

## **Algorithmes de segmentation hiérarchique basés sur Dissemblance Felzenszwalb-Huttenlocher**

Edward Jorge Yuri Cayllahua Cahuina, Jean Cousty, Yukiko Kenmochi,  
Arnaldo de Albuquerque Araujo, Guillermo Cámara-Chávez

► **Pour citer cette version:**

Edward Jorge Yuri Cayllahua Cahuina, Jean Cousty, Yukiko Kenmochi, Arnaldo de Albuquerque Araujo, Guillermo Cámara-Chávez.  
Algorithmes de segmentation hiérarchique basés sur la dissemblance Felzenszwalb-Huttenlocher. Conférence internationale sur la  
reconnaissance de formes et l'intelligence artificielle, mai 2018, Montréal, Canada. ??? hal-01710920 ???

**Id HAL: hal-01710920**

**<https://hal-upec-upem.archives-ouvertes.fr/hal-01710920>**

Soumis le 1 mars 2018

**HAL** est un libre accès multidisciplinaire archives pour le dépôt et la diffusion de documents scientifiques destinés au dépôt et à la diffusion de documents de recherche, qu'ils soient publics de niveau recherche, publiés ou non, liés ou non. Les documents peuvent provenir des émanant des établissements d'enseignement et des établissements d'enseignement et de recherche en France ou de recherche français ou étrangers, des laboratoires à l'étranger, ou de centres de recherche publics ou privés. publics ou privés.

L'archive ouverte pluridisciplinaire **HAL**, est

# Algorithmes de segmentation hiérarchique basés sur la dissemblance Felzenszwalb-Huttenlocher

Edward Cayllahua Cahuina <sup>\*</sup>, Jean Cousty <sup>\*</sup>, Yukiko Kenmochi <sup>\*</sup>, Arnaldo de Albuquerque Araujo <sup>‡</sup>

Guillermo Cámara-Chávez <sup>§</sup>

<sup>\*</sup> Université Paris-Est, LIGM, ESIEE Paris - CNRS, France

<sup>‡</sup> Université Paris Descartes, Laboratoire MAP5, UMR CNRS 8145, France

<sup>‡</sup> Universidade Federal de Minas Gerais, Département d'informatique, Brésil

<sup>§</sup> Universidade Federal de Ouro Preto, Département d'informatique, Brésil

**Abstrait**—La segmentation hiérarchique des images fournit une région-échelle-espace orienté, *c'est à dire*, un ensemble de segmentations d'image à différents niveaux de détail dans lequel les segmentations à des niveaux plus fins sont imbriquées par rapport à celles à des niveaux plus grossiers. La plupart des algorithmes de segmentation d'image, tels que les algorithmes de fusion de régions, reposent sur un critère de fusion qui ne conduit pas à une hiérarchie. Guimarães *et coll.*

a proposé en 2012 une méthode de segmentation d'image hiérarchique basée sur des graphes reposant sur un critère popularisé par Felzenszwalb et Huttenlocher en 2004, hiérarchisant ainsi la méthode populaire Felzenszwalb-Huttenlocher. Cependant, Guimarães *et coll.*

n'a pas fourni d'algorithme pour calculer la hiérarchie proposée. Nous proposons une série d'algorithmes pour calculer le résultat de cette méthode de segmentation d'image basée sur un graphe hiérarchique. Pour une image de taille 321 × 481 pixels, l'algorithme le plus efficace produit le résultat en une demi-seconde alors que le plus naïf demande plus de quatre heures.

## INTRODUCTION

Une segmentation d'image hiérarchique est une série de segmentations d'image à différents niveaux de détail où les segmentations à des niveaux de détail plus élevés sont produites en fusionnant des régions à partir de segmentations à des niveaux de détail plus fins. Par conséquent, les régions à des niveaux de détail plus fins sont imbriquées dans des régions à des niveaux plus grossiers. Une segmentation d'image hiérarchique est illustrée sur la figure 1.

La segmentation hiérarchique des images offre une approche multi-échelles de l'analyse d'images. L'analyse d'image hiérarchique a été lancée par [1] et a reçu beaucoup d'attention depuis lors, comme en témoigne la popularité de [2]. Dans [3], les informations globales sont utilisées pour créer les régions initiales, puis le processus de fusion de régions est traité comme une série de problèmes d'optimisation. La morphologie mathématique est également utilisée dans l'analyse d'image hiérarchique avec, *par exemple*, bassins versants hiérarchiques [4, 5], arbres de partition binaires [6], théorie des ensembles d'échelle [7] ou hiérarchies de zones quasi-plates [8].

Dans [9] (voir [10] pour sa version préliminaire), la hiérarchie de zone quasi plate est utilisée pour effectuer une segmentation hiérarchique de l'image. Ce travail s'appuie sur l'algorithme de segmentation d'image basé sur le graphe (GB) proposé dans [11]. L'algorithme GB utilise un prédicat de fusion pour décider si, à un certain paramètre d'échelle, deux régions adjacentes d'une image doivent être fusionnées en une seule, produisant ainsi une image segmentée. Dans sa forme originale, l'algorithme GB ne conduit pas directement à une segmentation hiérarchique de l'image. Dans [9], le prédicat de fusion de l'algorithme GB est utilisé avec la hiérarchie de zone quasi-plate pour produire une version hiérarchique de la méthode GB appelée HGB.

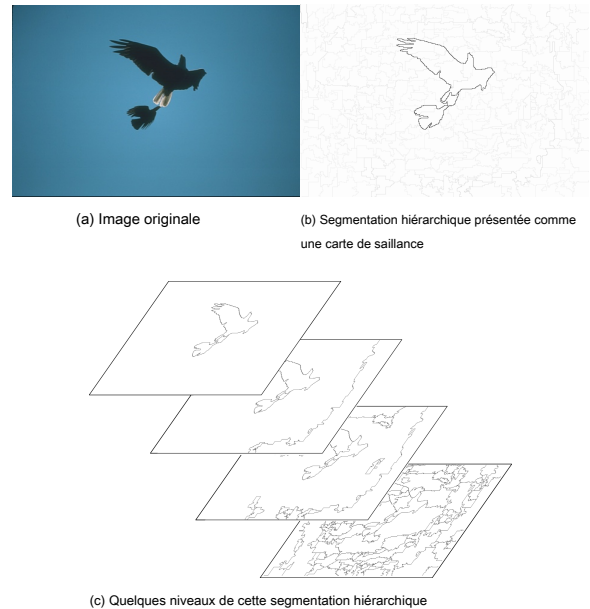


Fig. 1: Illustration d'une segmentation hiérarchique d'image.

La méthode HGB produit une segmentation satisfaisante résultats (voir [9]). Néanmoins, un algorithme précis pour calculer efficacement le résultat de la méthode HGB n'est pas fourni dans [9]. Le cœur de la méthode HGB est basé sur la résolution d'un problème de minimisation dont la solution est l'échelle d'observation minimale à laquelle les régions adjacentes de l'image doivent être fusionnées. Pour résoudre cette minimisation, la méthode considère toutes les valeurs réelles positives pour trouver une telle échelle d'observation minimale.

Dans cet article, nous étudions la méthode HGB et nous nous concentrons sur deux problèmes qui rendent difficile sa mise en œuvre. Une première difficulté est liée à la résolution du problème de minimisation de la méthode HGB pour lequel une solution algorithmique précise n'est pas donnée dans [9]. Nous analysons ce processus de minimisation et proposons trois algorithmes qui le résolvent. Le premier résout la minimisation en recherchant le résultat dans un grand espace de valeurs possibles, nous réduisons ensuite cet espace de recherche pour éviter les calculs redondants, conduisant à deux algorithmes efficaces. Le deuxième problème est lié au calcul de la zone quasi plate. Une approche peut

être d'utiliser un algorithme efficace, tel que [12], pour le calculer à chaque étape de la méthode HGB. Cependant, l'efficacité peut être améliorée en ne mettant à jour qu'à chaque itération la hiérarchie de zone quasi plate existante au lieu de la recalculer à partir de zéro. Ceci est fait avec une procédure similaire à celle proposée dans [13, 14]. Dans l'ensemble, l'algorithme proposé le plus efficace calcule le résultat de la méthode HGB pour une image de taille 321

× 481 pixels en une demi-seconde environ alors que cela prend plus de quatre heures avec l'algorithme le plus naïf.

## II. H SEGMENTATION D'IMAGE BASÉE SUR UN GRAPHIQUE HIÉRARCHIQUE

Cette section vise à expliquer la méthode de segmentation hiérarchique d'images basée sur des graphes (HGB) [10]. La hiérarchie est construite à partir d'une image via une représentation graphique, basée sur la notion de hiérarchie de zones quasi-plate [8]. Nous donnons d'abord une série de notions nécessaires, puis expliquons la méthode HGB.

### A. Notions de base

1) *Hiérarchies*: Étant donné un ensemble fini  $V$ , une *cloison* de  $V$  est un ensemble  $P$  de sous-ensembles disjoints non vides de  $V$  dont l'union est

$V$ . Tout élément de  $P$  est un appelé un *Région* de  $P$ . Étant donné deux partitions  $P$  et  $P'$  de  $V$ ,  $P'$  est dit être un raffinement de  $P$ , désigné par  $P' \leq P$ , si une région de  $P'$  est inclus dans une région de  $P$ . Une hiérarchie sur  $V$  est une séquence  $H = (P_0, \dots, P_n)$  de partitions de  $V$ , tel que  $P_{j+1} \leq P_j$ , pour toute  $j \in \{0, \dots, n-1\}$ .

2) *Partition graphique et composants connectés*: UNE *graphique* est une paire  $G = (V, E)$  où  $V$  est un ensemble fini et  $E$  est un sous-ensemble de  $\{\{x, y\} \subseteq V \mid x \neq y\}$ . Chaque élément de  $V$  s'appelle un *sommet* de  $G$ , et chaque élément de  $E$  s'appelle un *bord* de  $G$ . Un sous-graphe de  $G$  est un graphique  $(V', E')$  tel que  $V' \subseteq V$  et  $E' \subseteq E$ . Si  $X$  est un graphe, ses ensembles de sommets et d'arêtes sont désignés par  $V(X)$  et  $E(X)$ , respectivement.

Laisser  $x$  et  $y$  être deux sommets d'un graphe  $G$ . UNE *chemin* de  $x$  à  $y$  dans  $G$  est une séquence  $(x_0, \dots, x_m)$  de sommets de  $G$  tel que  $x_0 = x$ ,  $x_m = y$  et  $\{x_{j-1}, x_j\}$  est un bord de  $G$  pour toute  $j$  dans  $\{1, \dots, m\}$ . Le graphique  $G$  est *connecté* si, pour tous les sommets  $x$  et  $y$  de  $G$ , il existe un chemin depuis  $x$  à  $y$ . Laisser  $UNE$  être un sous-ensemble de  $V(G)$ . Le graphe induit par  $UNE$  dans  $G$  est le graphe dont l'ensemble de sommets est  $UNE$  et dont le jeu d'arêtes contient n'importe quel bord de  $G$  composé de deux éléments en  $UNE$ . Si le graphe induit par  $UNE$  est connecté, alors on dit que  $UNE$  est *connecté*. Le sous-ensemble  $UNE$  de

$V(G)$  est un *composant connecté* de  $G$  s'il est connecté pour  $G$  et maximal pour cette propriété. Nous désignons par  $C(G)$  l'ensemble de tous les composants connectés de  $G$ . Notez que  $C(G)$  est une partition de  $V(G)$ , qui s'appelle le *partition* des composants connectés induit par  $G$ .

3) *Hiérarchies de zones quasi plates*: Présentons maintenant les hiérarchies de zones quasi-plates qui fournissent une bijection entre un graphe pondéré par les bords et une hiérarchie (voir plus de détails dans [15]).

Étant donné un graphique  $G = (V, E)$ , laisser  $w$  être une carte de  $E$  dans le décor  $R$  de nombres réels. Pour n'importe quel bord  $u$  de  $G$ , la valeur  $w(u)$  s'appelle le poids de  $u$  (pour  $w$ ), et la paire  $(G, w)$  s'appelle un *graphique pondéré par les bords*.

Étant donné un graphique pondéré par les bords  $(G, w)$ , laisser  $X$  être un sous-graphe de  $G$  et laissez  $\lambda$  être une valeur de  $R$ . le  *$\lambda$ -jeu d'arêtes de niveau* de  $X$  pour

$w$  est défini par  $w_{\lambda}(X) = \{u \in E(X) \mid w(u) < \lambda\}$ , et le

*$\lambda$ -graphique de niveau* de  $X$  pour  $w$  est défini comme le sous-graphe  $w_{\lambda}(X)$  de  $X$ , tel que  $w_{\lambda}(X) = (V(X), w_{\lambda}(X))$ . Ensuite, le connecté-partition des composants  $C(w_{\lambda}(X))$  induit par  $w_{\lambda}(X)$  est appelé les  *$\lambda$ -partition de niveau* de  $X$  pour  $w$ .

Comme nous ne considérons que des graphes et des hiérarchies finis, l'ensemble des valeurs de niveau considérées est réduit à un sous-ensemble fini de  $R$  qui est désigné par  $E$  dans les autres parties de cet article. Afin de parcourir les valeurs de cet ensemble et d'arrondir les valeurs réelles aux valeurs de  $E$  nous définissons, pour tout  $\lambda \in R$ :

$$p_{\in}(\lambda) = \max \{ \mu \in E \cup \{-\infty\} \mid \mu < \lambda \};$$

$$n_{\in}(\lambda) = \min \{ \mu \in E \cup \{\infty\} \mid \mu > \lambda \}; \text{ et}$$

$$\hat{n}_{\in}(\lambda) = \min \{ \mu \in E \cup \{\infty\} \mid \mu \geq \lambda \}.$$

Laisser  $(G, w)$  être un graphique pondéré par les bords et soit  $X$  être un sous-graphe de  $G$ . La séquence de tous  $\lambda$ -partitions de niveau de  $X$

pour  $w$  ordonné en augmentant la valeur de  $\lambda$ , à savoir  $(C(w_{\lambda}(X)) \mid \lambda \in E \cup \{\infty\})$ , est une hiérarchie, appelée le *hiérarchie de zone quasi plate* de  $X$  pour  $w$ , désigné par  $QFZ(X, w)$ . Laisser  $H$  être la hiérarchie de zone quasi plate de  $G$  pour  $w$ . Étant donné un sommet  $X$  de

$G$  et une valeur  $\lambda$  dans  $E$ , la région qui contient  $X$  dans le  $\lambda$ -partition de niveau du graphe  $G$  est désigné par  $H_{\lambda}^X$ .

Dans les autres parties de cet article, le symbole  $g$  désigne un graphe connexe, le symbole  $w$  désigne une carte de  $E$  dans  $R$ , et le symbole  $T$  désigne un arbre couvrant minimum de  $(G, w)$ .

Il a été montré dans [15] que la hiérarchie des zones quasi-plates  $QFZ(T, w)$  de  $T$  pour  $w$  est identique à la hiérarchie des zones quasi-plates  $QFZ(G, w)$  de  $G$  pour  $w$ . Cela indique que la hiérarchie de zone quasi plate pour  $G$  peut être géré par son arbre couvrant minimum.

### Méthode de segmentation hiérarchique basée sur des graphiques

Dans cet article, nous considérons que l'entrée est le graphe pondéré par les bords  $(G, w)$  représentant une image, où les pixels correspondent aux sommets de  $G$  et les bords lient les pixels adjacents. Le poids de chaque bord est donné par une mesure de dissimilarité entre les pixels liés telle que la différence absolue d'intensité entre eux.

Avant d'expliquer la méthode HGB, nous décrivons d'abord la dissimilarité d'échelle d'observation suivante [9], qui est requise par la méthode et dont l'idée provient du critère de fusion de région proposé dans [11].

*Différence d'échelle d'observation*: Dans le cas de l'algorithme de segmentation d'image Felzenszwalb-Huttenlocher [11], deux régions d'une image sont fusionnées sur la base d'un prédicat de fusion de région. Ce prédicat a ensuite été reformulé en tant que mesure de dissimilarité à l'échelle d'observation pour produire la segmentation hiérarchique d'une image [9] comme suit.

Laisser  $R_1$  et  $R_2$  être deux régions adjacentes, la mesure de dissimilarité compare le soi-disant inter-composant et différences intra-composant [11]. le *différence inter-composants* entre  $R_1$  et  $R_2$  est défini par  $\Delta_{Inter}(R_1, R_2) = \min \{ w(\{x, y\}) \mid x \in R_1, y \in R_2, \{x, y\} \in E(T) \}$ , tandis que les *différence intra-composant* d'une région  $R$  est défini par  $\Delta_{Intra}(R) = \max \{ w(\{x, y\}) \mid x, y \in R, \{x, y\} \in E(T) \}$ . Il

## Méthode 1: Méthode HGB

Contribution : Un arbre couvrant minimum  $T$  d'un

graphique pondéré par les bords ( $G, w$ )

Production: Une hiérarchie  $H = QFZ(T, f)$

```

1 pour chaque  $u \in E(T)$  faire  $f(u) := \max \{ \lambda \in E \}$ ;
2 pour chaque  $u \in E(T)$  par ordre non décroissant pour  $w$  faire
3    $H := QFZ(T, f); f(u) := p \in$ 
4    $(\lambda ?$ 
5    $H(u));$ 
6 fin
7  $H := QFZ(T, f);$ 

```

conduit à l'échelle d'observation de  $R_1$  relatif à  $R_2$ , défini par

$S_{R_1}(R_2) = \Delta_{R_1}(R_2) / \Delta_{R_1}(R_1)$ , où  $\Delta_{R_1}$  est la mesure de dissimilarité de  $R_1$  et  $R_2$ , appelée la dissimilarité de l'échelle d'observation, est défini par

$$D(R_1, R_2) = \max \{ S_{R_2}(R_1), S_{R_1}(R_2) \}. \quad (1)$$

Cette dissimilitude est utilisée pour déterminer si deux régions doivent être fusionnées ou non à une certaine échelle d'observation dans ce qui suit. L'entrée est

2) *Méthode*: La méthode HGB est présentée dans la méthode 1. fonction de une image représentée par un graphique  $g$  avec son arbre  $T$  de  $g$  est pris en effet. poids associée  $w$ , où la portée minimale calcule une nouvelle fonction de poids  $F$  de  $(T, w)$ . Hiérarchie des méthodes HGB  $H = QFZ(T, f)$ . La hiérarchie résultante  $H$  qui conduit à une nouvelle prise en compte des segmentations hiérarchiques de est l'image initiale. Ainsi, le cœur de la méthode est la génération l'image de la fonction de poids  $F$  pour  $T$ .

Pour calculer la nouvelle carte  $F$ , la méthode HGB initialise d'abord la valeur toutes les valeurs de  $F$  à l'infini (voir Ligne 1). Ensuite, une observation d'ordre d'échelle  $f(u)$  est calculé pour chaque arête  $u \in E(T)$  in (voir Ligne 2). Notez que non décroissant par rapport au poids d'origine  $w$  chaque itération de la boucle nécessite calculer la hiérarchie  $H = QFZ(T, f)$  (voir ligne 3). Une fois que  $H$  est obtenue, la valeur  $\lambda ?$   $H(u)$  d'un sous-ensemble fini  $E$  de  $R$  est obtenu par la minimisation:

$$\lambda ?_{H(\{x, y\})} = \min \{ \lambda \in E \mid \text{ré} (H_\lambda(x, y)) \} \quad (2)$$

Nous considérons d'abord les régions  $H_\lambda(x)$  et  $H_\lambda(y)$  à un niveau  $\lambda$ . En utilisant la mesure de dissimilarité  $\text{ré}$  nous vérifions si  $\text{ré} H_\lambda(x, H_\lambda(y)) \leq \lambda$ .

L'équation (2) indique que  $\lambda ?_{H(\{x, y\})}$  est la valeur minimale  $\lambda$  qui remplit cette minimisation. Observez que la minimisation impliqué dans l'équation (2) n'a de solution que si le maximum de  $E$  est supérieure à la valeur de dissimilarité maximale possible. Dans ce qui suit, nous supposons que cette hypothèse est toujours vraie. La figure 2 illustre un exemple d'application de la méthode 1.

Comme mentionné ci-dessus, Guimarães *et coll.* n'a pas fourni d'algorithme pratiquement efficace pour calculer la méthode 1. Afin de combler cette lacune, le problème est double. En effet, il est nécessaire de proposer des algorithmes efficaces (ie, exacts et rapides) pour (i) résoudre la minimisation impliquée dans l'équation (2); et (ii) calculer la hiérarchie des zones quasi-plates  $QFZ(T, f)$  à chaque itération de la méthode 1 (lignes 3 et 6).

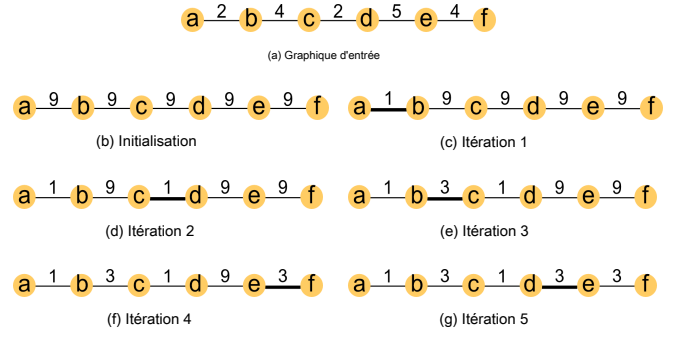


Fig.2: Illustration de la méthode 1 avec  $E = \{0, 1, \dots, 9\}$ : (a) le graphe d'entrée ( $T, w$ ), (b) le graphe ( $T, f$ ) à chaque itération de la méthode 1 et (g) la hiérarchie de zone quasi-plate résultante correspondant au graphe ( $T, f$ ).

## III. UNE LGORITHMES POUR HGB MÉTHODE

Dans cette section, nous étudions des algorithmes pour calculer les résultats de la méthode HGB. Dans les sections III-A, III-B et III-C, trois algorithmes pour effectuer la minimisation impliquée actuellement des La ligne 4 de la méthode 1 est présentée. Dans la section III-C, nous avons la hiérarchie de algorithmes non incrémentaux et incrémentaux pour obtenir zone quasi plate d'une carte de poids comme demandé à Lignes 3 et 6 de la méthode 1.

### A. Algorithme de minimisation naïf

Nous présentons d'abord un algorithme naïf, à savoir l'algorithme 1, pour calculer la valeur  $\lambda ?_{H(\{x, y\})}$  étant donné une hiérarchie  $H$  et un bord  $\{x, y\}$ . Selon l'équation (2), il s'agit simplement d'une valeur  $\lambda \in E$  tel que  $\text{ré}$  considérant les valeurs de  $E$  ( dans l'ordre croissant jusqu'à ce que  $E$  est un ensemble

d'entiers consécutifs, pour tout  $\lambda$  dans  $E$ , les résultat de  $n \in (\lambda)$  et  $p \in (\lambda)$  peut être obtenu avec le simple instruction entière  $\lambda + 1$  et  $\lambda - 1$ , respectivement.

### Algorithme 1: Minimisation naïve HGB de l'équation 2

Contribution : Une hiérarchie  $H$ , un bord  $u = \{x, y\}$

Production: ( La valeur  $\lambda ?$  tel que  $\lambda ? = \lambda ?_{H(\{x, y\})}$  )

```

1  $\lambda ? := \min \{ \lambda \in E \}$ ;
2 tandis que  $\text{ré} H_\lambda(x, H_\lambda(y)) > \lambda$  faire
3    $\lambda ? := n \in (\lambda ?)$ ;
4 fin

```

### B. Minimisation par ra (nge)

Dans l'algorithme 1,  $\text{ré} H_\lambda(x, H_\lambda(y))$  est calculé pour chaque valeur  $\lambda$  dans  $E$ . Cependant, il se peut bien qu'à deux reprises successives valeurs de  $E$ , les régions  $H_\lambda(x)$  et  $H_\lambda(y)$  rester le même et dans ce cas, il en va de même  $\text{ré} H_\lambda(x, H_\lambda(y))$ . Dans cette section, nous présentons un deuxième algorithme pour calculer  $\lambda ?_{H(\{x, y\})}$  ce qui nous permet pour réduire la quantité de calcul redondant par rapport à l'algorithme 1.

Afin d'obtenir cette réduction du calcul redondant, nous recherchons une partition de  $E$  ( qui, selon l'équation (2), est la plage des valeurs possibles de  $\lambda ?$

$$H(\{x, y\})$$

en un nombre fini d'intervalles discrets tels que, dans chaque

intervalle ( $I = K_{je_{\min}, je_{\max}} K \subseteq E$ , pour deux valeurs quelconques  $\lambda_1$  et  $\lambda_2$

dans  $je$  les régions) contiennent ( $g X$  et  $y$  au niveau  $\lambda_1$  et à

niveau  $\lambda_2$  restent inchangés. Ainsi, dans) ce cas, nous

avoir  $ré H_{\lambda_{X,Y}} H_{\lambda_1} = ré H_{\lambda_{X,Y}} H_{\lambda_2} = ré_{je}$  ce qui signifie

bref ça, J dans un tel  $je K$  nterval  $JE$ , la dissemblance devrait être

calculé une seule fois. Ensuite, on peut observer que, pour tout

valeur  $\lambda$  dans  $je \cap ré_{JE}, je_{\max}$ , la dissemblance entre  $K$  een les régions la valeur  $\lambda$  et cela, pour

contenant  $X$  et  $y$  au niveau  $\lambda$  de la hiérarchie  $H$  est inférieure à

n'importe quelle valeur  $\lambda$  dans  $je \cap je_{\min}, ré_{je} J$ , les

dissemblance entre les régions contenant  $X$  et  $y$  au niveau  $\lambda$

de la hiérarchie  $H$  est au dessus  $\lambda$ . Par conséquent, la solution  $\lambda^?$

à notre problème d'optimisation peut être obtenu en parcourant tous les intervalles

de la partition considérée de  $E$ .

Afin d'obtenir une telle partition de  $E$ , nous remarquons que si

il n'y a pas d'avantage  $E(T)$  dont le poids est compris entre  $\lambda_1$  et  $\lambda_2$ ,

puis le  $\lambda_1$ - jeu d'arêtes de niveau  $T$  pour  $F$  est égal au  $\lambda_2$ -

jeu d'arêtes de niveau  $T$  pour  $F$ , ce qui implique à son tour que le  $\lambda_1$

niveau de  $H = QFZ(T, f)$  est égal au  $\lambda_2$  niveau de  $H$ .

Par conséquent, pour obtenir un partiti  $K$  sur de  $E$ , tel que celui décrit dans le

paragraphe précédent, il est suf fi K cient de s'assurer que pour

chaque intervalle considéré  $je_{\min}, je_{\max}$ , il n'y a pas d'arête dont

le poids est strictement compris entre  $je_{\min}$  et  $je_{\max}$ . Afin de trouver la partition souhaitée  $E$ ,

il est pertinent de considérer les successives

valeurs de la plage  $R_{f=f(u) | u \in E(T)}$  de la fonction poids  $F$ . Plus

précisément, à partir de la discussion ci-dessus, nous

déduire la propriété suivante.

*Propriété 1:* Laisser  $H$  être une hiérarchie, laissez  $F$  être une carte de  $E(T)$

à  $E$  tel que  $H = QFZ((T, f))$  a (nd let  $\{x\}, y\}$  be (any edg) e)

de  $T$ . Ensuite nous avons

$$\lambda_H^?(x, y) = \min_{\lambda} \max_{\lambda \in R_F, RE(H_X, H_Y, \lambda) \leq \lambda} \hat{n} \in ré H_{\lambda} \quad (3)$$

Grâce à la propriété 1, nous calculons  $\lambda_H^?(x, y)$  par les sourcils

ing le va (lues de  $R$  par ordre croissant jusqu'à une valeur  $\lambda$

tel que  $RE(H_X, H_Y, \lambda) \leq \lambda$

la valeur  $\lambda^?$   $H((x, y))$  qui est simplement (le maximum de) en prenant (branche) gamme de  $H$  pour  $X$  et  $y$  désigné par  $R_X$  et  $R_Y$  qui est défini comme le

entre  $\hat{n} \in RE(H_X, H_Y, \lambda)$  et  $n \in p_B(\lambda)$ . Dans le but de faire

tel processus calculable, il est nécessaire de parcourir la gamme de  $F$  par ordre

croissant. Pour cela, nous proposons de stocker les valeurs de  $F$  dans une liste

chaînée triée. L'algorithme 2 fournit une description précise de ce processus. On

peut observer que

quand la valeur  $p \in (\lambda^? H((x, y)))$  n'est pas encore présent dans le

gamme de  $F$ , la liste chaînée représentant cette plage est mise à jour

afin qu'il soit prêt pour la prochaine itération de la boucle principale de la méthode 1. Il

faut également noter que dans la méthode 1, le poids de chaque arête est initialisé à

la valeur maximale de  $E$ . En d'autres termes, la liste chaînée doit être initialisée dans

la méthode 1 avec le singleton  $\{\max \{ \lambda \in E \} \}$ .

### C. Minimisation par branche

Dans la section précédente, nous réduisons la taille des branches de recherche de  $X$  et de  $y$  jusqu'à av (aleur  $\lambda$ , tel que  $RE(H_X, H_Y, \lambda) \leq \lambda$

espace de la minimisation dé fi nie dans l'équation (2) par con-  $\lambda$ , est trouvé. Enfin, la valeur  $\lambda^?$

sider la gamme  $R_F$  de la fonction  $F$  (c'est à dire, une caractéristique comme maximum entre  $\hat{n} \in RE(H_X, H_Y, \lambda)$  et  $n \in p_B(\lambda)$ ),

fonction de la hiérarchie considérée  $H$ ) au lieu de l'ensemble  $E$  où  $B = R_X$

### Algorithme 2: Minimisation HGB par plage

Contribution : Une hiérarchie  $H$ , une carte de poids  $F$  tel que

$H = QFZ(T, f)$ , un bord  $\{x, y\}$  de  $T$ , une liste liée  $L$  des valeurs

de  $R$  par ordre croissant  $F$

Production: La valeur  $\lambda^?$  tel que  $\lambda^? = \lambda_H^?(x, y)$ , les

liste liée mise à jour  $L$  des valeurs de

$R_F \cup \{p(\lambda^?)\}$  par ordre croissant

1  $l := L.he(ad; \lambda := ) l.valeur; \lambda_{prev} := -\infty;$

2 tandis que  $ré H_X, H_Y, \lambda$  faîte(

3  $\lambda_{prev} := \lambda; l := l.next; \lambda := l.valeur;$

4 fin

5  $\lambda^? := \max (n \in (\lambda_{prev}), \hat{n} \in RE(H_X, H_Y, \lambda))$

6 si  $p \in (\lambda^?) = \lambda_{prev}$  puis  $L.insert(p \in (\lambda^?))$ ;

de toutes les échelles possibles de la hiérarchie  $H$  (voir Propriété 1). Dans cette

section, nous montrons que cet espace de recherche peut être encore réduit,

conduisant à un troisième algorithme de calcul du

valeur  $\lambda^? H((x, y))$ , étant donné toute hiérarchie  $H$  et tout bord  $\{x, y\}$ .

Pour obtenir cette seconde réduction, on observe en

Équation (2) selon laquelle les seules régions de la hiérarchie impliquées dans la

minimisation sont celles contenant  $X$  et  $y$ . Donc,

en recherchant la valeur  $\lambda_H^?(x, y)$ , il n'est pas nécessaire de

considérer une échelle de  $H$  (c'est à dire, une valeur en  $R_F$ ) où les régions contenant  $X$  et  $y$  sont

les mêmes que ceux de l'échelle précédente.

En d'autres termes, plutôt que de considérer les échelles  $R_F$  pour lequel il y a un

changement global de la hiérarchie, on peut se concentrer

sur les échelles pour lesquelles le changement de hiérarchie est local

à  $X$  et  $y$ , c'est à dire, lorsque le changement concerne une région contenant soit  $X$  ou  $y$ .

Laisser  $X$  être n'importe quel sommet de  $V$  et désignons par  $B_H(X)$  l'ensemble qui

contient chaque région  $R$  de la hiérarchie  $H$  tel

cette  $X$  appartient à  $R$ . L'ensemble  $B_H(X)$  s'appelle la *branche de  $X$*

dans  $H$  le niveau d'une région  $R$ , désigné par  $niveau_H(R)$ , dans  $H$

est l'indice le plus bas d'une partition qui contient  $R$  dans  $H$ . le

$R_X$  et  $R_Y$  sont définies par  $R_X = \{R \in B_H(X) | X \subseteq R\}$  et  $R_Y = \{R \in B_H(Y) | Y \subseteq R\}$ .

On peut alors définir la *branche de  $X$  et  $y$*  désigné par  $R_{X,Y}$  qui est défini comme le

dans  $H: R_X \cap R_Y = \{R \in B_H(X) | R \in B_H(Y)\}$ . Utiliser cette notion

d'une plage de branches, la propriété suivante peut être déduite. La différence

avec la propriété 1 est que la plage de  $f$  (étant tel que  $H = QFZ(T, f)$ ) est remplacé

par l'union des gammes de branches de  $H$  pour  $X$  et pour  $y$ .

*Propriété 2:* Laisser  $H$  être une hiérarchie et laissez  $\{x, y\}$  être n'importe quel bord de  $T$ . Ensuite nous avons:

$$\lambda_H^?(x, y) = \min_{\lambda} \max_{\lambda \in B, RE(H_X, H_Y, \lambda) \leq \lambda} \hat{n} \in ré H_{\lambda} \quad (4)$$

où  $B = R_X \cup R_Y \cup F$ .

En raison de la propriété 2, pour calculer  $\lambda_H^?(x, y)$ , c'est suf fi sant

pour parcourir par ordre croissant les niveaux des régions du

$R_X$  et  $R_Y$  et  $F$ . Afin de proposer un tel algorithme,

il faut parcourir par ordre croissant les niveaux des régions dans les branches de  $X$  et de  $y$ . Cela peut être fait avec une structure de données arborescente, appelée arborescence des composants, qui représente la hiérarchie. L'arbre des composants est utilisé pour diverses tâches de traitement d'image et est bien étudié dans le domaine de la morphologie mathématique (voir, *par exemple*, [16] pour sa définition sur les graphes pondérés par sommets, [17] pour le cas des graphes pondérés par les arêtes et des zones quasi-plates, et [18] pour leur généralisation aux graphes orientés). En classification, cet arbre est souvent appelé le dendrogramme de la hiérarchie.

Comme tout arbre, l'arbre des composants de  $H$  peut être défini comme un couple constitué d'un ensemble de nœuds et d'une relation binaire (parent) sur l'ensemble des nœuds. Plus précisément, le *arborescence des composants de  $H$*  est la paire  $T_H = (N, \text{parent})$  tel que  $N$  est l'ensemble de toutes les régions de  $H$  et telle qu'une région  $R_1$  dans  $N$  est un *parent* d'une région  $R_2$  dans  $N$  n'importe quand  $R_1$  est un minimum (pour l'inclusion relation) sur-ensemble approprié de  $R_2$ . Notez que chaque région de  $N$  a exactement un parent sauf la région  $V$  qui n'a pas de parent et s'appelle la *racine* de l'arborescence des composants de  $H$ . Toute région qui n'est pas la mère d'une autre est appelée *feuille* de l'arbre. On peut observer que tout singleton de  $V$  est une feuille de  $T_H$  et que inversement toute feuille de  $T_H$  est un singleton de  $V$ .

Afin de parcourir la branche de  $X$  dans  $H$  à partir de son arborescence de composants, il suffit de suivre les étapes suivantes: (1) commencer par le nœud  $C$  c'est la feuille  $\{X\}$ , (2) considérez le parent de  $C$ , et (3) répétez l'étape (2) jusqu'à ce que la racine soit trouvée. De plus, il peut être observé que le *niveau*  $H$  l'attribut augmente dans la branche de  $X$ : pour tout nœud non racine  $C$  dans  $N$ , le niveau du parent de  $C$  n'est jamais inférieur au niveau de  $C$ . Par conséquent, le processus de navigation dans les branches permet également de parcourir la plage de branches de  $H$  pour  $X$  par ordre croissant. Selon la propriété 2, dans l'ordre pour trouver la valeur  $\lambda?$   $H(\{X, y\})$ , pour n'importe quel bord  $\{x, y\}$  de  $T$  et toute hiérarchie  $H$ , il faut considérer l'union des plages de  $H$  pour  $X$  et pour  $y$ , triés par ordre croissant. Cela peut être fait en naviguant simultanément dans l'arborescence des composants  $T_H$  les branches de  $X$  et de  $y$ . L'algorithme 3 fournit une description d'un algorithme complet pour trouver  $\lambda?$   $H(\{X, y\})$  en utilisant une telle navigation de branche simultanée.

#### Algorithme 3: Minimisation HGB par branche

Contribution : L'arborescence des composants  $T = (N, \text{parent})$  d'un

hiérarchie  $H$ , un bord  $u = \{x, y\}$  de  $T$ , un tableau

*niveau* qui stocke le niveau de chaque région de  $H$

Production: La valeur  $\lambda?$  tel que  $\lambda? = \lambda?$

$H(\{X, y\})$

1  $C_x := \{X\}; C_y := \{y\}; \lambda := -\infty; \lambda_{prev} := -\infty;$

2 tandis que  $D(C_x, C_y) > \lambda$  faire

3  $\lambda_{prev} := \lambda;$

4  $\lambda := \min(\text{niveau}[\text{parent}[C_x]], \text{niveau}[\text{parent}[C_y]]);$

5 si  $\text{niveau}[\text{parent}[C_x]] < \lambda$  puis  $C_x := \text{parent}[C_x];$

6 si  $\text{niveau}[\text{parent}[C_y]] < \lambda$  puis  $C_y := \text{parent}[C_y];$

7 fin

8  $\lambda? := \max(\lambda_{prev}, \lambda) \in RE(H_{X, H_{\lambda, y}});$

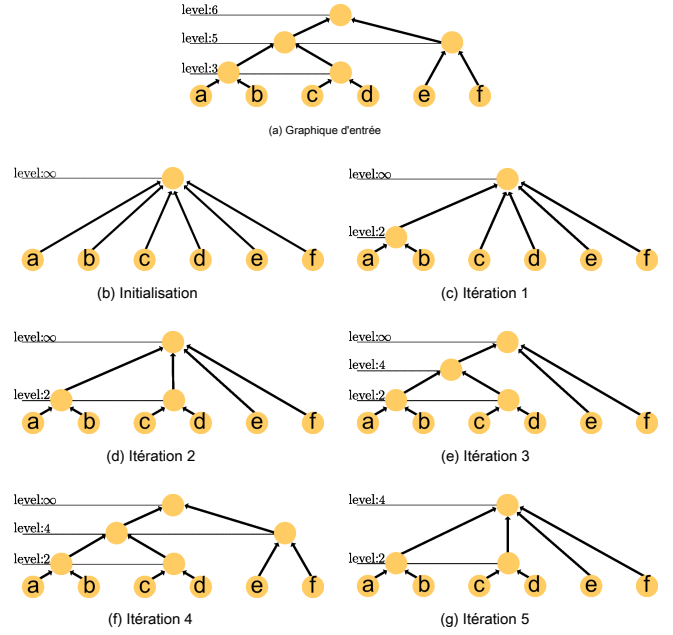


Fig. 3: Représentations arborescentes des hiérarchies de zones quasi-plates des graphiques de la Fig. 2; (g) montre la hiérarchie de sortie calculé par la méthode HGB.

#### D. Calcul QFZ

Dans cette section, nous nous concentrons sur les lignes 3 et 6 de la méthode 1, c'est-à-dire sur le calcul de la hiérarchie de zone quasi plate d'une carte de poids  $F$ . Ce calcul est répété à chaque itération de la méthode (c'est à dire, pour chaque bord de l'arbre  $T$ ). Par conséquent, trouver un moyen efficace d'accomplir cette tâche dans le contexte de la méthode 1 présente un potentiel d'accélération élevé.

Une première implémentation pour cette tâche consiste simplement à calculer, à chaque itération, la hiérarchie de zone quasi plate de  $F$  en utilisant un algorithme efficace tel que celui présenté dans [12]. Cependant, d'une itération de la boucle principale de la méthode 1 à la suivante, un seul poids du graphique est mis à jour et par conséquent la plupart des parties de l'arbre des composants restent inchangées (voir, par exemple, la figure 3). Par conséquent, plutôt que de recalculer à partir de zéro l'ensemble de l'arbre des composants à chaque itération, on peut deviner qu'une accélération importante peut être atteinte en ne mettant à jour que la partie de l'arbre des composants qui est affectée par la mise à jour de poids unique considérée à la présente itération. Un tel calcul est appelé mise à jour incrémentielle de zone quasi-plate. Dans [13] et [14], les auteurs proposent un algorithme pour fusionner les arbres composants de deux blocs d'image disjoints (adjacents) afin d'obtenir l'arbre composant de l'image constitué de ces deux blocs.  $\max \{ \lambda \in E \}$

à sa valeur finale, les algorithmes décrits dans [13] et [14] peuvent être adaptés au problème de cet article. L'algorithme de mise à jour modifie l'arborescence de la manière suivante: d'abord, étant donné une arête  $u = \{x, y\}$  du poids mis à jour  $\lambda = f(u)$ , les composants  $H_{\lambda, x}$  et  $H_{\lambda, y}$  sont identifiés dans l'arborescence puis un

TABLEAU I: Temps d'exécution à partir de l'image de la Fig.4 (a) ( 321 × 481 pixels). La hiérarchie résultante contient 5218 niveaux.

QFZ Algorithme	Minimisation Algorithme	Temps d'exécution (secondes)		
		Total	QFZ	Minimisation
Non-Incrémentale	Algorithme 1	14666,08	13186,31	1479,56
	Algorithme 2	13392,51	13375,29	17,02
	Algorithme 3	13166,25	13165,54	0,49
Incrémentale	Algorithme 1	1487,96	0,13	1487,75
	Algorithme 2	15,42	0,13	15,21
	Algorithme 3	0,49	0,10	0,32

un nouveau nœud est créé dans l'arborescence au niveau  $\lambda + 1$ , qui représente l'union de ces deux composants. Enfin, l'algorithme identifie les ancêtres de ces composants dans l'arborescence et met à jour la relation de parenté de ces nœuds. Ceci est fait jusqu'à ce que la racine soit trouvée. Par conséquent, seuls les composants contenant  $X$  et  $y$  sont impliqués dans l'algorithme de mise à jour et nous n'avons pas besoin de recalculer une hiérarchie entière à chaque itération.

#### IV. UNE ÉVALUATIONS

L'expérience vise à mesurer et comparer les temps d'exécution de toutes les variations de nos algorithmes, qui sont précédemment présentées pour la méthode HGB; comme nous avons trois variantes pour le pas de minimisation (ligne 4 dans la méthode 1), les algorithmes 1, 2 et 3, et deux variantes pour le calcul de zone quasi-plate (lignes 3 et 6 dans la méthode 1), la non-incrémentale [12] et l'incrémental [13, 14], le nombre total de toutes les combinaisons est de six. Les algorithmes ont été implémentés en C et exécutés sur un ordinateur avec un processeur 3,2 GHz, 8 Go de RAM. Les six algorithmes ont été exécutés sur l'image de la figure 4 (a). La figure 4 (b) montre la segmentation hiérarchique résultante, à partir de laquelle nous voyons qu'un grand nombre de régions et de niveaux hiérarchiques ont été produits.

Le tableau I montre les résultats pour toutes les variations. Nous observons que l'utilisation du calcul incrémental de zone quasi-plate offre un grand gain d'efficacité par rapport à l'approche non incrémentale. Pour l'étape de minimisation, l'algorithme 1 est le moins efficace de tous. Il est important de noter que l'algorithme 3 est beaucoup plus rapide que l'algorithme 2, qui valide que la minimisation par branche est le plus efficace des trois algorithmes pour résoudre le problème de minimisation. Pour une évaluation plus approfondie, nous avons testé notre algorithme le plus rapide sur les 500 images de l'ensemble de données Berkeley, ce qui a conduit à un temps d'exécution moyen de  $0,19 \pm 0,02$  seconde.

#### V. CONCLUSIONS

Nous avons d'abord étudié la méthode HGB [10] dans le but de proposer des algorithmes pratiques pour sa mise en œuvre sur des images. Nous nous sommes concentrés sur deux étapes pour améliorer l'efficacité: (i) la minimisation impliquée dans l'équation (2), et (ii) le calcul des hiérarchies de zones quasi-plates. Concernant (i), nous avons présenté deux propriétés qui nous permettent d'améliorer un

algorithme pour calculer la valeur minimale  $\lambda$ ?  $H(X, y)$  pas à pas étape. Le plus efficace est l'algorithme 3, comme le confirme les évaluations (voir tableau I). Afin de calculer efficacement



Fig. 4: Image utilisée pour l'évaluation de l'algorithme et la hiérarchie HGB résultante représentée sous forme de carte de saillance.

la hiérarchie de zone quasi-plate (ii), la stratégie de mise à jour incrémentale [13] a été utilisée. L'amélioration par rapport à la stratégie non incrémentale est également confirmée par l'expérience.

#### RÉFÉRENCES

- [1] T. Pavlidis, *Reconnaissance de modèle structurel*. Springer, 1977.
- [2] P. Arbelaez, M. Maire, C. Fowlkes et J. Malik, «Contour detection and hierarchical image segmentation», *TPAMI*, pp. 898–916, 2011.
- [3] J.-H. Syu, S.-J. Wang et L. Wang, «Segmentation hiérarchique des images basée sur la contraction et la fusion itératives», *POINTE*, 2246 à 2260, 2017.
- [4] S. Beucher, «Bassin versant, segmentation hiérarchique et algorithme de cascade», dans *ISMM*, 1994, pp. 69–76.
- [5] L. Najman et M. Schmitt, «saillance géodésique des contours des bassins versants et segmentation hiérarchique», *TPAMI*, pp. 1163–1173, 1996.
- [6] P. Salembier et L. Garrido, «L'arbre de partition binaire comme représentation efficace pour le traitement d'image, la segmentation et la recherche d'informations», *POINTE*, 561–576, 2000.
- [7] L. Guigues, JP Cocquerez et H. Le Men, «Scale-sets image analysis», *IJCV*, pp. 289–317, 2006.
- [8] F. Meyer et P. Maragos, «Représentation morphologique d'espace-échelle avec nivellement», dans *Théories de l'Echelle-Espace en Vision par Ordinateur*, 1999, pp. 187–198.
- [9] S. Guimarães, Y. Kenmochi, J. Cousty, Z. Patrocínio Jr. et L. Najman, «Hierarchizing graph-based image segmentation algorithms based on region dissimilarity - the case of the Felzenszwalb-Huttenlocher method», *Math. Morphol. Théorie Appl.*, pp. 1 à 22, 2017.
- [dix] S. Guimarães, J. Cousty, Y. Kenmochi et L. Najman, «A hierarchical image segmentation algorithm based on an observation scale», dans *SSPR*, 2012, p. 116–125.
- [11] PF Felzenszwalb et DP Huttenlocher, «Efficient graph-based image segmentation», *IJCV*, pp. 167–181, 2004.
- [12] L. Najman, J. Cousty et B. Perret, «Playing with kruskal: algorithms for morphological tree in edge-weighted graphs», dans *ISMM*, 2013, pp. 135–146.
- [13] J. Havel, F. Merciol et S. Lefèvre, «Construction d'arbres efficace pour la représentation et le traitement d'images multi-échelles», *JRTIP*, pp. 1 à 18, 2016.
- [14] MH Wilkinson, H. Gao, WH Hesselink, J.-E. Jonker et A. Meijster, «Calcul simultané de filtres d'attributs sur des machines parallèles à mémoire partagée», *TPAMI*, p. 1800–1813, 2008.
- [15] J. Cousty, L. Najman, Y. Kenmochi et S. Guimarães, «Segmentation hiérarchique avec graphes: zones quasi-plates, arbres à portée minimale et cartes de saillance», *JMIV*, pp. 1 à 22, 2017.
- [16] P. Salembier, A. Oliveras et L. Garrido, «Opérateurs connectés anti-extensifs pour le traitement d'images et de séquences», *POINTE*, 555–570, 1998.
- [17] J. Cousty, L. Najman et B. Perret, «Liens constructifs entre certaines hiérarchies morphologiques sur des graphes pondérés par les bords», dans *ISMM*, 2013, pp. 135–146.
- [18] B. Perret, J. Cousty, O. Tankyevych, H. Talbot et N. Passat, «Opérateurs connectés dirigés: hiérarchies asymétriques pour le filtrage et la segmentation d'images», *TPAMI*, pp. 1162–1176, 2015.