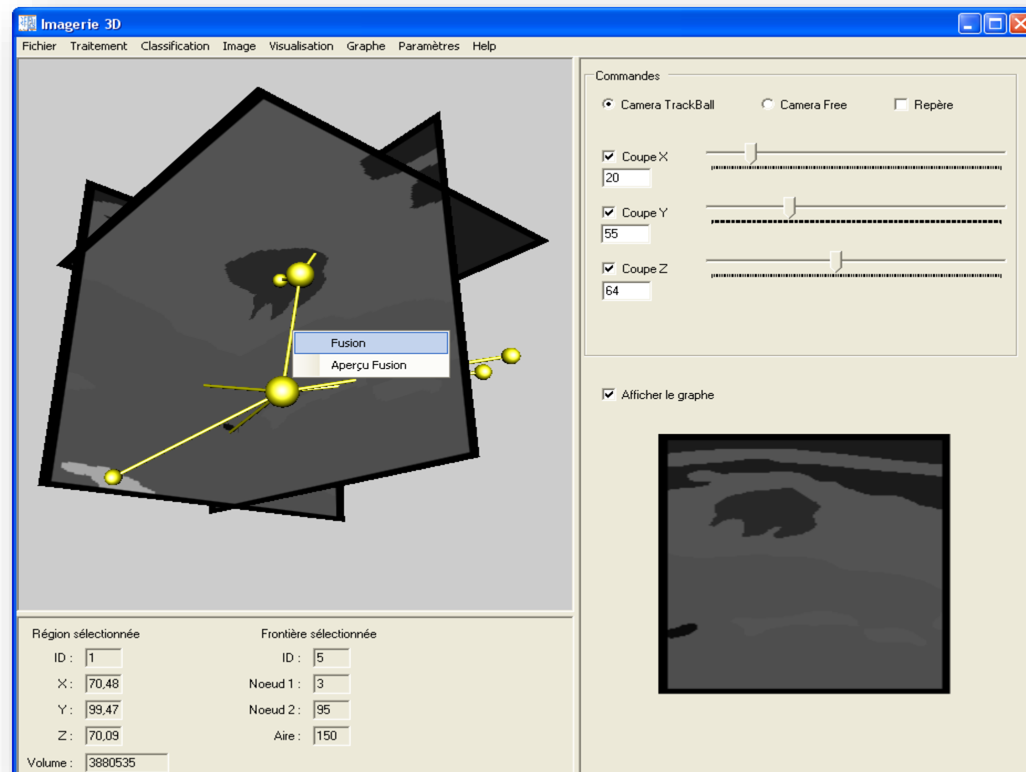

Configuration paramètres
Image volumétrique
Sélection des caractéristiques de texture à considérer
Calcul des caractéristiques de textures
Choix du nombre de classes désiré
Gestion du GAR
Division, Fusion
Visualisation 3D par maillage, mesure sur la région extraite
Utilisateur
Système

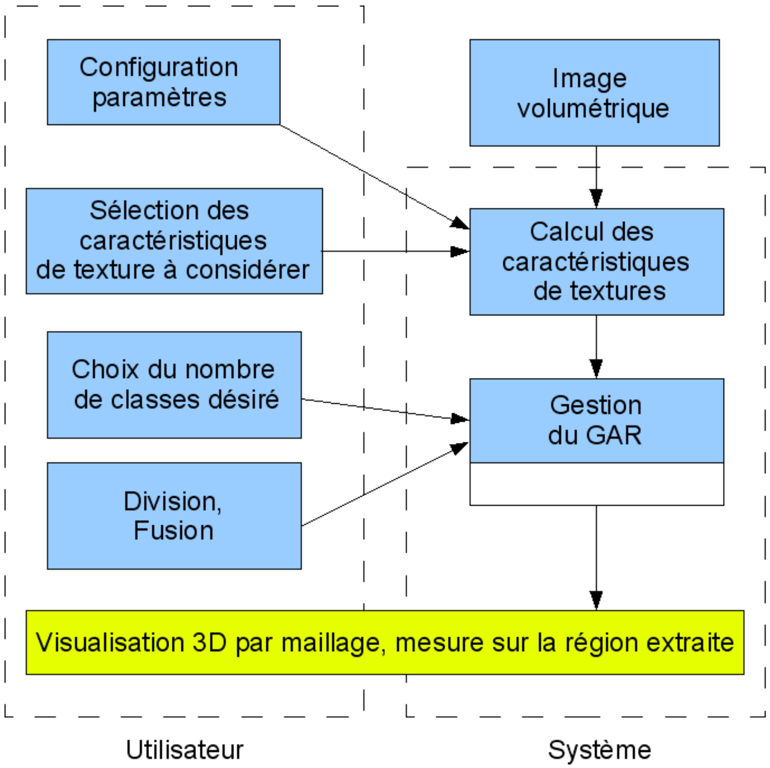

Imagerie 3D
Fichier Traitement Classification Image Visualisation Graphe Paramètres Help
Commandes
Camera TrackBall
Camera Free
Repère
Coupe X
20
Coupe Y
55
Coupe Z
36
Afficher le graphe
Région sélectionnée
ID : 0
X : 0
Y : 0
Z : 0
Volume : 0
Frontière sélectionnée
ID : 102
Noeud 1 : 1
Noeud 2 : 96
Aire : 150

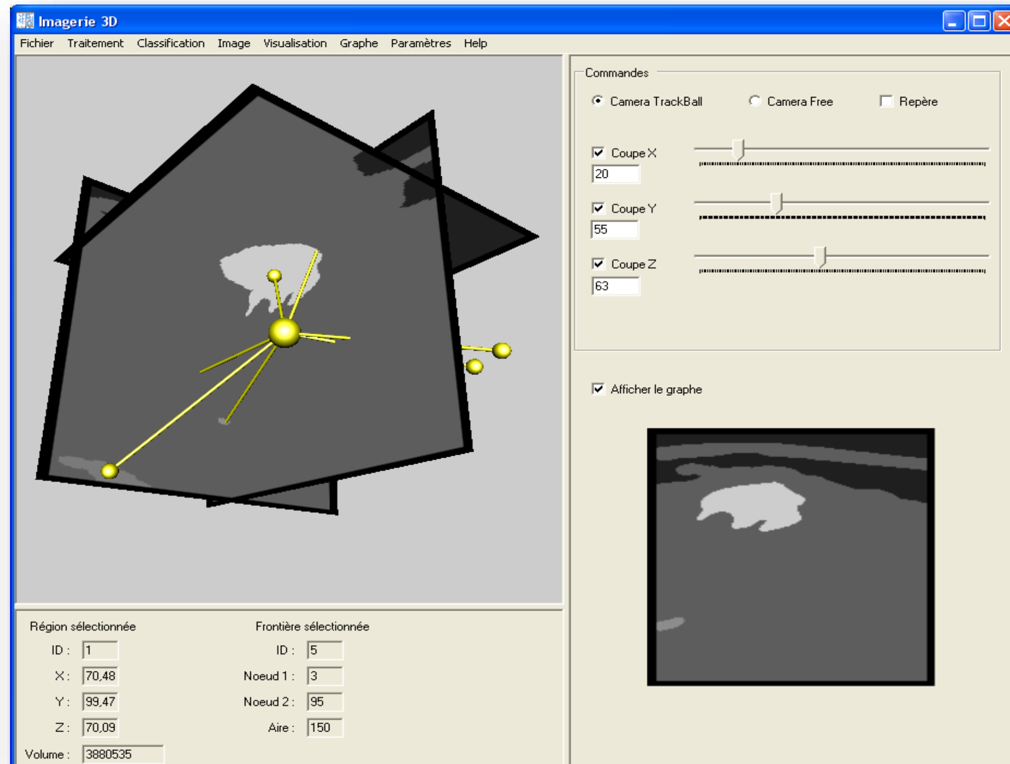

Configuration paramètres
Image volumétrique
Sélection des caractéristiques de texture à considérer
Calcul des caractéristiques de textures
Choix du nombre de classes désiré
Gestion du GAR
K-means
Division, Fusion
Visualisation 3D par maillage, mesure sur la région extraite
Utilisateur
Système

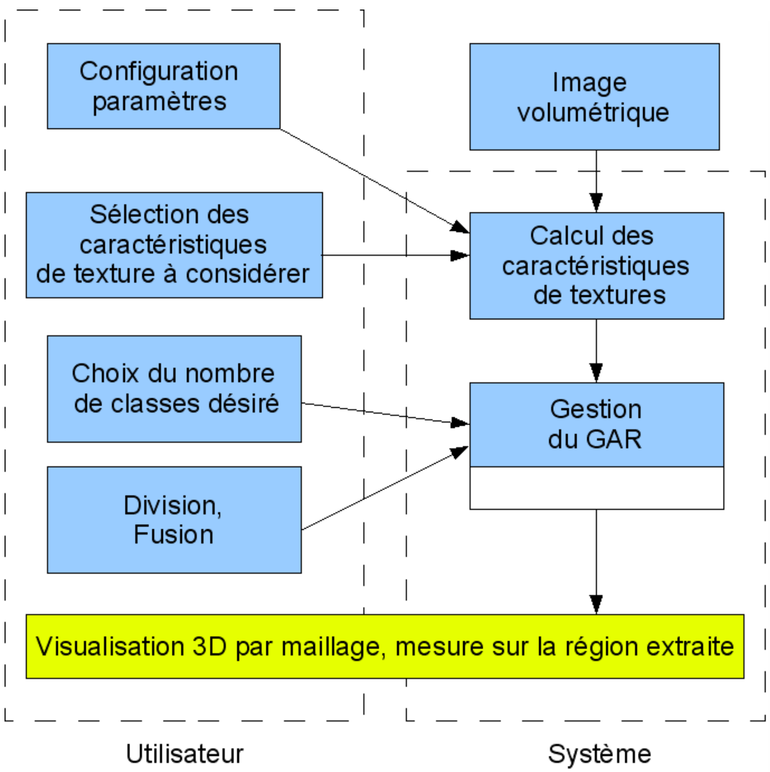

Imagerie 3D
Fichier Traitement Classification Image Visualisation Graphe Paramètres Help
Commandes
Camera TrackBall
Camera Free
Repère
Coupe X
20
Coupe Y
55
Coupe Z
64
Afficher le graphe
Région sélectionnée
ID : 1
X : 70,48
Y : 99,47
Z : 70,09
Volume : 3880535
Frontière sélectionnée
ID : 102
Noeud 1 : 1
Noeud 2 : 96
Aire : 150

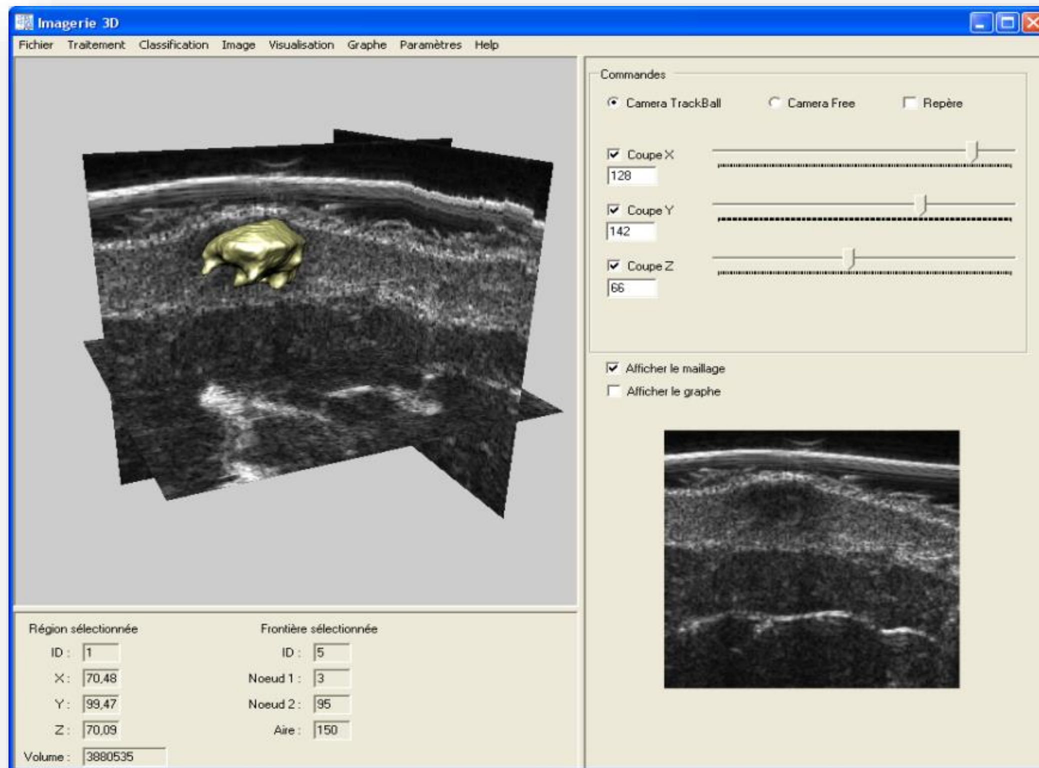

Configuration paramètres
Image volumétrique
Sélection des caractéristiques de texture à considérer
Calcul des caractéristiques de textures
Choix du nombre de classes désiré
Gestion du GAR
Division, Fusion
Visualisation 3D par maillage, mesure sur la région extraite
Utilisateur
Système


Imagerie 3D
Fichier Traitement Classification Image Visualisation Graphe Paramètres Help
Commandes
Camera TrackBall
Camera Free
Repère
Coupe X
20
Coupe Y
55
Coupe Z
64
Fusion
Aperçu Fusion
Afficher le graphe
Région sélectionnée
ID : 1
X : 70,48
Y : 99,47
Z : 70,09
Volume : 3880535
Frontière sélectionnée
ID : 5
Noeud 1 : 3
Noeud 2 : 95
Aire : 150


Configuration paramètres
Image volumétrique
Sélection des caractéristiques de texture à considérer
Calcul des caractéristiques de textures
Choix du nombre de classes désiré
Gestion du GAR
Division, Fusion
Visualisation 3D par maillage, mesure sur la région extraite
Utilisateur
Système


Imagerie 3D
Fichier Traitement Classification Image Visualisation Graphe Paramètres Help
Commandes
Camera TrackBall
Camera Free
Repère
Coupe X
20
Coupe Y
55
Coupe Z
63
Afficher le graphe
Région sélectionnée
ID : 1
X : 70,48
Y : 99,47
Z : 70,09
Volume : 3880535
Frontière sélectionnée
ID : 5
Noeud 1 : 3
Noeud 2 : 95
Aire : 150


Configuration paramètres
Image volumétrique
Sélection des caractéristiques de texture à considérer
Calcul des caractéristiques de textures
Choix du nombre de classes désiré
Gestion du GAR
Division, Fusion
Visualisation 3D par maillage, mesure sur la région extraite
Utilisateur
Système


Imagerie 3D
Fichier Traitement Classification Image Visualisation Graphe Paramètres Help
Commandes
Camera TrackBall
Camera Free
Repère
Coupe X
128
Coupe Y
142
Coupe Z
66
Afficher le maillage
Afficher le graphe
Région sélectionnée
ID : 1
X : 70,48
Y : 99,47
Z : 70,09
Volume : 3880535
Frontière sélectionnée
ID : 5
Noeud 1 : 3
Noeud 2 : 95
Aire : 150