

Segmentation et appariement hiérarchiques basés sur les pyramides combinatoires

Jean-Hugues PRUVOT

Université de Caen Basse-Normandie
GREYC CNRS UMR 6072
Équipe Image

Soutenance de thèse
mercredi 22 octobre 2008 à Caen.



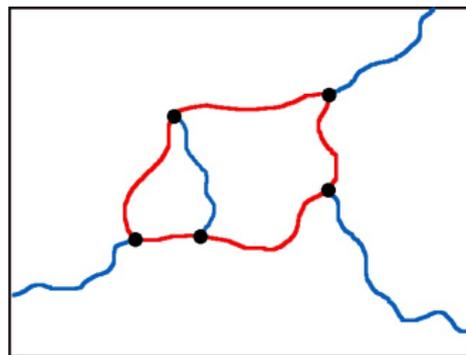
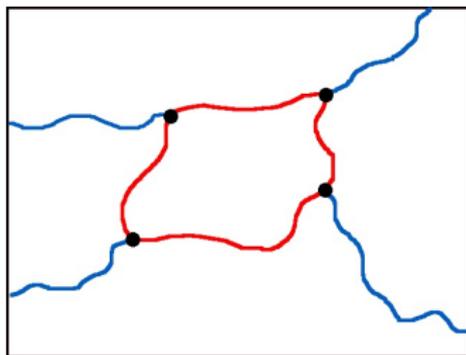
Plan

- 1 **Introduction**
- 2 **Segmentation hiérarchique**
 - Représentations pyramidales
- 3 **Théorie ensemble/échelle**
 - Principe de causalité
 - Coupes optimales
- 4 **Huristiques de fusion**
 - Fusion séquentielles
 - Fusions parallèles
- 5 **Conclusion partielle**

Introduction

Context

- Retrouver un objet ou la plus grande zone commune à deux images
 - Indexation ,reconnaissance/suivi d'objets , reconstruction 3D
- Problème d'appariement
 - appairer des régions
 - inconsistance lors de l'étape de segmentation
problème d'inconsistance \Rightarrow un même objet présent dans 2 images
peut être codé différemment dans les 2 partitions



Introduction

Appariement hiérarchique

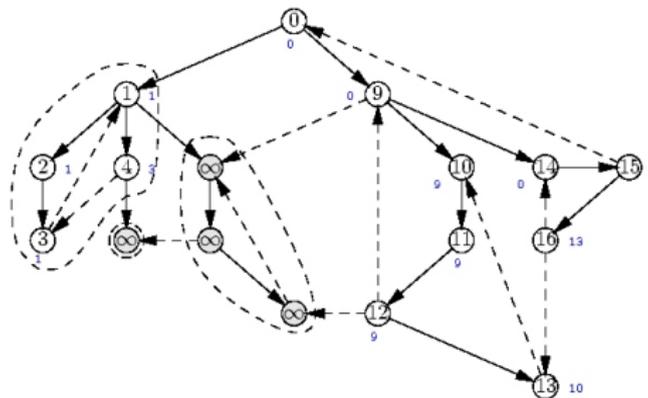
- retrouver des structures communes à deux hiérarchies de partitions
- structures pyramides
⇒ permettent de prendre en compte :
 - les différentes segmentations d'un objet aux différents niveaux de la pyramide



Introduction

Appariement sur le plan

- prendre en compte et mettre à profit la planarité
 - ▶ relations d'adjacence
 - ▶ relations d'inclusion
 - ▶ voisinage orienté



Introduction

Utilisation d'un critère objectif de segmentation

- Pour maximiser les chances d'obtenir les mêmes segmentations des 2 côtés.
 - ▶ cadre énergétique
 - ▶ plusieurs heuristiques de fusions

Codage de Hiérarchies

pyramides régulières

- Pile d'images de résolution décroissante

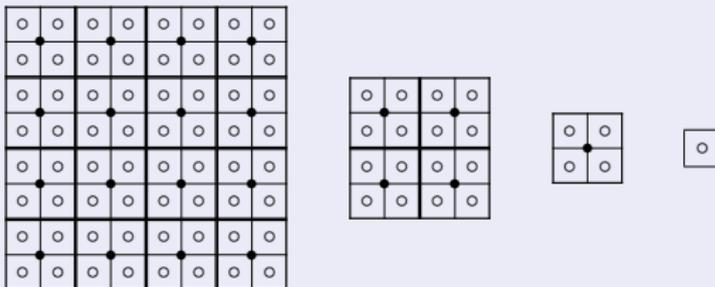


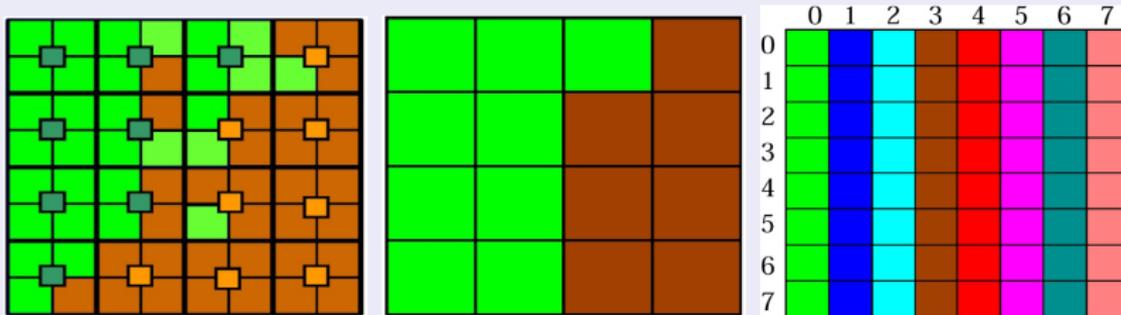
Fig.: Une pyramide régulière $2 \times 2/4$

- ▶ 😊 réduit l'influence du bruit
- ▶ 😊 Rend les traitements indépendants de la résolution
- ▶ 😊 Converti des propriétés globales en propriétés locales
- ▶ 😊 Réduit les coûts de calcul
- ▶ 😊 Analyse d'image a coût réduit en utilisant des images faible résolution

Codage de Hiérarchies

pyramides régulières

- Pile d'images de résolution décroissante

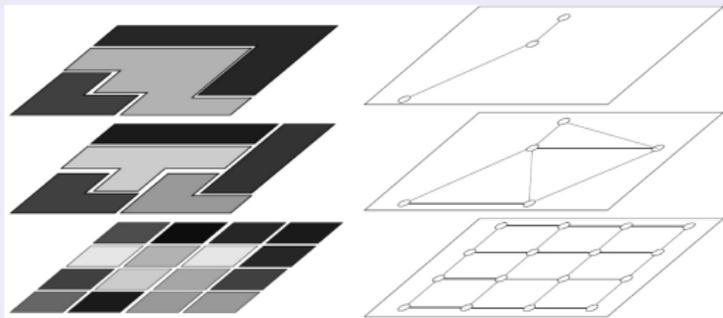


- ▶ ☹ Sensible aux : Décalages - Zooms - Rotations
- ▶ ☹ La préservation de la connexité n'est pas garantie.
- ▶ ☹ Nombre limité de régions à un certain niveau
- ▶ ☹ Difficile de coder les longues régions

Codage de Hiérarchies

pyramides irrégulières de graphes simples

- Pile de graphes progressivement réduits



- ▶ permettent de s'affranchir de la plupart des limitations des pyramides régulières
- ▶ la forme des régions extraites n'est contrainte par aucun critère géométrique

Construction de la hiérarchie

2 approche pour la segmentation

- *Horowitz[1976]* avec un prédicat P
 - ▶ découpe/fusion par rapport à un critère d'homogénéité
 - ▶ critère local pour chaque région
- Schéma de minimisation d'énergie
 - ▶ Level-Set, Min-cut / N-Cut
 - ▶ permet de définir un critère global sur la partition
 - ▶ permet une évaluation objective de la segmentation
 - ▶ \Rightarrow minimisation par rapport à un paramètre d'échelle λ
 - ▶ Level Set / N-Cut/Min-Cut \Rightarrow local minimum in a full search space

Théorie ensemble/échelle

- Approche proposé par Guigues[2001]
 - ▶ recherche d'un minimum global dans un espace de recherche restreint
 - ▶ pra le biais d'une approche ascendante
 - ▶ fournit la partition optimal pour chaque valeur d'un paramètre d'échelle $\lambda \in \mathbb{R}^+$

Principe

- minimiser une énergie de partition $E(P)$:

Energie pour chaque région R_i :

$$E(P) = D(P) + \lambda C(P)$$

$D(R_i)$: énergie interne de la région (attache aux données)

$C(R_i)$: énergie de complexité (régularisation)

λ : paramètre d'échelle

- Modèle constant par morceaux

Théorie ensemble/échelle

- Approche proposé par Guigues[2001]
 - ▶ recherche d'un minimum global dans un espace de recherche restreint
 - ▶ pra le biais d'une approche ascendante
 - ▶ fournit la partition optimal pour chaque valeur d'un paramètre d'échelle $\lambda \in \mathbb{R}^+$

Principe

- minimiser une énergie de partition $E(P)$:
 - ▶ Energie pour chaque région R_i :

$$E(P) = D(P) + \lambda C(P)$$

- ▶ $D(R_i)$: énergie interne de la région (attache aux données)
 - ▶ $C(R_i)$: énergie de complexité (régularisation)
 - ▶ λ : paramètre d'échelle
- Modèle constant par morceaux

énergie utilisée

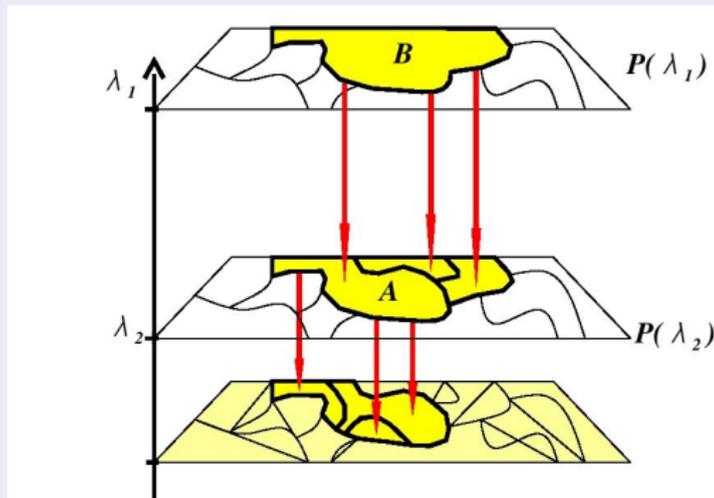
Energy $E_\lambda(P) = \lambda C(P) + D(P)$ *énergie affine*

- D : énergie interne de la région (attache aux données) :
 - ▶ *erreur quadratique* :
$$SE(R) = \sum_{i \in R} \|c_i - \mu_R\|^2$$
 - ▶ \Rightarrow minimal si chaque région contient un pixel
- C : énergie de complexité (régularisation) :
 - ▶ *longueur totale des frontières*
 - ▶ \Rightarrow faible si la segmentation comporte peu de régions

Théorie ensemble/échelle

Principe de causalité

- introduit par Witkin en 1984
 - ▶ des structures cohérentes sont présentes à différentes échelles dans une image



- $\forall (\lambda_1, \lambda_2) \in \mathbb{R}^{+2}$ avec $\lambda_2 \leq \lambda_1 \Rightarrow P_{\lambda_1}$ peut être déduit de P_{λ_2} en

Théorie ensemble/échelle

Principe de causalité

- si P_λ est *causale* , $H = \{P_\lambda(I), \lambda \in \mathbb{R}^+\}$ est une hiérarchie

$$P_\lambda(I) = \min_{p \in \mathcal{P}} D(P) + \lambda C(P)$$

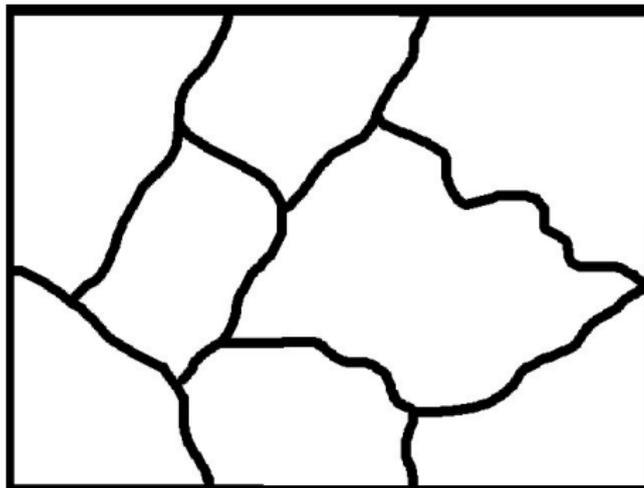
(partition qui minimise $E_\lambda(I)$)

- Guigues \Rightarrow si C est sous-additive alors P est causale
 - ▶ C sous-additive $\iff C(R_1 \cup R_2) \leq C_{R_1} + C_{R_2}$
 - ▶ condition naturelle pour les algorithmes de segmentation (MDL \Rightarrow moins de paramètres pour décrire l'union de 2 régions que 2 régions)

Construction de la hiérarchie

Escalade

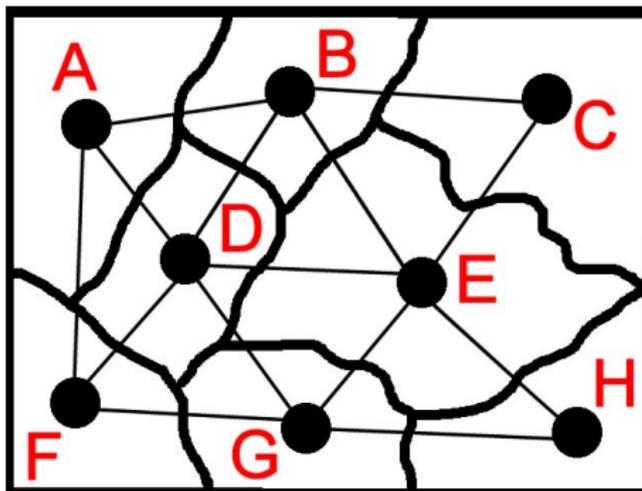
- Partir d'une segmentation initiale



Construction de la hiérarchie

Escalade

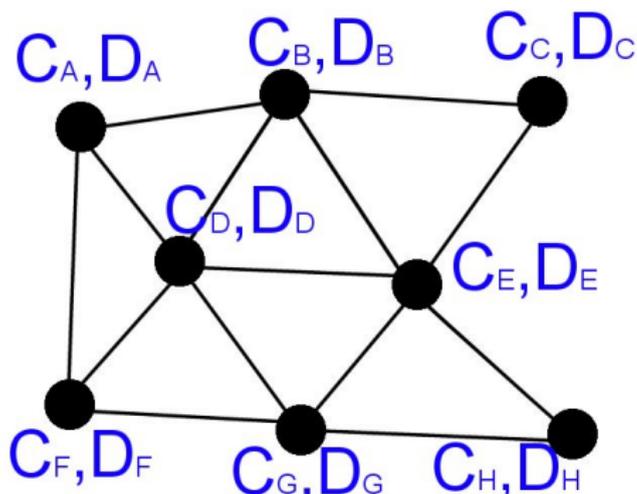
- construire un RAG



Construction de la hiérarchie

Escalade

- Pour chaque région calculer
 - l'énergie interne $D(R_i)$
 - l'énergie de complexité $C(R_i)$

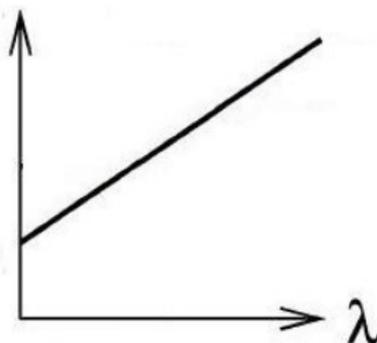


Construction de la hiérarchie

Escalade

- L'énergie de la partition initiale est alors décrite par une fonction affine

$$E(P) = D(P) + \lambda C(P)$$



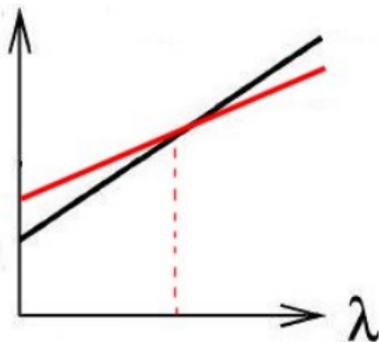
Construction de la hiérarchie

Escalade

- calculer pour chaque fusion possible, la nouvelle énergie et fusionner les régions qui occasionnent la plus petites augmentation d'énergie

$$E(P'_{ij}) = D(P'_{ij}) + \lambda C(P'_{ij})$$

avec P'_{ij} même partition que P avec les régions R_i et R_j fusionnées



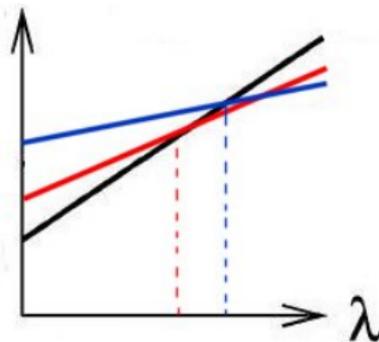
Construction de la hiérarchie

Escalade

- calculer pour chaque fusion possible, la nouvelle énergie et fusionner les régions qui occasionnent la plus petites augmentation d'énergie

$$E(P'_{ij}) = D(P'_{ij}) + \lambda C(P'_{ij})$$

avec P'_{ij} même partition que P avec les régions R_i et R_j fusionnées



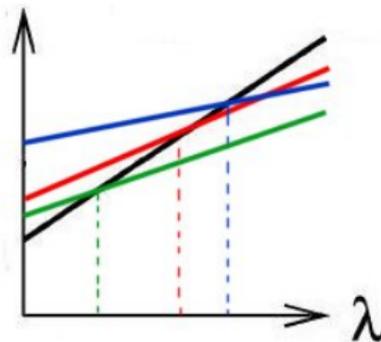
Construction de la hiérarchie

Escalade

- calculer pour chaque fusion possible, la nouvelle énergie et fusionner les régions qui occasionnent la plus petites augmentation d'énergie

$$E(P'_{ij}) = D(P'_{ij}) + \lambda C(P'_{ij})$$

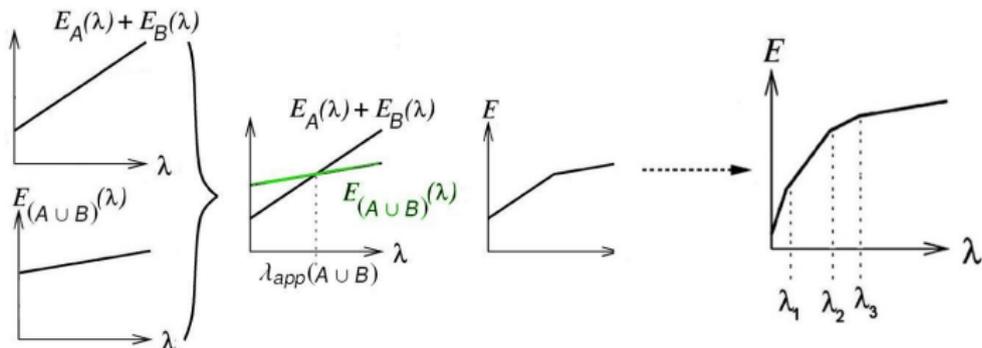
avec P'_{ij} même partition que P avec les régions R_i et R_j fusionnées



Construction de la hiérarchie

Escalade

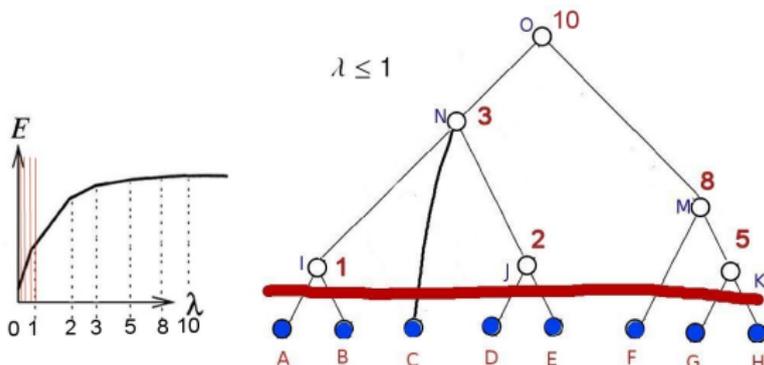
- itérer ce processus pour obtenir une série de λ_{app} qui fournissent un ensemble de coupes optimales dans H
- l'énergie des coupes optimales dans cette hiérarchie est alors décrite par une fonction concave affine par morceaux



Construction de la hiérarchie initiale

Cuts

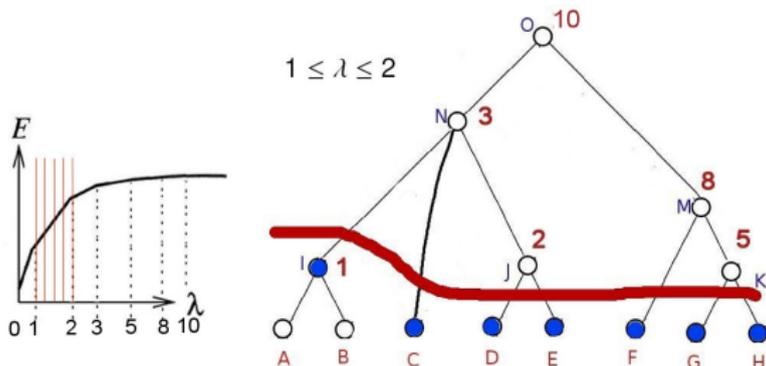
- On peut donc facilement pour un λ donné, retrouver la partition optimale correspondante dans H :



Construction de la hiérarchie initiale

Cuts

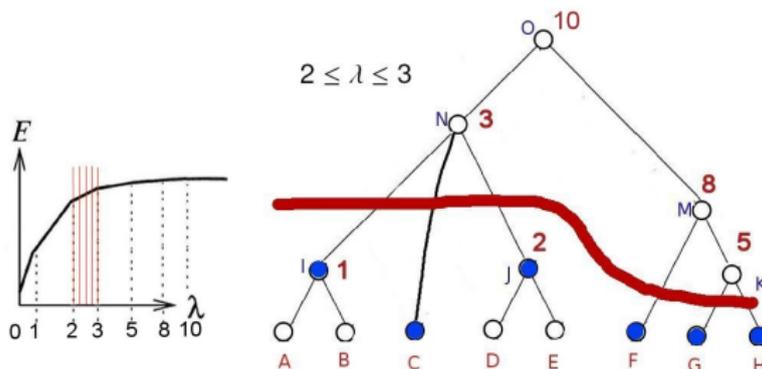
- On peut donc facilement pour un λ donné, retrouver la partition optimale correspondante dans H :



Construction de la hiérarchie initiale

Cuts

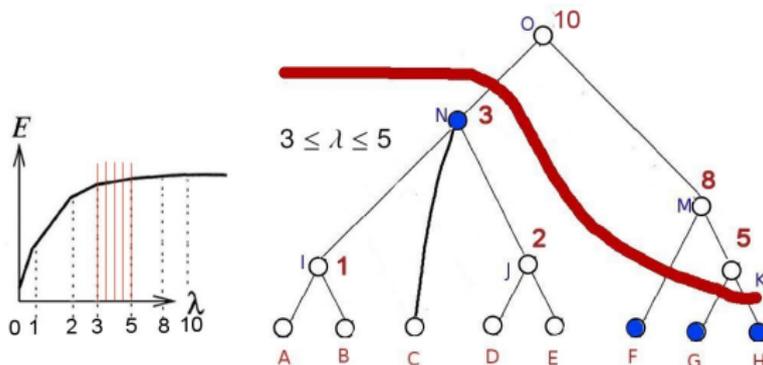
- On peut donc facilement pour un λ donné, retrouver la partition optimale correspondante dans H :



Construction de la hiérarchie initiale

Cuts

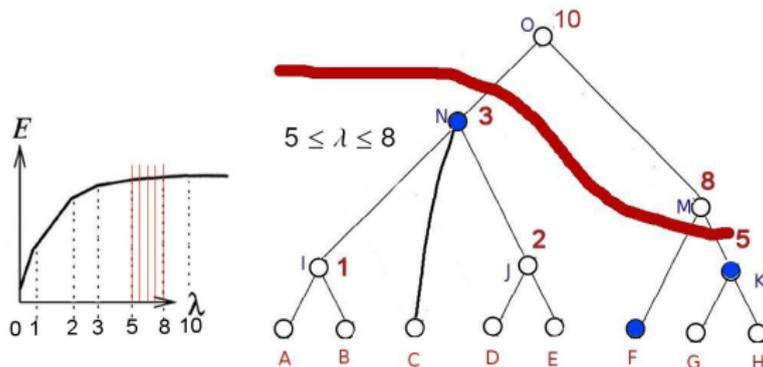
- On peut donc facilement pour un λ donné, retrouver la partition optimale correspondante dans H :



Construction de la hiérarchie initiale

Cuts

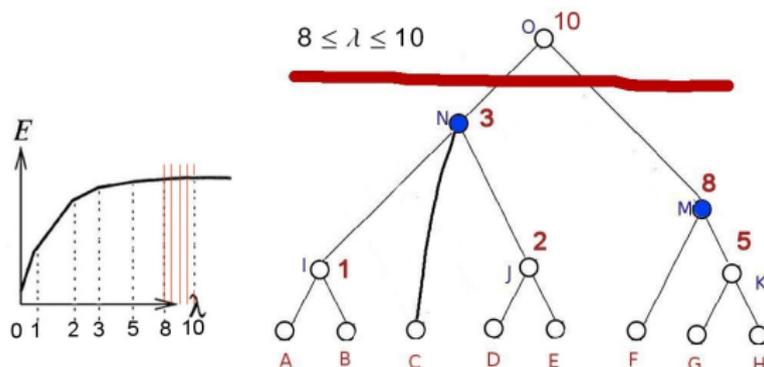
- On peut donc facilement pour un λ donné, retrouver la partition optimale correspondante dans H :



Construction de la hiérarchie initiale

Cuts

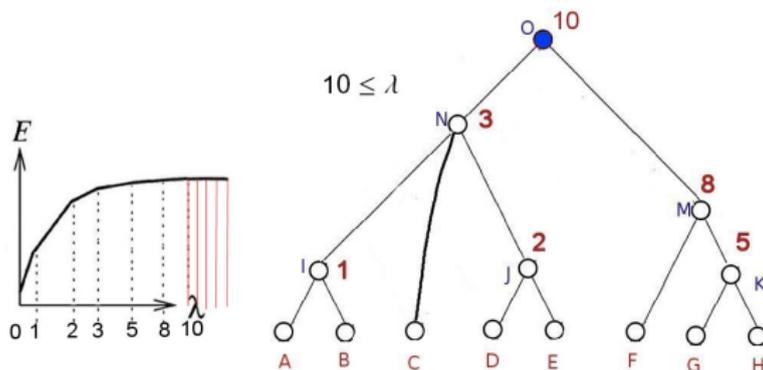
- On peut donc facilement pour un λ donné, retrouver la partition optimale correspondante dans H :



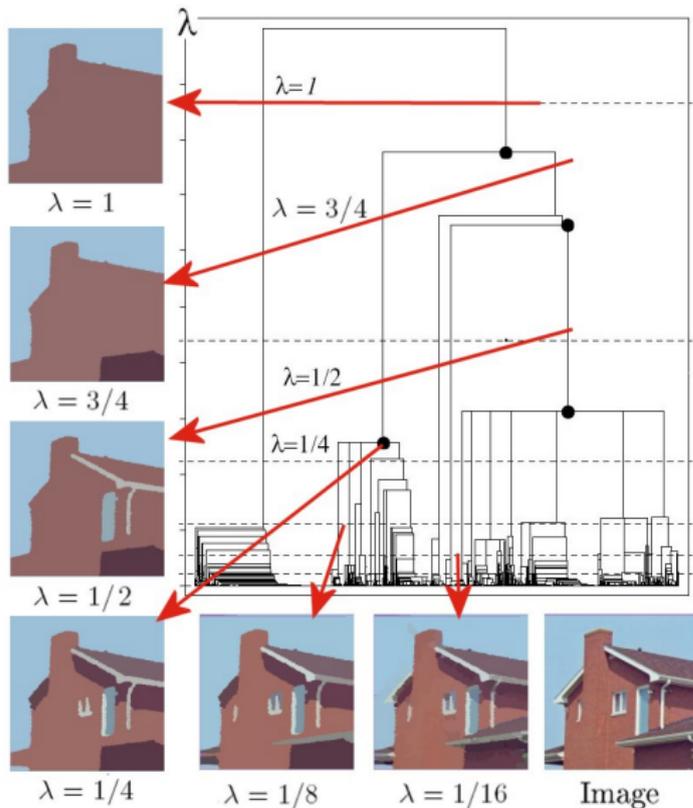
Construction de la hiérarchie initiale

Cuts

- On peut donc facilement pour un λ donné, retrouver la partition optimale correspondante dans H :



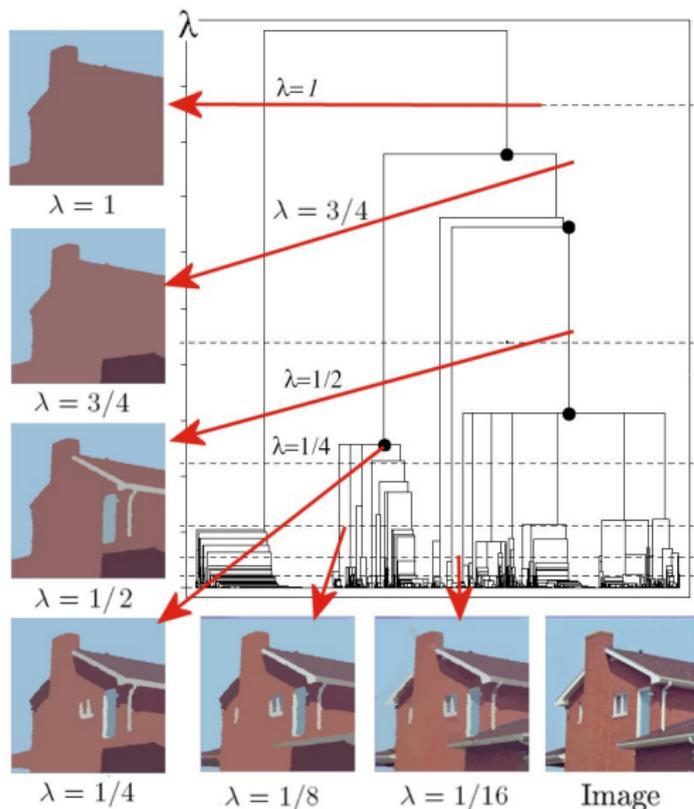
Coupes optimales



Pour résumer

- fournit l'ensemble des solutions pour tout λ
- les solutions données sont optimales pour la hiérarchie correspondant à un espace de recherche restreint
- Les partitions demeurent stables sur des intervalles complets

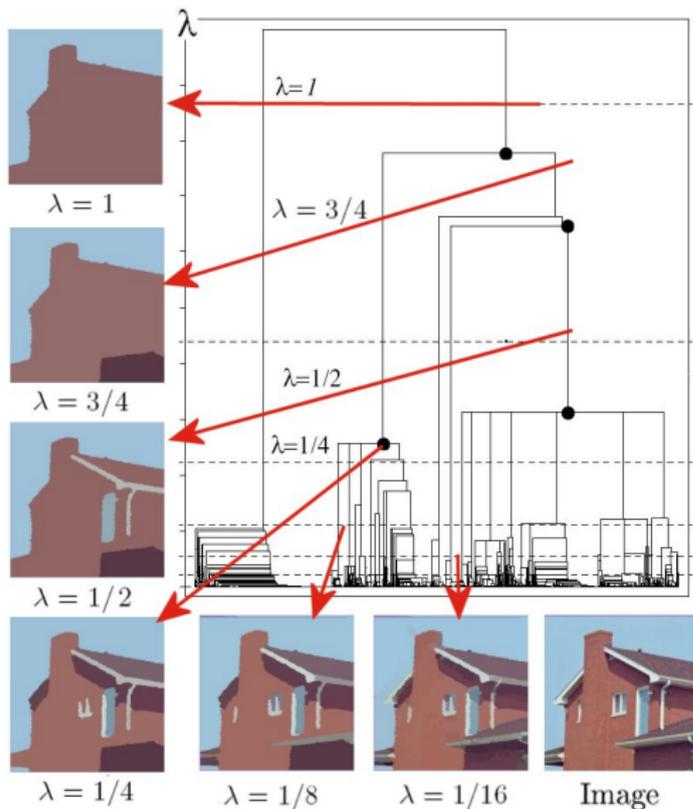
Coupes optimales



Pour résumer

- fournit l'ensemble des solutions pour tout λ
- les solutions données sont optimales pour la hiérarchie correspondant à un espace de recherche restreint
- Les partitions demeurent stables sur des intervalles complets

Coupes optimales



Pour résumer

- fournit l'ensemble des solutions pour tout λ
- les solutions données sont optimales pour la hiérarchie correspondant à un espace de recherche restreint
- Les partitions demeurent stables sur des intervalles complets

Importance de l'espace de recherche

Importance de la hiérarchie

- L'espace de recherche utilisé dans ce cadre est restreint à la partition initiale
 - ▶ Le schéma de construction est donc d'une importance capitale pour les partitions optimales dans H

Importance de l'espace de recherche

Importance de la hiérarchie

- L'espace de recherche utilisé dans ce cadre est restreint à la partition initiale
 - ▶ Le schéma de construction est donc d'une importance capitale pour les partitions optimales dans H

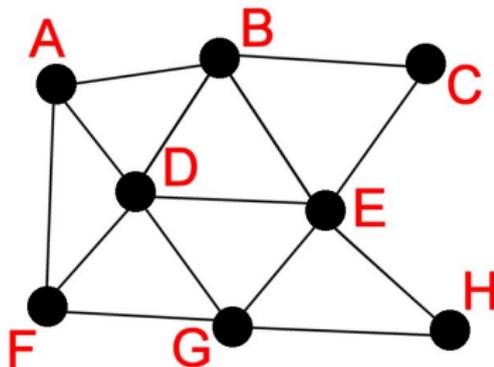
Compromis entre temps/énergie

- fusions séquentielles
- fusions parallèles

Fusion séquentielles

Multi-fusion

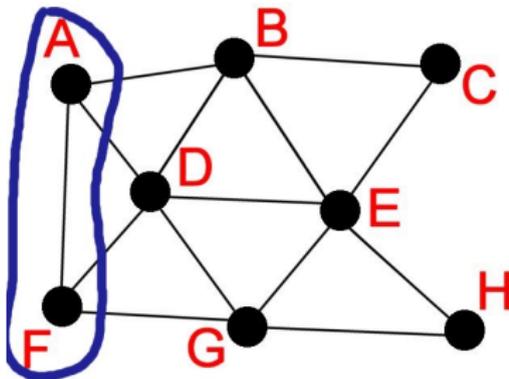
- l'espace de recherche est agrandi en autorisant la fusion simultanée de plus de 2 régions
 - ▶ Nous considérons pour chaque région R de P , son ensemble $V(R)$ défini comme $\{R\}$ union l'ensemble de ses voisins ainsi que l'ensemble $\mathcal{P}^*(V(R))$ définissant tous les sous ensembles possible de $V(R)$ incluant R including R



Fusion séquentielles

Multi-fusion

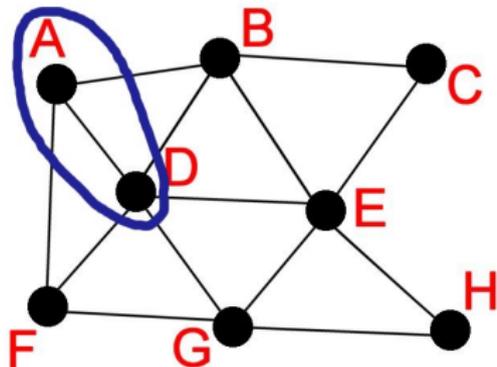
- l'espace de recherche est agrandi en autorisant la fusion simultanée de plus de 2 régions
 - ▶ Nous considérons pour chaque région R de P , son ensemble $V(R)$ défini comme $\{R\}$ union l'ensemble de ses voisins ainsi que l'ensemble $\mathcal{P}^*(V(R))$ définissant tous les sous ensembles possible de $V(R)$ incluant R including R



Fusion séquentielles

Multi-fusion

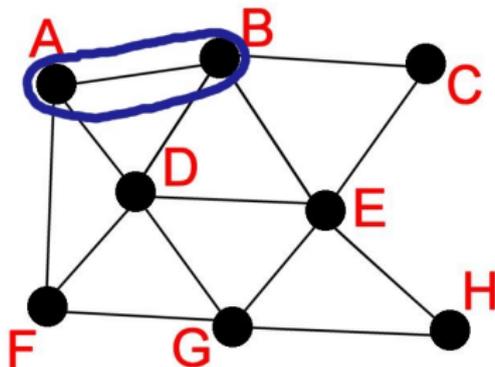
- l'espace de recherche est agrandi en autorisant la fusion simultanée de plus de 2 régions
 - ▶ Nous considérons pour chaque région R de P , son ensemble $V(R)$ défini comme $\{R\}$ union l'ensemble de ses voisins ainsi que l'ensemble $\mathcal{P}^*(V(R))$ définissant tous les sous ensembles possible de $V(R)$ incluant R including R



Fusion séquentielles

Multi-fusion

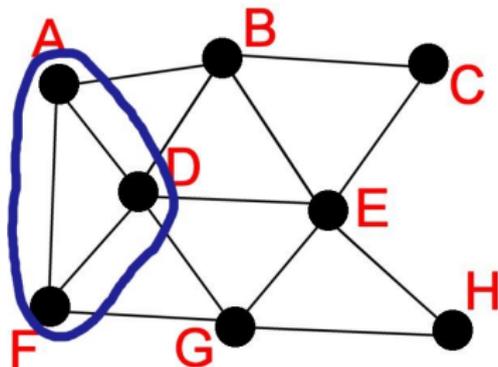
- l'espace de recherche est agrandi en autorisant la fusion simultanée de plus de 2 régions
 - Nous considérons pour chaque région R de P , son ensemble $V(R)$ défini comme $\{R\}$ union l'ensemble de ses voisins ainsi que l'ensemble $\mathcal{P}^*(V(R))$ définissant tous les sous ensembles possible de $V(R)$ incluant R including R



Fusion séquentielles

Multi-fusion

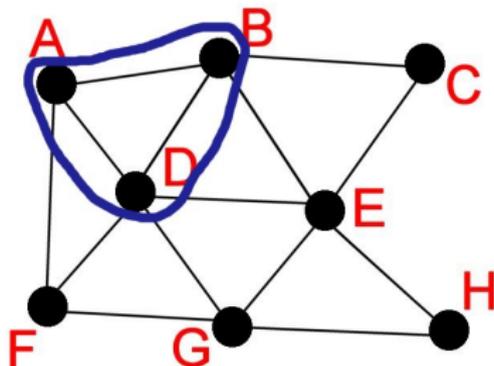
- l'espace de recherche est agrandi en autorisant la fusion simultanée de plus de 2 régions
 - ▶ Nous considérons pour chaque région R de P , son ensemble $V(R)$ défini comme $\{R\}$ union l'ensemble de ses voisins ainsi que l'ensemble $\mathcal{P}^*(V(R))$ définissant tous les sous ensembles possible de $V(R)$ incluant R including R



Fusion séquentielles

Multi-fusion

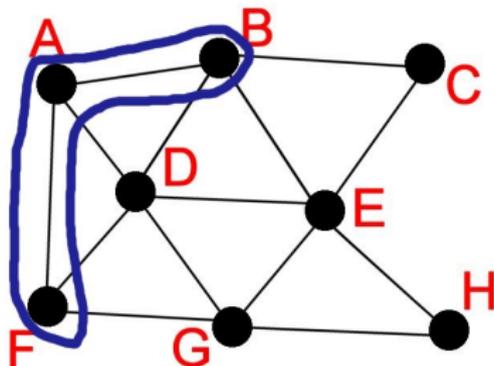
- l'espace de recherche est agrandi en autorisant la fusion simultanée de plus de 2 régions
 - ▶ Nous considérons pour chaque région R de P , son ensemble $V(R)$ défini comme $\{R\}$ union l'ensemble de ses voisins ainsi que l'ensemble $\mathcal{P}^*(V(R))$ définissant tous les sous ensembles possible de $V(R)$ incluant R including R



Fusion séquentielles

Multi-fusion

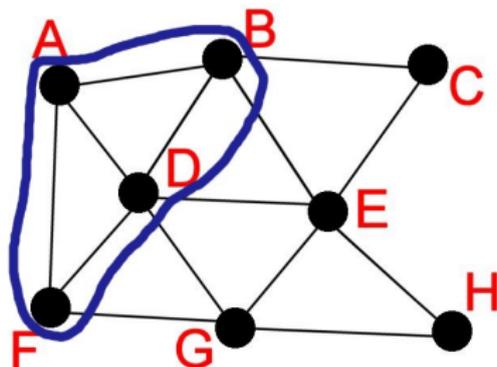
- l'espace de recherche est agrandi en autorisant la fusion simultanée de plus de 2 régions
 - Nous considérons pour chaque région R de P , son ensemble $V(R)$ défini comme $\{R\}$ union l'ensemble de ses voisins ainsi que l'ensemble $\mathcal{P}^*(V(R))$ définissant tous les sous ensembles possible de $V(R)$ incluant R including R



Fusion séquentielles

Multi-fusion

- l'espace de recherche est agrandi en autorisant la fusion simultanée de plus de 2 régions
 - ▶ Nous considérons pour chaque région R de P , son ensemble $V(R)$ défini comme $\{R\}$ union l'ensemble de ses voisins ainsi que l'ensemble $\mathcal{P}^*(V(R))$ définissant tous les sous ensembles possible de $V(R)$ incluant R including R



Fusion séquentielles

Multi-fusion

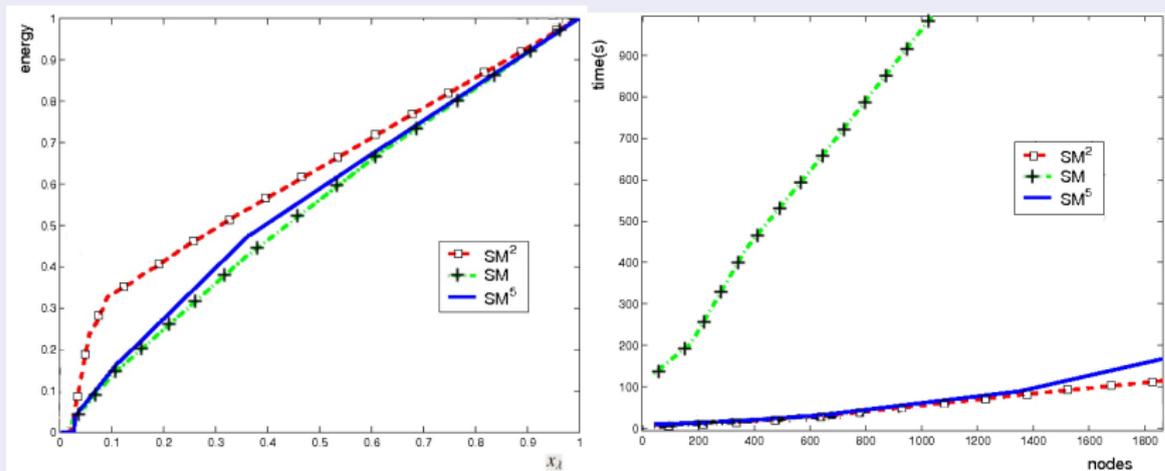
- Utilisant le principe d'excalade, notre algorithme séquentiel calcul pour chaque région R de la partition l'échelle d'apparition minimum pour une région R^W

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

- la complexité est bornée par $O(|V|2^k)$
avec $|V|$ représentant le nombre de sommets (i.e régions)
et k le degré maximum des sommets dans G .
 - ▶ - peut induire des temps de calcul importants
 - ▶ + le cardinal du sous-ensemble de région à fusionner peut être réduit à 5 sans altéré de façon significative l'énergie des coupes optimales

Fusion séquentielles

Energies moyennes et temps d'exécution



 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

$\lambda = 0.2$

$\lambda = 0.4$

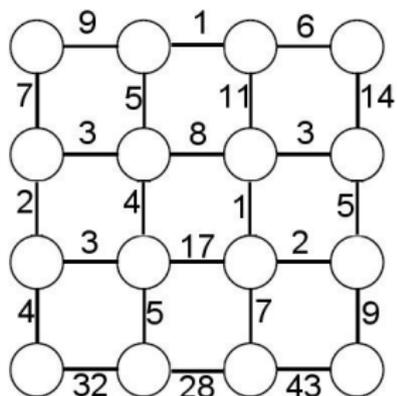
$\lambda = 0.6$

$\lambda = 0.8$

Algorithmes de fusions parallèles

Fusions parallèles

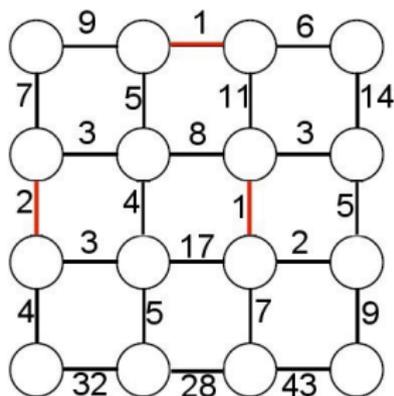
- 2 algorithmes basées sur la notion de *maximal matching*
- utilisant la même approches que [Haxhimusa]
⇒ Ensemble indépendant maximal (MIS) basé sur l'ensemble des arêtes du graphe



Algorithmes de fusions parallèles

Fusions parallèles

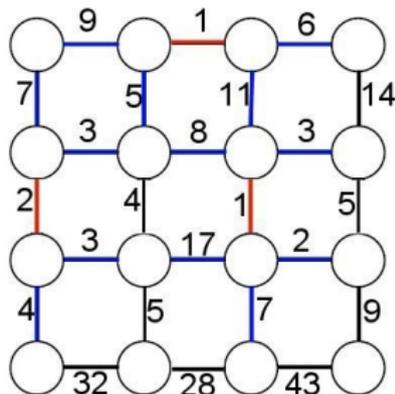
- 2 algorithmes basées sur la notion de *maximal matching*
- utilisant la même approches que [Haxhimusa]
⇒ Ensemble indépendant maximal (MIS) basé sur l'ensemble des arêtes du graphe



Algorithmes de fusions parallèles

Fusions parallèles

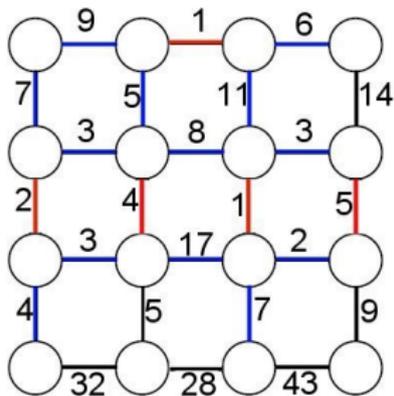
- 2 algorithmes basées sur la notion de *maximal matching*
- utilisant la même approches que [Haxhimusa]
 ⇒ Ensemble indépendant maximal (MIS) basé sur l'ensemble des arêtes du graphe



Algorithmes de fusions parallèles

Fusions parallèles

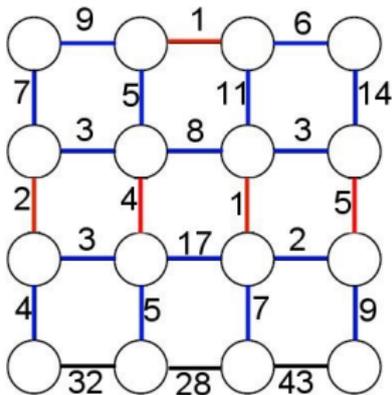
- 2 algorithmes basées sur la notion de *maximal matching*
- utilisant la même approches que [Haxhimusa]
 ⇒ Ensemble indépendant maximal (MIS) basé sur l'ensemble des arêtes du graphe



Algorithmes de fusions parallèles

Fusions parallèles

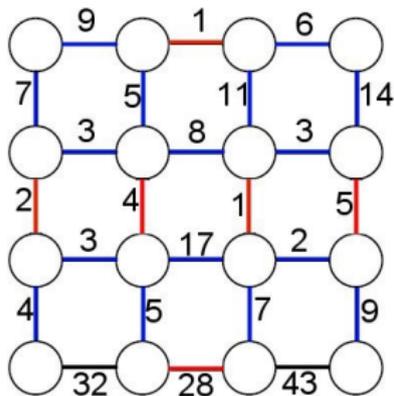
- 2 algorithmes basées sur la notion de *maximal matching*
- utilisant la même approches que [Haxhimusa]
 ⇒ Ensemble indépendant maximal (MIS) basé sur l'ensemble des arêtes du graphe



Algorithmes de fusions parallèles

Fusions parallèles

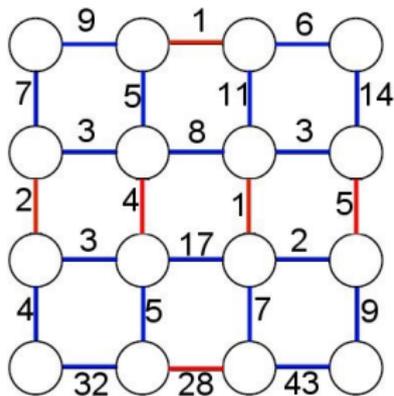
- 2 algorithmes basées sur la notion de *maximal matching*
- utilisant la même approches que [Haxhimusa]
 ⇒ Ensemble indépendant maximal (MIS) basé sur l'ensemble des arêtes du graphe



Algorithmes de fusions parallèles

Fusions parallèles

- 2 algorithmes basées sur la notion de *maximal matching*
- utilisant la même approche que [Haxhimusa]
 ⇒ Ensemble indépendant maximal (MIS) basé sur l'ensemble des arêtes du graphe



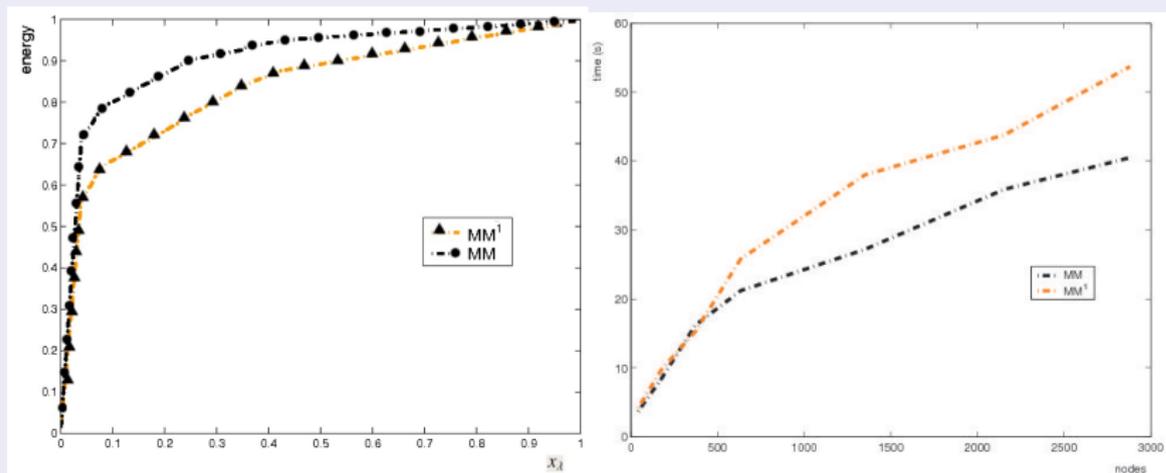
Fusions parallèles

Algos *MM* et *MM*¹

- *MM*
 - ▶ + Taux de décimation
 - ▶ - Les minima détectés sont de moins en moins pertinents au fur et à mesure des itérations
- nous proposons une solution alternative *MM*¹
- ⇒ contracter à chaque étapes uniquement les arêtes sélectionnées lors de la première itération
- ⇒ peut être interprété comme une combinaison de la méthode proposée par [Haxhimusa2003] et le processus de décimation stochastique de [Jolion2001]
 - ▶ + fusionne immédiatement uniquement les arêtes correspondant à des minima locaux
 - ▶ + les taux de decimation moyen de 1.73 sur les 100 images de la base de Berkeley (comparable au 2.0 obtenu par Haxhimusa)

Fusions parallèles

Energies moyennes et temps d'exécution



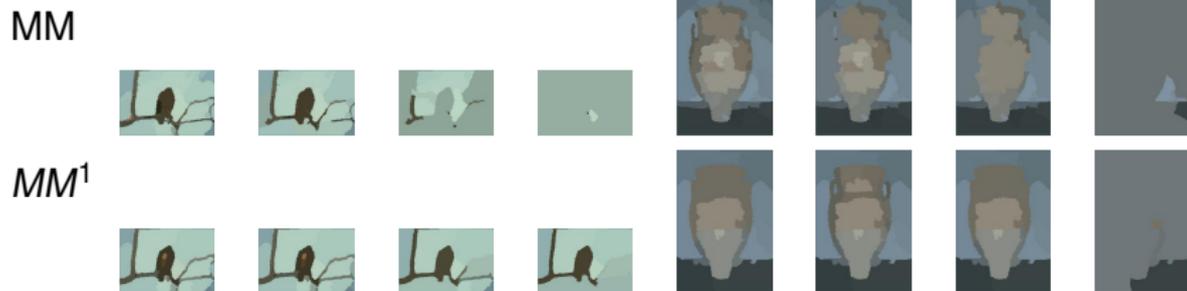
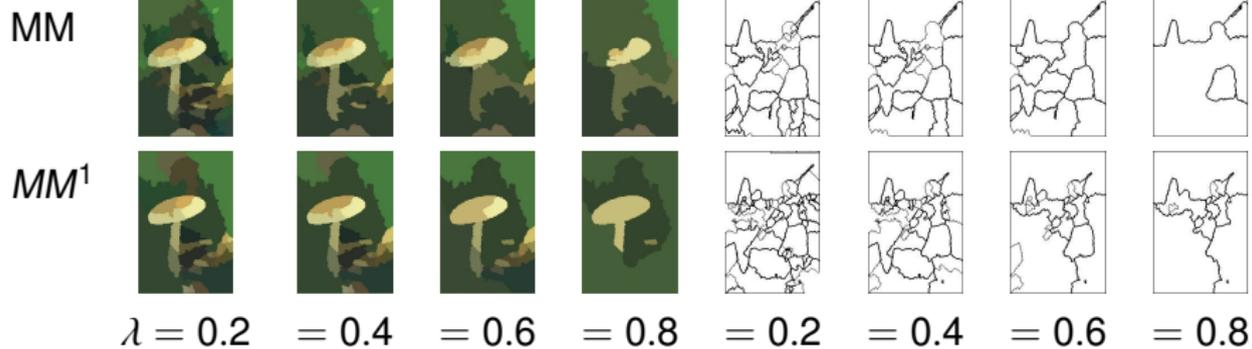
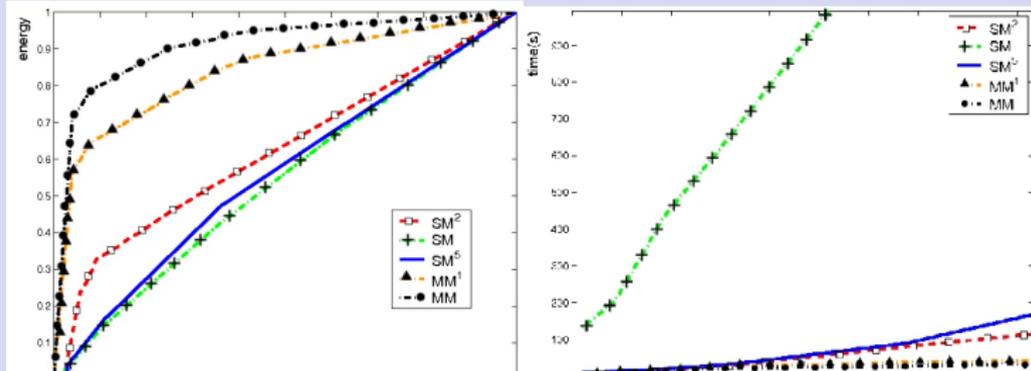


Schéma de construction de hiérarchie

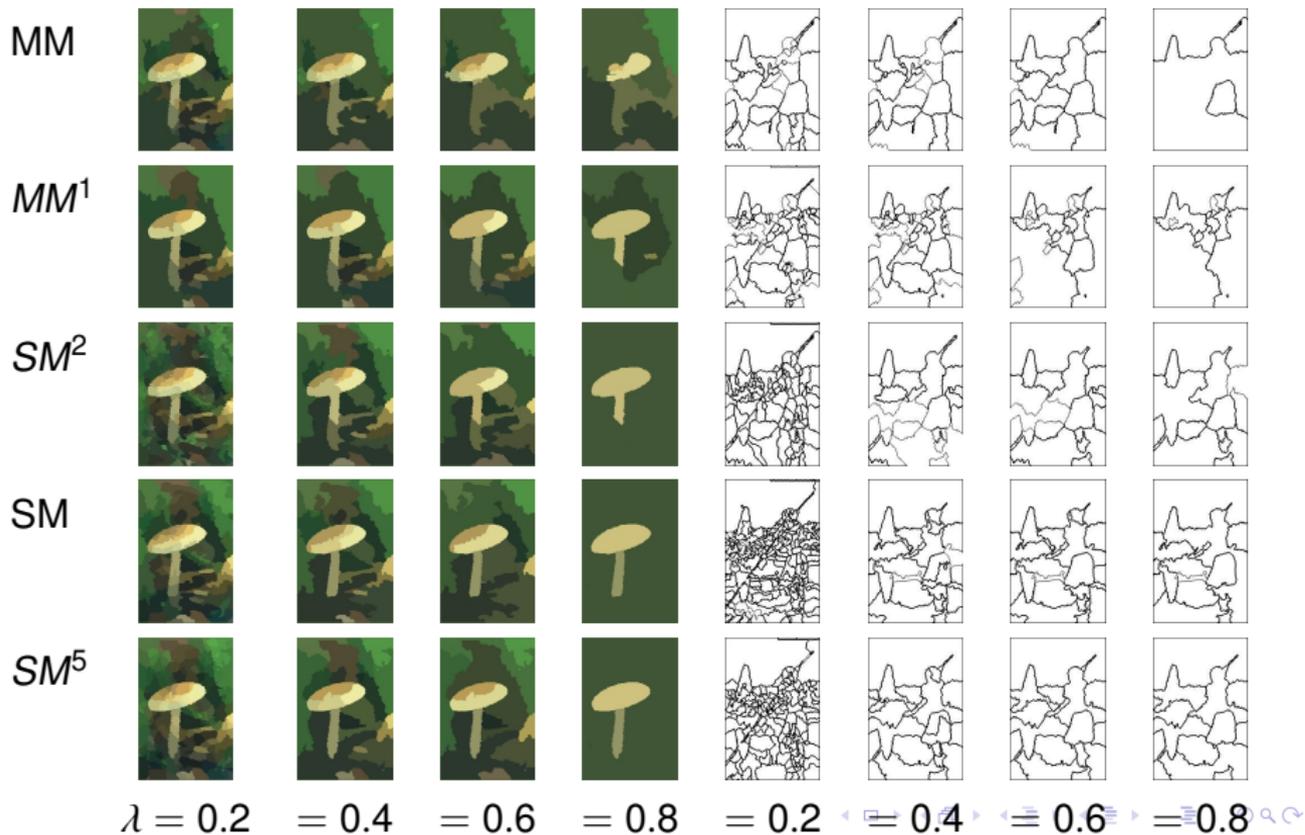
Conclusion partielle

- Nous avons présenté différentes heuristiques de fusion permettant de construire de hiérarchies
 - ▶ séquentielle permettant d'obtenir des énergies proches de la borne inférieure
 - ▶ parallèle fournissant des énergies supérieures mais avec des temps d'exécution inférieurs (même sur des machines séquentielles)

Energies moyennes et temps d'exécution



Résultats



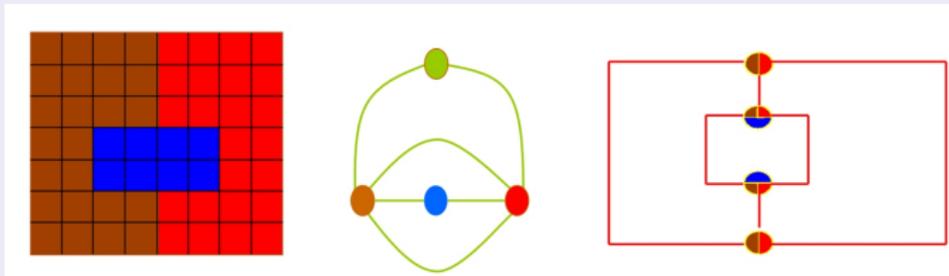
Codage de Hiérarchies

Pour palier aux limitation des graphes simple dans le cas du codage de partition :

- 😞 absence de boucle \Rightarrow ne permet pas de caractériser les relations d'inclusion
- 😞 une seule arête entre 2 sommets \Rightarrow réduit l'information de frontière à une simple relation d'adjacence (présence d'au moins une frontière)

pyramides irrégulieres de graphes duaux

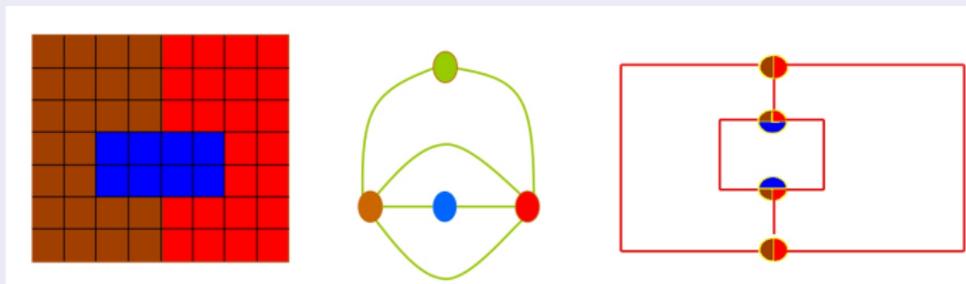
- Deux graphes codant les relations entre les régions et les segments



Codage de Hiérarchies

pyramides irrégulières de graphes duaux

- Deux graphes codant les relations entre les régions et les segments



- 😊 [Kropatsch96] Code les propriétés des nœuds et des faces
- 😞 [BrinKropatsch00] Nécessite le stockage et la mise à jour de deux structures de données.

Notre représentation pyramidale

Pyramide combinatoire

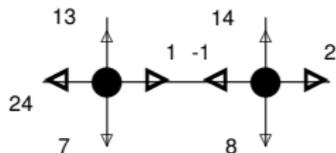
Encodage implicite de

- 😊 la hiérarchie
- 😊 l'orientation
- 😊 du voisinage
- 😊 nécessite peu d'espace mémoire

$\mathcal{P} = (G_0, G_1 \dots, G_n)$: Une pile de cartes combinatoires successivement réduites

La carte combinatoire initiale G_0 code la grille de pixels initiale

Cartes combinatoires



$$G = (D, \sigma, \alpha)$$

$$\alpha = (1, -1) \dots$$

$$\sigma = (13, 24, 7, 1)(14, -1, 8, 2) \dots$$

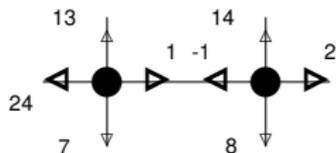
- Découper chaque arête en deux **brins**. D représente l'ensemble de ces brins.

$$\alpha = (1, -1) \dots$$

$$\varphi = \sigma \circ \alpha$$

$$\varphi = (13, -14, -1)(1, 8, -7) \dots$$

Cartes combinatoires

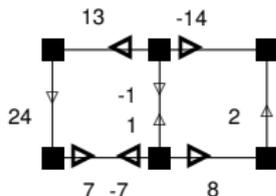


$$\alpha = (1, -1) \dots$$

$$\sigma = (13, 24, 7, 1)(14, -1, 8, 2) \dots$$

$$G = (D, \sigma, \alpha)$$

$$\overline{G} = (D, \varphi, \alpha)$$



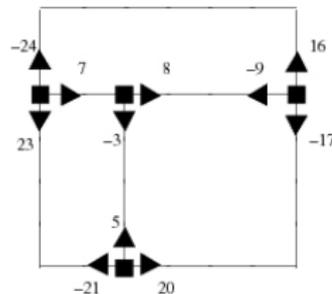
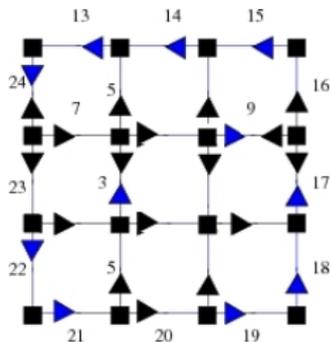
$$\alpha = (1, -1) \dots$$

$$\varphi = \sigma \circ \alpha$$

$$\varphi = (13, -14, -1)(1, 8, -7) \dots$$

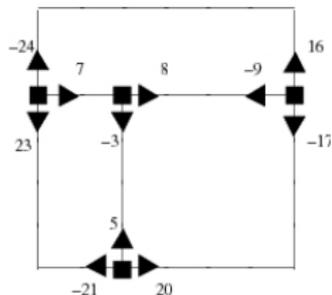
Brins et Segments

- $G_i = (D_i, \sigma_i, \alpha_i)$, $d \in D_i$.
- d code une frontière entre $\sigma_i^*(d)$ et $\sigma_i^*(\alpha(d))$.
- Le plongement de d est défini comme une séquence de brins (s_d).



- $s_{16} = 16.15.14.13.24$: Une séquence de brin orientés.

Orientation



- $\sigma^*(16) = (16, 7, 8)$ définit une traversée des frontières de la région dans le sens inverse des aiguilles d'une montre.