

# Hierarchy Construction Schemes within the Scale Set Framework

**Jean-Hugues PRUVOT, Luc BRUN**

GreyC Laboratory, Image Team  
CNRS UMR 6072 ENSICAEN

6th IAPR -TC-15 Workshop on Graph-based Representations in  
Pattern Recognition



# Outline

- 1 **Introduction**
- 2 **The Scale Set Framework**
  - The Causality principle
  - Optimal Cuts
- 3 **Merging Heuristics**
  - Sequential Merging
  - Parallel Merging
- 4 **Conclusion and Outlooks**

# Introduction

## 2 approaches for Segmentation

- Segmentation  $\Rightarrow$  *Horowitz[1976]* with a predicate  $P$ 
  - ▶ split/merge while an homogeneity criteria
  - ▶ only a local criteria
- The use of energy minimisation scheme within the region based segmentation framework
  - ▶ Level-Set, Bayesian, Min-cut / N-Cut, Minimum Description Length
  - ▶ allows to define criteria which should be globally optimised over a partition
  - ▶ allows an objective evaluation the segmentations
  - ▶ Level-Set  $\Rightarrow$  minimisation for one scale parameter  $\lambda$

# The Scale Set Framework

- Level Set / N-Cut / MDL  $\Rightarrow$  local minimum in a full search space

## Principle

- minimize an energy partition  $E(P)$  :  
(  $E(P)$  is supposed to be an Affine Separable Energy (ASE) )  
Energy for each region  $R_i$  (weighted sum of 2 terms) :

$$E(R) = \sum_{R_i \in R} D(R_i) + \lambda C(R_i)$$

$D(R_i)$  : internal region energy (fit to data)

$C(R_i)$  : complexity energy (regularization term)

$\lambda$  : scale parameter

# The Scale Set Framework

- Guigues approach

- ▶ global minimum in a narrow search space
- ▶ *bottom-up* approach
- ▶ provides the optimal partition for each value of a scale parameter

$$\lambda \in \mathbb{R}^+$$

## Principle

- minimize an energy partition  $E(P)$  :  
(  $E(P)$  is supposed to be an Affine Separable Energy (ASE) )

Energy for each region  $R_i$  (weighted sum of 2 terms) :

$$E(R) = \sum_{R_i \in R} D(R_i) + \lambda C(R_i)$$

$D(R_i)$  : internal region energy (fit to data)

$C(R_i)$  : complexity energy (regularization term)

$\lambda$  : scale parameter

# The Scale Set Framework

- Guigues approach

- ▶ global minimum in a narrow search space
- ▶ *bottom-up* approach
- ▶ provides the optimal partition for each value of a scale parameter  
 $\lambda \in \mathbb{R}^+$

## Principle

- minimize an energy partition  $E(P)$  :  
(  $E(P)$  is supposed to be an Affine Separable Energy (ASE) )
  - ▶ Energy for each region  $R_i$  (weighted sum of 2 terms) :

$$E(R) = \sum_{R_i \in R} D(R_i) + \lambda C(R_i)$$

- ▶  $D(R_i)$  : internal region energy (fit to data)
- ▶  $C(R_i)$  : complexity energy (regularization term)
- ▶  $\lambda$  : scale parameter

# The Scale Set Framework

- Guigues approach

- ▶ global minimum in a narrow search space
- ▶ *bottom-up* approach
- ▶ provides the optimal partition for each value of a scale parameter  
 $\lambda \in \mathbb{R}^+$

## Principle

- minimize an energy partition  $E(P)$  :  
(  $E(P)$  is supposed to be an Affine Separable Energy (ASE) )
  - ▶ Energy for each region  $R_i$  (weighted sum of 2 terms) :

$$E(R) = \sum_{R_i \in R} D(R_i) + \lambda C(R_i)$$

- ▶  $D(R_i)$  : internal region energy (fit to data)
- ▶  $C(R_i)$  : complexity energy (regularization term)
- ▶  $\lambda$  : scale parameter

# used energy

## Energy $E_\lambda(P) = \lambda C(P) + D(P)$ *affine energy*

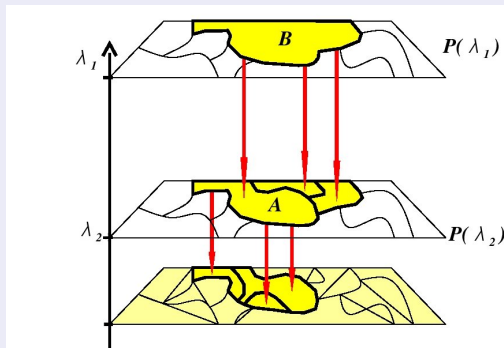
- $D$  : internal energy (fit to data) :
  - ▶ *squared error* :
$$SE(R) = \sum_{i \in R} \|c_i - \mu_R\|^2$$
  - ▶  $\Rightarrow$  minimal if each region is a pixel
- $C$  : energy of complexity (regularisation term) :
  - ▶ *total length of the boundaries*
  - ▶  $\Rightarrow$  low if the partition is composed of few regions



# The Scale Set Framework

## Causality principle

- introduced by Witkin in 1984
  - coherent structures are present at different scales in an image



- $\forall (\lambda_1, \lambda_2) \in \mathbb{R}^{+2}$  with  $\lambda_2 \leq \lambda_1 \Rightarrow P_{\lambda_1}$  can be deduced from  $P_{\lambda_2}$  by regions merging

# The Scale Set Framework

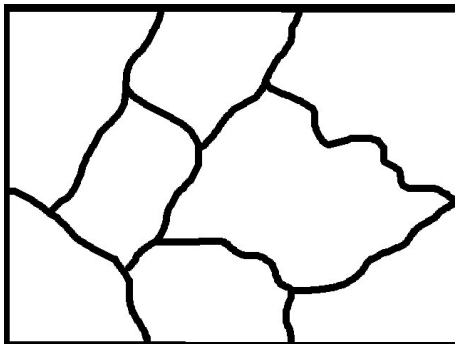
## Causality principle

- Conditions on energy  $E_\lambda(P) = \lambda C(P) + D(P)$ 
  - ▶  $P_\lambda(I)$  : partition which minimize  $E_\lambda(I)$
  - ▶ if  $P_\lambda$  is *causal* ,  $H = \{P_\lambda(I), \lambda \in \mathbb{R}^+\}$  is a hierarchy.
- Guigues  $\Rightarrow$  if C is sub-additive then P is causal
  - ▶ C sub-additive  $\iff C(R_1 \cup R_2) \leq C_{R_1} + C_{R_2}$
  - ▶ natural condition in segmentation task ( MDL  $\Rightarrow$  less parameters to describe the union of 2 regions than 2 regions)

# Construction of the initial hierarchy2

## scale-climbing

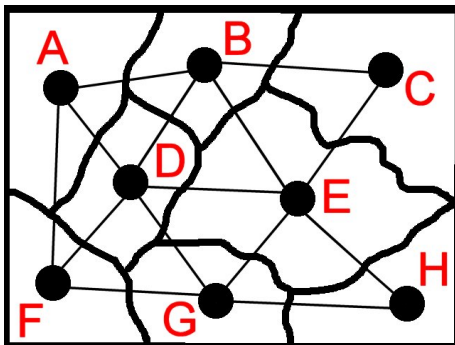
- start from an initial segmentation



# Construction of the initial hierarchy2

## scale-climbing

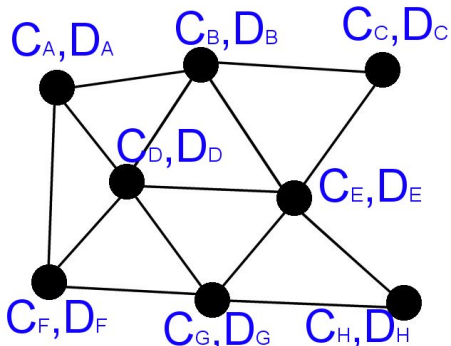
- built a region adjacency graph (RAG)



# Construction of the initial hierarchy2

## scale-climbing

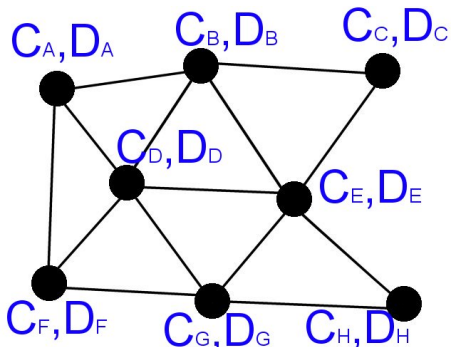
- For each region compute
  - the internal energy  $D(R_i)$
  - the complexity energy  $C(R_i)$



# Construction of the initial hierarchy2

## scale-climbing

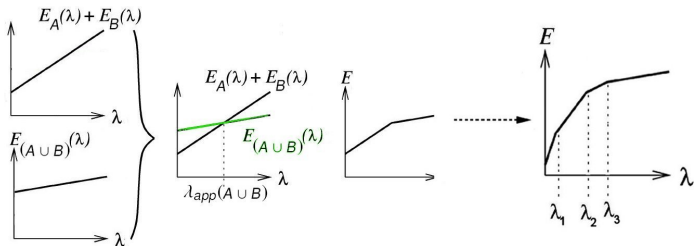
- Compute, for any couple of adjacent regions, the scale of appearance  $\lambda_{app}$ .  $\lambda_{app}$  depicts the minimum value, from which, merging those regions, contributes to less increase the global energy defined on the partition.



# Construction of the initial hierarchy<sup>2</sup>

## scale-climbing

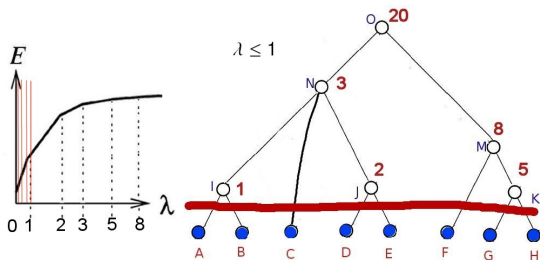
- $$\left. \begin{aligned} E_A(\lambda) + E_B(\lambda) &= \lambda(C_A + C_B) + (D_A + D_B) \\ E_{A \cup B}(\lambda) &= \lambda C_{A \cup B} + D_{A \cup B} \end{aligned} \right\} \lambda_{app}(A \cup B) = \frac{D_A + D_B - D_{A \cup B}}{C_B + C_A - C_{A \cup B}}$$
- iterating this process we get a serial of  $\lambda_{app}$  providing a set of optimal cut within  $H$
- energy of the optimal cuts within this global hierarchy is then depicted by a concave piecewise function



# Construction of the initial hierarchy

## Cuts

- For a given  $\lambda$ , we retrieve the optimal partition within  $H$

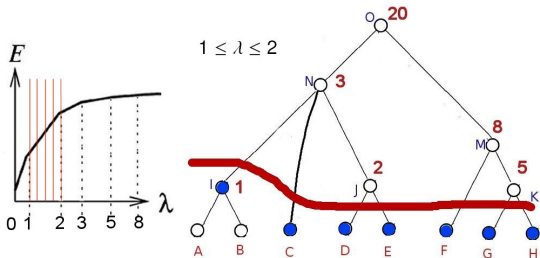




# Construction of the initial hierarchy

## Cuts

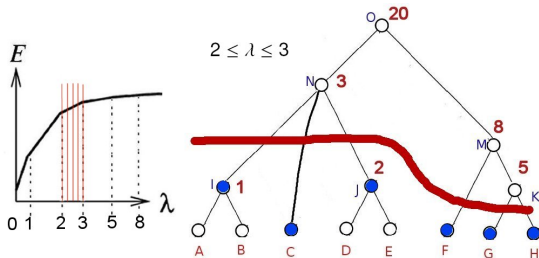
- For a given  $\lambda$ , we retrieve the optimal partition within  $H$



# Construction of the initial hierarchy

## Cuts

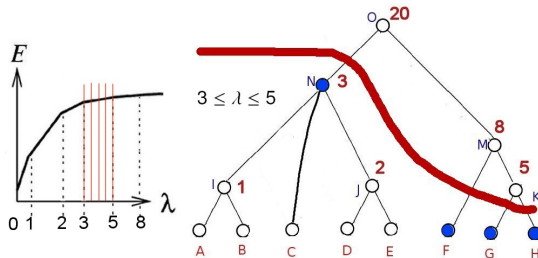
- For a given  $\lambda$ , we retrieve the optimal partition within  $H$



# Construction of the initial hierarchy

## Cuts

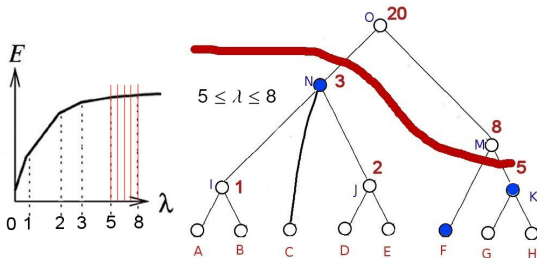
- For a given  $\lambda$ , we retrieve the optimal partition within  $H$



# Construction of the initial hierarchy

## Cuts

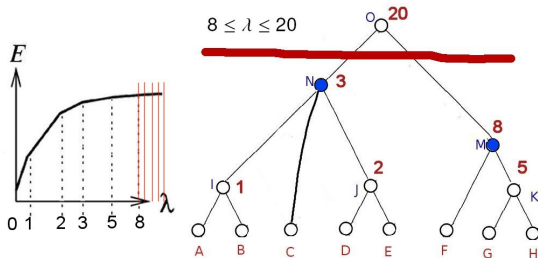
- For a given  $\lambda$ , we retrieve the optimal partition within  $H$



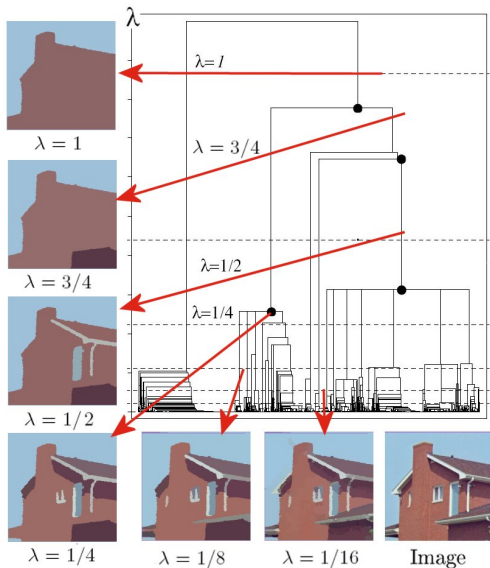
# Construction of the initial hierarchy

## Cuts

- For a given  $\lambda$ , we retrieve the optimal partition within  $H$



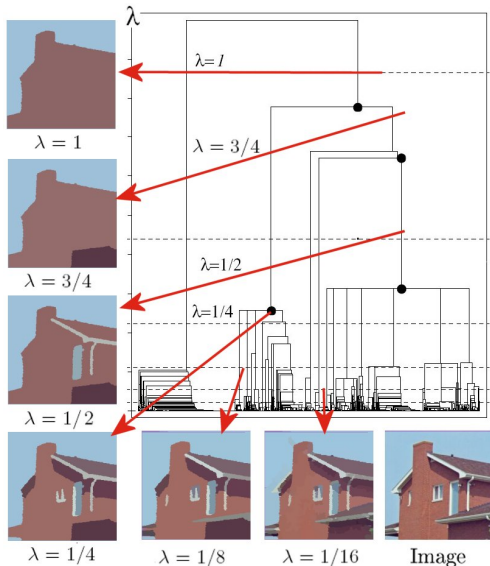
# Optimal Cut



## Briefly

- provide all solutions for any  $\lambda$
- the given solutions
  - are optimal within the hierarchy corresponding to a narrow search space
  - Partitions remains stable on whole intervals

# Optimal Cut



## Briefly

- provide all solutions for any  $\lambda$
- the given solutions
  - ▶ are optimal within the hierarchy corresponding to a narrow search space
  - ▶ Partitions remains stable on whole intervals

# Impact of the search space

## Hierarchy importance

- the research space used in this framework is restricted to the initial hierarchy  $H$ 
  - ▶ construction scheme is of a crucial importance for the optimal partitions within  $H$  built in the following steps



# Impact of the search space

## Hierarchy importance

- the research space used in this framework is restricted to the initial hierarchy  $H$ 
  - ▶ construction scheme is of a crucial importance for the optimal partitions within  $H$  built in the following steps

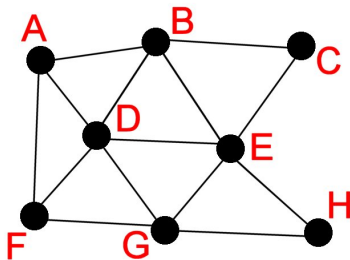
## compromise between energy/time

- sequential merging
- parallel merging

# sequential merging

## Multi-Merging

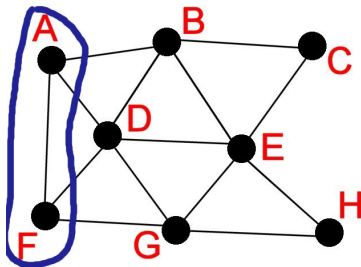
- to increasing the search space we allow more than 2 regions merging together
  - ▶ we consider for each region  $R$  of  $P$ , its set  $V(R)$  defined as  $\{R\}$  union its set of neighbours and the set  $\mathcal{P}^*(V(R))$  of all possible subsets of  $V(R)$  including  $R$
  - ▶ require computing  $(2^n - 1) \lambda_{app}$  for each node in the RAG :



# sequential merging

## Multi-Merging

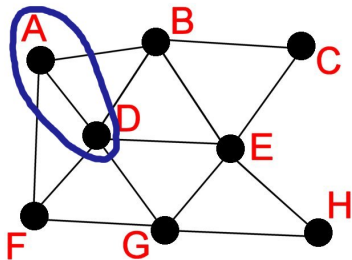
- to increasing the search space we allow more than 2 regions merging together
  - ▶ we consider for each region  $R$  of  $P$ , its set  $V(R)$  defined as  $\{R\}$  union its set of neighbours and the set  $\mathcal{P}^*(V(R))$  of all possible subsets of  $V(R)$  including  $R$
  - ▶ require computing  $(2^n - 1) \lambda_{app}$  for each node in the RAG :



# sequential merging

## Multi-Merging

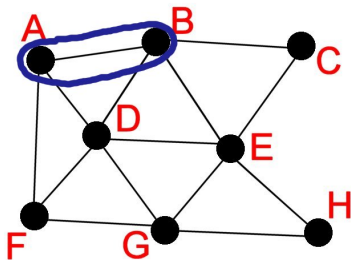
- to increasing the search space we allow more than 2 regions merging together
  - ▶ we consider for each region  $R$  of  $P$ , its set  $V(R)$  defined as  $\{R\}$  union its set of neighbours and the set  $\mathcal{P}^*(V(R))$  of all possible subsets of  $V(R)$  including  $R$
  - ▶ require computing  $(2^n - 1) \lambda_{app}$  for each node in the RAG :



# sequential merging

## Multi-Merging

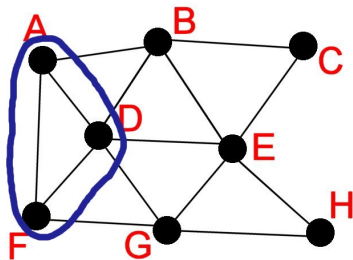
- to increasing the search space we allow more than 2 regions merging together
  - ▶ we consider for each region  $R$  of  $P$ , its set  $V(R)$  defined as  $\{R\}$  union its set of neighbours and the set  $\mathcal{P}^*(V(R))$  of all possible subsets of  $V(R)$  including  $R$
  - ▶ require computing  $(2^n - 1) \lambda_{app}$  for each node in the RAG :



# sequential merging

## Multi-Merging

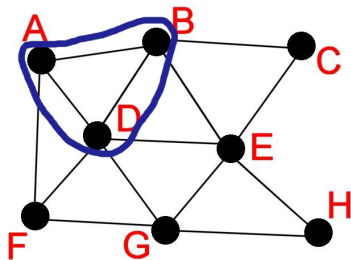
- to increasing the search space we allow more than 2 regions merging together
  - ▶ we consider for each region  $R$  of  $P$ , its set  $V(R)$  defined as  $\{R\}$  union its set of neighbours and the set  $\mathcal{P}^*(V(R))$  of all possible subsets of  $V(R)$  including  $R$
  - ▶ require computing  $(2^n - 1) \lambda_{app}$  for each node in the RAG :



# sequential merging

## Multi-Merging

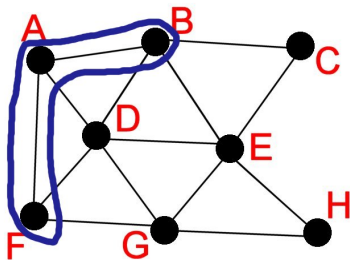
- to increasing the search space we allow more than 2 regions merging together
  - ▶ we consider for each region  $R$  of  $P$ , its set  $V(R)$  defined as  $\{R\}$  union its set of neighbours and the set  $\mathcal{P}^*(V(R))$  of all possible subsets of  $V(R)$  including  $R$
  - ▶ require computing  $(2^n - 1) \lambda_{app}$  for each node in the RAG :



# sequential merging

## Multi-Merging

- to increasing the search space we allow more than 2 regions merging together
  - ▶ we consider for each region  $R$  of  $P$ , its set  $V(R)$  defined as  $\{R\}$  union its set of neighbours and the set  $\mathcal{P}^*(V(R))$  of all possible subsets of  $V(R)$  including  $R$
  - ▶ require computing  $(2^n - 1) \lambda_{app}$  for each node in the RAG :

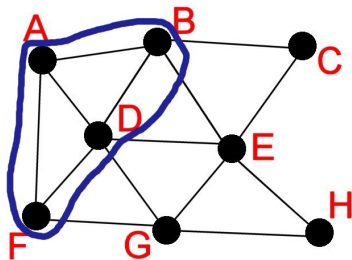




# sequential merging

## Multi-Merging

- to increasing the search space we allow more than 2 regions merging together
  - ▶ we consider for each region  $R$  of  $P$ , its set  $V(R)$  defined as  $\{R\}$  union its set of neighbours and the set  $\mathcal{P}^*(V(R))$  of all possible subsets of  $V(R)$  including  $R$
  - ▶ require computing  $(2^n - 1) \lambda_{app}$  for each node in the RAG :



# sequential merging

## Multi-Merging

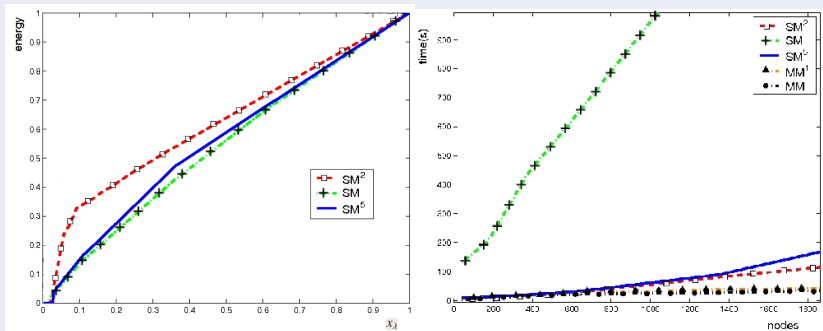
- Using the scale climbing principle, our sequential merging algorithm computes for each region  $R$  of the partition the minimal scale of appearance of a region  $R^W$

$$\lambda_{min}^+(R) = \arg \min_{W \in \mathcal{P}^*(V(R))} \frac{D(R^W) - D(W)}{C(W) - C(R^W)}$$

- complexity is bounded by  $O(|V|2^k)$  where  $|V|$  denotes the number of vertices (i.e. regions) and  $k$  represents the maximal vertices's degree of  $G$ 
  - may induce important execution time
  - +the cardinal of the subset of regions to be merge may be bounded (by 5) without altering significantly the energy of optimal cuts

# sequential merging

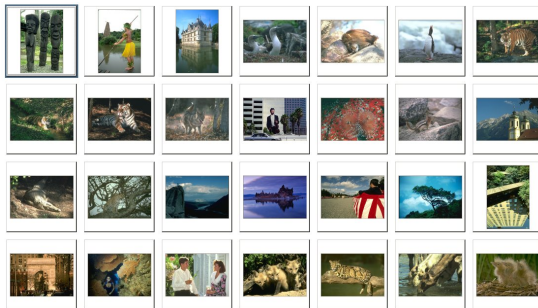
## mean energy and execution times



# Data Set

## Natural images database

- All experiments were performed on 100 natural images of the Berkeley database
- *The Berkeley Segmentation Dataset and Benchmark*
- available online at  
<http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/>



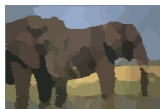

 $SM^2$ 

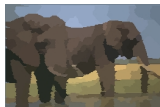
 $SM$ 

 $SM^5$ 

image

 $\lambda = 0.2$ 
 $\lambda = 0.4$ 
 $\lambda = 0.6$ 
 $\lambda = 0.8$


 $SM^2$ 

 $SM$ 

 $SM^5$ 

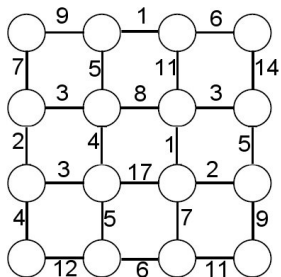
original image

 $\lambda = 0.2$  $\lambda = 0.4$  $\lambda = 0.6$  $\lambda = 0.8$

# parallel merge algorithms

## parallel merge

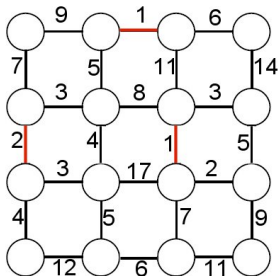
- 2 algorithms based on the notion of *maximal matching*
- using the same approach as [Haxhimusa]  
 ⇒ Maximal Independent Set on the set of edges of the graph



# parallel merge algorithms

## parallel merge

- 2 algorithms based on the notion of *maximal matching*
- using the same approach as [Haxhimusa]  
 ⇒ Maximal Independent Set on the set of edges of the graph

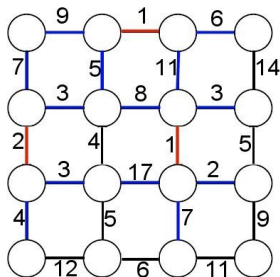




# parallel merge algorithms

## parallel merge

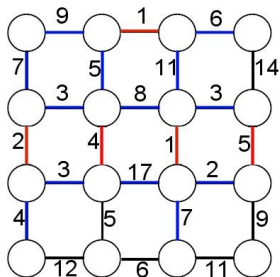
- 2 algorithms based on the notion of *maximal matching*
- using the same approach as [Haxhimusa]  
 ⇒ Maximal Independent Set on the set of edges of the graph



# parallel merge algorithms

## parallel merge

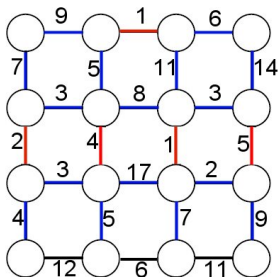
- 2 algorithms based on the notion of *maximal matching*
- using the same approach as [Haxhimusa]  
 ⇒ Maximal Independent Set on the set of edges of the graph



# parallel merge algorithms

## parallel merge

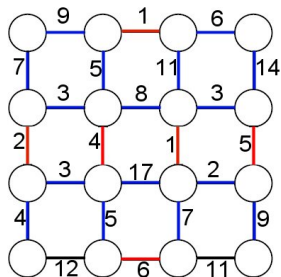
- 2 algorithms based on the notion of *maximal matching*
- using the same approach as [Haxhimusa]  
 ⇒ Maximal Independent Set on the set of edges of the graph



# parallel merge algorithms

## parallel merge

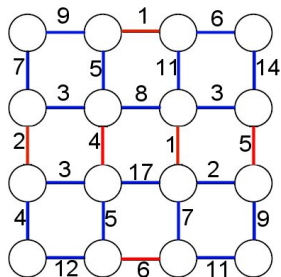
- 2 algorithms based on the notion of *maximal matching*
- using the same approach as [Haxhimusa]  
 ⇒ Maximal Independent Set on the set of edges of the graph



# parallel merge algorithms

## parallel merge

- 2 algorithms based on the notion of *maximal matching*
- using the same approach as [Haxhimusa]  
 ⇒ Maximal Independent Set on the set of edges of the graph



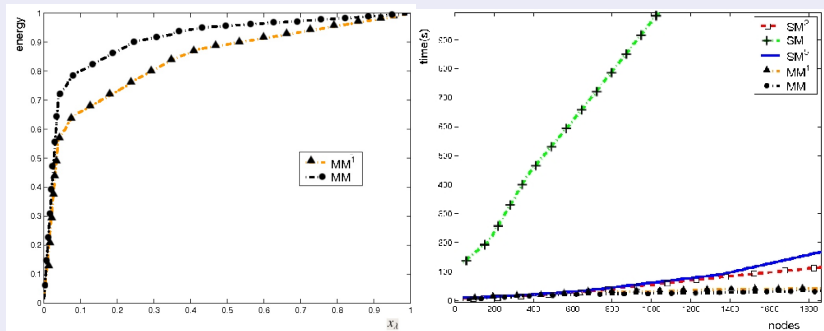
# parallel merging

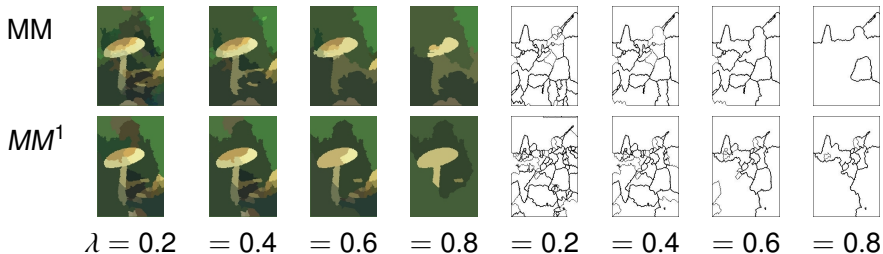
## Algos *MM* et *MM*<sup>1</sup>

- *MM*
  - ▶ + good decimation ratio
  - ▶ - detected minima  $\Rightarrow$  less and less significant as the iterations progress.
- we thus propose an alternative solution *MM*<sup>1</sup>
- $\Rightarrow$  only contract at each step edges selected at the first iteration
- $\Rightarrow$  can be seen as a combination of the method proposed by [Haxhimusa2003] and the stochastic decimation process of [Jolion2001]
  - ▶ + merge immediately only vertices corresponding to local minima.
  - ▶ + mean decimation ratio equal to 1.73 on the 100 images of the Berkeley database (comparable to the 2.0 obtained by Haxhimusa)
  - ▶ + improve the energy of optimal cuts.

# parallel merging

## mean energy and execution times





**AJOUTER D'AUTRES RESULTATS**

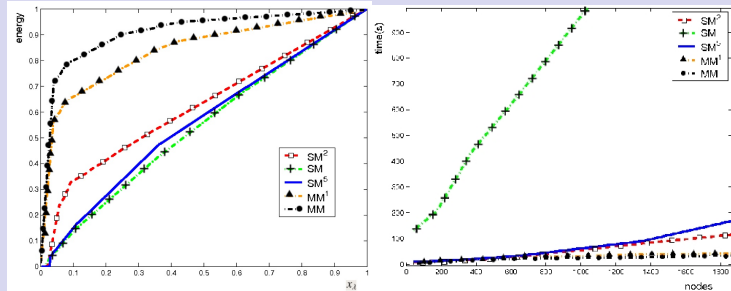


# Conclusion

## Conclusion

- We presented different heuristics to build such hierarchies
  - ▶ sequential ones whose energy is closed from lower bound.
  - ▶ parallel ones providing greater energies but require less execution time (even on sequential machine)

## mean energy and execution times



Thanks for your attention.

## End

