



**HAL**  
open science

## Une méthode de réduction exacte pour la segmentation par graph cuts

Nicolas Lermé, François Malgouyres, Lucas Létocart, Jean-Marie Rocchisani

► **To cite this version:**

Nicolas Lermé, François Malgouyres, Lucas Létocart, Jean-Marie Rocchisani. Une méthode de réduction exacte pour la segmentation par graph cuts. 2011. hal-00558895v1

**HAL Id: hal-00558895**

**<https://hal.science/hal-00558895v1>**

Preprint submitted on 24 Jan 2011 (v1), last revised 11 Apr 2011 (v2)

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Une méthode de réduction exacte pour la segmentation par graph cuts

N. Lermé<sup>1,2</sup>

F. Malgouyres<sup>1</sup>

L. Létocart<sup>2</sup>

J-M. Rocchisani<sup>3,4</sup>

(1) LAGA UMR CNRS 7539, (2) LIPN UMR CNRS 7030, (3) SMBH  
Université Paris 13 –Avenue J.B. Clément  
93430 Villetaneuse - France

(4) Hôpital Avicenne, 93009 Bobigny - France

{nicolas.lerme, lucas.letocart}@lipn.univ-paris13.fr  
malgouy@math.univ-paris13.fr  
jean-marie.rocchisani@univ-paris13.fr

## Résumé

*En quelques années, les graph cuts sont devenus un standard au sein de la communauté de la vision par ordinateur. Néanmoins, leur grande consommation mémoire reste un problème majeur : les graphes sous-jacents contiennent des milliards de noeuds et encore davantage d'arcs. Excepté quelques méthodes [16, 10, 4] exactes, les heuristiques présentes dans la littérature ne permettent d'obtenir qu'une solution approchée [13, 14, 8, 17]. Dans un premier temps, nous présentons une nouvelle stratégie pour réduire exactement ces graphes : le graphe est construit en ajoutant les noeuds qui satisfont localement une condition donnée et correspond à une bande étroite autour des contours de l'objet à segmenter. Les expériences présentées pour segmenter des images en niveaux de gris et en couleur mettent en évidence une faible consommation mémoire et une grande précision sur les segmentations. Dans un second temps, nous présentons aussi une application de cette méthode pour segmenter des tumeurs et des nodules dans des images scanner.*

**Mots-clés :** graph cuts, réduction, segmentation, tumeurs.

## Abstract

*In few years, graph cuts are a standard in the computer vision community. However, their huge memory consumption remains a challenging problem since underlying graphs contains billions of nodes and even more edges. Except some exact methods [16, 10, 4], the heuristics in the literature can only obtain an approached solution [13, 14, 8, 17]. First, we present a new strategy for reducing exactly these graphs : the graph is built by adding nodes which satisfy locally a given condition and corresponds in a narrow band around the segmented object edges. The experiments presented for segmenting gray-levels and color images highlight low memory usage and high accuracy in segmentations. Secondly, we present an application of this method for segmenting lung tumors and nodules in CT images.*

**Keywords :** graph cuts, reduction, segmentation, tumors.

## 1 Introduction

L'approche par minimisation d'énergie et l'optimisation discrète sont devenues des domaines importants en vision par ordinateur. Cela est notamment dû à la capacité de pouvoir calculer efficacement le Maximum a Posteriori (MAP) de Champs de Markov aléatoires (CMA) issus de nombreux problèmes à l'aide de méthodes comme les graph cuts. Initialement introduits par Greig et *al.* pour la restauration d'images binaires [6], les graph cuts ont connu un nouvel essor avec l'arrivée d'un algorithme de flot maximum rapide [2].

Parallèlement, les avancées technologiques dans le domaine de l'acquisition des images ont augmenté de manière importante à la fois le volume et la diversité des données à traiter. Par exemple, les derniers appareils photo grand public sont dotés de capteurs à 20 millions de pixels. De manière similaire, les systèmes d'acquisition médicaux actuels sont capables d'acquérir des volumes de données 3D contenant plusieurs milliards de voxels.

Dans le contexte de la segmentation d'images, les méthodes d'optimisation globale telles que les graph cuts sont incapables de résoudre de telles instances à cause des besoins mémoire requis. Pour pallier ce problème, certains auteurs ont proposé des méthodes exactes [16, 10, 4] pour réduire ces graphes. Dans [4], les auteurs présentent un algorithme de flot maximum parallèle avec une accélération quasi linéaire en fonction du nombre de processeurs. Cependant, le schéma d'accélération utilisé est limité puisqu'une telle accélération n'est atteinte que pour un nombre élevé de processeurs. De plus, l'algorithme reste moins efficace sur des graphes de petite taille. Dans [10], les auteurs font évoluer une bande autour de l'objet qui s'agrandi lorsque la coupe minimum rentre en contact avec le bord de la bande. Ce processus est itéré jusqu'à ce que la bande ne soit plus déformée. L'algorithme converge rapidement mais aucune borne sur la taille de la bande est donnée. Bien que les bandes générées soient relativement minces pour le problème considéré, rien ne garanti qu'elles le soient aussi

dans le cas de la segmentation. Dans [16], l'approche utilisée est différente : le problème est cette fois-ci décomposé en sous-problèmes résolus indépendamment en parallèle ou en distribué. L'optimalité sur la solution est garantie par décomposition duale, ou plus précisément, les solutions des sous-problèmes sont contraintes à être égales sur une bande de chevauchement commune.

Par ailleurs, des heuristiques basées par exemple sur des schémas multi-résolution [14, 15, 8] ont été proposées. Le principe est de construire un graphe dans une bande étroite contrainte par la segmentation issue d'une image sous-échantillonnée. L'algorithme réduit considérablement le temps de calcul et la consommation mémoire mais ne parvient généralement pas à conserver les détails. Bien que réduit dans [15], cet inconvénient reste vrai pour des images peu contrastées. L'approche est raffinée dans [8] en introduisant une mesure de confiance sur chaque pixel. L'approche permet d'obtenir des bandes assez fines tout en étant très proche de la solution optimale. Cependant, aucune expérience n'est présentée pour de grandes images.

Une autre approche s'appuie sur les graphes d'adjacence [13, 3]. Le principe est de pré-segmenter l'image à partir d'un outil de segmentation bas niveau (par exemple, ligne de partage des eaux [13] ou mean shift [3]), de construire un graphe d'adjacence où chaque noeud correspond à une région et de calculer un flot maximum dans ce graphe. Là encore, le temps de calcul ainsi que la consommation mémoire sont grandement diminués. Néanmoins, les résultats dépendent de la sensibilité au bruit de l'outil de segmentation bas niveau. De plus, les résultats se dégradent rapidement à mesure que la taille des régions de la pré-segmentation augmente.

Dans [17], les graphes réduits correspondent à des bandes étroites obtenues en écartant les zones homogènes, au sens d'un critère de similarité défini sur les niveaux de gris. Les bandes obtenues sont étroites mais l'approche est grandement limitée par l'étroitesse de son champ applicatif.

Dans cet article, nous décrivons une nouvelle stratégie de réduction. Le graphe est progressivement construit en ajoutant les noeuds qui satisfont localement une condition donnée. À l'instar de [14, 15, 8], les noeuds sont situés dans une bande étroite autour des contours de l'objet à segmenter. Empiriquement, les solutions obtenues après réduction sont identiques à celles obtenues sans réduction. Grâce à un algorithme rapide, la complexité initiale de la condition est calculée en temps constant, hors bords de l'image. Les expériences montrent que le temps requis par notre algorithme est parfois compensé par le temps qui serait nécessaire à l'allocation du graphe ainsi qu'au calcul du flot maximum sur le graphe réduit.

Le reste de cet article est organisé comme suit. En section 2, nous rappelons le cadre général des graph cuts. Après avoir introduit quelques définitions, nous détaillons notre stratégie de réduction en section 3 et présentons des expériences pour segmenter des images multidimensionnelles en niveaux de gris et couleur. Nous concluons en

montrant une application de cette méthode pour la segmentation de tumeurs pulmonaires en section 4.

## 2 Principe des graph cuts

Soit  $I : \mathcal{P} \subset \mathbb{Z}^d \rightarrow \mathbb{R}^m$  ( $d > 0, m > 0$ ) une image où chaque point  $p \in \mathcal{P}$  correspond à un pixel de valeur  $I_p$ . Habituellement,  $\mathcal{P}$  correspond à un carré lorsque  $d = 2$ , un cube lorsque  $d = 3$  et à un cube doté d'un intervalle de temps lorsque  $d = 4$ . On définit une segmentation binaire comme une application  $u$  affectant à chaque pixel  $p \in \mathcal{P}$  la valeur 0 (fond) ou 1 (objet) et l'on écrit  $u \in \{0, 1\}^{\mathcal{P}}$ . Le problème de segmentation peut alors être résolu efficacement en minimisant un CMA de la forme [1] :

$$E(u) = \beta \cdot \sum_{p \in \mathcal{P}} E_p(u_p) + \sum_{(p,q) \in \mathcal{N}} E_{p,q}(u_p, u_q), \quad (1)$$

parmi  $u \in \{0, 1\}^{\mathcal{P}}$  et pour  $\beta \in \mathbb{R}^+$ . Le système de voisinage  $\mathcal{N}$  contient toutes les paires de points voisins  $(p, q) \in \mathcal{N}$ . Par la suite, nous utiliserons les voisinages suivants :

$$\begin{aligned} \mathcal{N}_0 &= \{(p, q) \mid p, q \in \mathcal{P}^2, \sum_{i=1}^d |q_i - p_i| = 1\} \quad \text{ou,} \\ \mathcal{N}_1 &= \{(p, q) \mid p, q \in \mathcal{P}^2, |q_i - p_i| \leq 1 \forall 1 \leq i \leq d\}. \end{aligned}$$

où  $p_i$  désigne la  $i^{\text{ème}}$  coordonnée du point  $p$  et  $|\cdot|$  désigne la valeur absolue. Par la suite, les termes « connexité 0 » et « connexité 1 » indiqueront respectivement l'utilisation d'un voisinage de type  $\mathcal{N}_0$  et  $\mathcal{N}_1$ . Dans (1), le terme de région  $E_p(\cdot)$  favorise l'appartenance de chaque pixel au fond ou à l'objet. Ce terme est calculé en fonction de  $I$  ainsi que des graines objet  $\mathcal{O} \subset \mathcal{P}$  et fond  $\mathcal{B} \subset \mathcal{P}$ . Le terme de frontière  $E_{p,q}(\cdot)$  pénalise les pixels voisins  $p$  et  $q$  ayant des étiquettes différentes et favorise dans la segmentation les contours liés à un fort gradient. Notons que cette détection peut être améliorée en incluant d'autres caractéristiques telles que des informations sur la texture ou la direction du gradient.

Selon [9], lorsque les termes  $E_{p,q}(\cdot)$  sont sous-modulaires, le minimiseur de (1) peut être obtenu en calculant une coupe minimum / flot maximum dans un graphe orienté et pondéré  $\mathcal{G} = (\mathcal{V}, \mathcal{E}, c)$  où  $\mathcal{V} = \mathcal{P} \cup \{s, t\}$ ,  $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$  désigne l'ensemble des arcs et où chaque arc est muni d'une capacité  $c : \mathcal{E} \rightarrow \mathbb{R}^+$ . Les noeuds  $s$  et  $t$  correspondent respectivement au terminal « objet » et « fond ». L'ensemble des arcs  $\mathcal{E}$  est divisé en deux ensembles disjoints  $\mathcal{E}_n$  et  $\mathcal{E}_t$  correspondant respectivement aux n-links (arcs reliant deux noeuds de  $\mathcal{P}$ ) et aux t-links (arcs reliant un noeud à  $s$  ou  $t$ ). Lorsque la coupe minimum est calculée, on fixe  $u_p = 1$  si un noeud  $p$  est connecté à  $s$  et  $u_p = 0$  si  $p$  est connecté à  $t$ .

## 3 Réduction

### 3.1 Définitions et notations

Comme nous l'avons évoqué précédemment, la consommation mémoire des graph cuts pour la segmentation

d'images haute-résolution peut avoir un coût prohibitif. Par exemple, l'algorithme de flot maximum de Boykov et Kolmogorov en version 2.2 alloue  $24|\mathcal{P}| + 14|\mathcal{E}_n|$  octets. Pour une quantité fixée de RAM fixée à 2 Go, on observe que la taille maximum d'une image décroît rapidement à mesure que la dimension  $d$  croît (voir Tableau 1). Néanmoins,

	Connexité 0	Connexité 1
2D	6426	4459
3D	319	219
4D	68	45

TAB. 1: Taille maximum d'une image carrée pour que le graphe associé puisse tenir en mémoire en fonction de  $d$  et de la connexité, pour une quantité de mémoire fixée.

comme observé dans [11], certains noeuds dans un graphe sont inutiles du point de vue du flot maximum. En effet, la plupart de ces noeuds ne sont traversés par aucun flot. Partant de ce constat, on souhaiterait donc pouvoir extraire le plus petit sous-graphe  $\mathcal{G}' = (\mathcal{V}', \mathcal{E}', c)$  à partir de  $\mathcal{G}$ , tout en gardant une solution  $u'$  identique (ou très proche) de  $u$ . En d'autres termes, on souhaite maximiser le taux de réduction  $\rho = 1 - \frac{v}{v'}$  sous la contrainte  $u \simeq u'$ . Cela correspond à un problème d'optimisation que nous n'essayerons pas de résoudre puisque la méthode pour déterminer  $\mathcal{G}'$  doit rester rapide.

Avant de détailler notre stratégie pour construire  $\mathcal{G}'$ , nous avons besoin d'introduire quelques définitions. En accord avec [9], on considère (sans perte de généralités) qu'un noeud n'est connecté qu'à au plus un terminal dans  $\mathcal{G}$  :

$$(s, p) \in \mathcal{E}_t \Rightarrow (p, t) \notin \mathcal{E}_t, \quad p \in \mathcal{P}.$$

On contracte la notation des capacités sur les t-links connectés à un noeud  $p \in \mathcal{P}$  :

$$c(p) = c(s, p) - c(p, t).$$

Considérons maintenant une fenêtre carrée  $B$  de taille  $(2r + 1)$  ( $r > 0$ ) centrée à l'origine. On désigne par  $\tilde{B}_p$  la translation de  $B$  à une position  $p \in \mathcal{P}$  :

$$\tilde{B}_p = \{b + p \mid b \in B\}.$$

Pour un ensemble  $Z \subset \mathcal{P}$ , on désigne aussi par  $\tilde{Z}_B$  la dilatation de  $Z$  par l'élément structurant  $B$  :

$$\tilde{Z}_B = \{p + b \mid p \in Z, b \in B\} = \bigcup_{p \in Z} \tilde{B}_p.$$

L'idée pour construire  $\mathcal{G}'$  est alors la suivante : supprimer les noeuds d'un ensemble  $Z \subset \mathcal{P}$  dont tous les noeuds sont reliés à  $s$  (respectivement à  $t$ ) tel que la quantité maximale de flot qui peut entrer (respectivement sortir) de  $\tilde{Z}_B \setminus Z$  par les t-links est plus grande que la quantité de flot qui peut sortir (respectivement entrer) de  $\tilde{Z}_B$  par les n-links (voir

Figure 1). La construction d'un tel ensemble  $Z$  est réalisée en testant chaque noeud  $p \in \mathcal{P}$ . Par conséquent, les noeuds se trouvent localisés autour des contours de l'objet à segmenter. En prenant pour hypothèse que les capacités sur les n-links sont inférieures ou égales à un (ce qui reste vrai pour la plupart des modèles d'énergie en segmentation), nous proposons d'utiliser la condition suivante (légèrement plus restrictive) pour tester chaque pixel  $p \in Z$  :

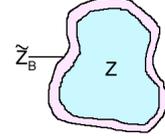


FIG. 1: Principe de la réduction : les noeuds issus de  $Z$  sont supprimés puisque chaque noeud  $p \in Z$  satisfait (2). Les noeuds restants sont typiquement localisés dans une bande étroite  $\tilde{Z}_B \setminus Z$ .

$$\begin{cases} \forall q \in \tilde{B}_p, c(q) \geq +\delta & \text{ou} \\ \forall q \in \tilde{B}_p, c(q) \leq -\delta, \end{cases} \quad (2)$$

où  $\delta = \frac{P(B)}{(2r+1)^2 - 1}$  et  $P(B)$  définissant le périmètre de  $B$  :

$$P(B) = \max(|\{(p, q) : p \in B, q \notin B \text{ et } (p, q) \in \mathcal{N}\}|, |\{(p, q) : p \in B, q \notin B \text{ et } (p, q) \in \mathcal{N}\}|).$$

Clairement, pour un noeud  $p \in \mathcal{P}$  satisfaisant (2),  $p$  est seulement relié à  $s$  (respectivement à  $t$ ) et le flot qui peut rentrer (respectivement sortir) à travers les t-links dans  $\tilde{B}_p \setminus \{p\}$  suffit à saturer les n-links sortant (respectivement entrant) de  $\tilde{B}_p$ . Le noeud  $p$  devient inutile et peut alors être supprimé de  $\mathcal{G}$ . La condition (2) permet d'exhiber un algorithme simple pour construire  $\mathcal{G}'$  en évaluant (2) de manière incrémentale pour atteindre une complexité  $O(|\mathcal{P}|)$  (hors bords de l'image), indépendante du rayon de la fenêtre.

Les expériences présentées dans [11] confirment la dépendance intuitive entre taux de réduction et les paramètres du modèle. Par exemple, puisque les capacités contractées sont multipliées par  $\beta$  qui est présent dans (1), il est facile de voir que la condition (2) est satisfaite plus difficilement lorsque  $\beta$  décroît (forte régularisation). Ainsi, l'algorithme a besoin d'un rayon de fenêtre plus grand pour faire décroître  $\delta$  et génère alors des bandes plus larges. Inversement, lorsque  $\beta$  augmente, le graphe réduit est constitué de bandes plus étroites. Enfin, nous avons à la fois des éléments théoriques et empiriques (voir section 3.2) suggérant que cette réduction est exacte.

### 3.2 Expériences

Dans cette section, nous comparons les performances des graph cuts et de notre algorithme en terme de temps d'exécution et d'utilisation mémoire pour segmenter des images à l'aide d'un modèle Boykov/Jolly [1] en connexité 1 (voir Tableau 2 et Figure 2). Les temps d'exécution sont mesurés sur une moyenne de dix lancements et incluent le temps

de construction du graphe, le calcul de la coupe minimum et la construction de la solution. La machine utilisée pour ces expériences est un Athlon Dual Core 6000+ cadencé à 3 GHz avec 2 Go de RAM. Pour le moment, les distributions de probabilités sont estimées à l'aide d'histogrammes normalisées à partir des ensembles  $\mathcal{O}$  et  $\mathcal{B}$ .

Avec un taux de réduction moyen de 79.42%, les résultats présentés dans le Tableau 2 sont encourageants. Dans le cas de l'image « cellules-souris », le taux de réduction est plus faible car les données sont très bruitées. Cela implique donc une valeur de  $\beta$  plus faible pour limiter l'attache aux données dans (1). Pour toutes les images, il est important de noter que la différence de flot maximum sur le graphe non réduit et le graphe réduit est toujours égale à zéro. Il en est de même pour la différence concernant le nombre de pixels appartenant à la partie objet dans la segmentation. Le Tableau 2 montre également que le rayon optimal qui maximise le taux de réduction est toujours égal à un pour des images contrastées comme c'est le cas ici.

Excepté pour une instance, les graph cuts standards sont incapables de segmenter les volumes 2D+t et 3D. Notre approche permet de les segmenter et ce, en moins d'une minute. Pour certaines images, notre méthode est même parfois plus rapide mais devient moins compétitive en couleur. En effet, bien que la condition (2) soit calculée en  $O(1)$ , le temps passé sur les bords croît progressivement avec  $r$ . Cet effet est accentué lorsque le nombre de canaux augmente.

## 4 GCSSLP : application de la réduction à la segmentation de tumeurs

Durant ces dix dernières années, mesurer précisément la taille des tumeurs pulmonaires est devenu un enjeu considérable pour estimer au mieux leur réponse aux traitements thérapeutiques chez les patients. Dans ce contexte, nous souhaitons segmenter des tumeurs pulmonaires de forme irrégulière, directement en contact avec les tissus sains. Le problème est rendu difficile par l'absence d'interface tumeur/tissus sains qui ne permet pas de s'appuyer sur les niveaux de gris de l'image. Segmenter de manière semi-interactive ces tumeurs se révèle être la seule solution. Récemment, Ye *et al.* ont proposé une méthode efficace basée sur les graph cuts pour segmenter des nodules [18]. Néanmoins, le problème considéré est plus simple que le nôtre et utilisent un a priori sur la forme.

Dans ce cadre, nous proposons une nouvelle méthode pour segmenter de manière semi-interactive des tumeurs pulmonaires par graph cuts. L'originalité de ce travail consiste à la fois en (a) la réduction des graphes pour segmenter de larges volumes de données (voir section 3) et (b) l'introduction d'un modèle d'énergie avec un a priori sur l'emplacement des graines objet : GCSSLP (Graph Cuts-Based Segmentation With Seeds Location Prior). Après avoir décrit ce modèle, nous présentons quelques expériences que nous comparons avec les vérités terrain fournies par un expert.

### 4.1 Fonction d'énergie

Pour obtenir des segmentations de qualité, le modèle de Boykov/Jolly requiert que l'objet et le fond soient de couleur différente. Ce comportement n'est pas suffisant dans notre cas car les tumeurs et les tissus sains apparaissent dans la même gamme d'intensités. Pour pallier ce problème, nous proposons d'incorporer un a priori sur la position des graines objet dans (1). Le terme de frontière reste identique à celui proposé dans [1] :

$$E_{p,q}(u_p, u_q) = B_{p,q} \cdot |u_p - u_q|,$$

avec

$$B_{p,q} = \frac{1}{d_e(p,q)} \cdot \exp\left(-\frac{\|I_p - I_q\|^2}{2\sigma^2}\right),$$

où  $d_e(\cdot)$  représente la distance euclidienne entre deux points  $p$  et  $q$ ,  $I$  est l'image originale et  $\sigma > 0$  un paramètre. Le terme de région est défini dans le Tableau 3. Les ensembles  $\mathcal{O}$  et  $\mathcal{B}$  correspondent aux graines saisies par l'utilisateur,  $\sigma_a$  est un paramètre et  $d(p, \mathcal{O})$  est une fonction de distance définie entre un point  $p \in \mathcal{P}$  et  $\mathcal{O}$  :

$$d(p, \mathcal{O}) = \min \{dist(p, q) \mid q \in \mathcal{O}\}.$$

Le paramètre  $\sigma_a$  permet de contrôler le degré de propagation des graines et définit une zone d'influence  $A_{\sigma_a}$ . Au delà de cette zone, les noeuds sont uniquement reliés à  $t$  avec une capacité très grande. Cela permet à la fois d'assurer qu'un pixel appartienne bien au fond et que le noeud associé n'appartienne pas à  $\mathcal{G}'$ . Dans nos expériences, nous fixons  $\sigma_a = 10$ . La différence majeure entre l'énergie proposée et celle décrite dans [1] réside dans le choix de la fonction  $dist(\cdot)$ . Nous avons choisi pour  $dist(\cdot)$  une distance géodésique [7] dans le graphe  $\mathcal{G}$  muni des valuations suivantes :

$$\sqrt{|I_p - I_q|^2 + d_e(p, q)} \quad \text{si } (p, q) \in \mathcal{N}.$$

Ce choix permet de prendre en compte les zones à fort gradient et d'éviter d'inclure des zones qui seraient trop éloignées des graines de l'objet. L'algorithme procède comme suit : la carte de distance est calculée à partir de  $\mathcal{O}$  puis seuillée de manière à déterminer la zone d'influence  $A_{\sigma_a}$ . Le graphe réduit est ensuite créé à partir du sous-volume englobant  $A_{\sigma_a}$ . La coupe minimum est ensuite calculée dans ce sous-volume pour obtenir la segmentation désirée.

### 4.2 Validation

Nous présentons maintenant quelques expériences pour segmenter des tumeurs issues d'images scanner. Les tumeurs et les structures avoisinantes peuvent présenter un contraste très différent d'une image à l'autre, impliquant des valeurs différentes pour le paramètre  $\sigma$ . À titre d'exemple, lorsque la valeur moyenne du gradient autour de la tumeur est faible, on a besoin d'une valeur de  $\sigma$  faible. Les arcs autour de la tumeur seront plus susceptibles d'appartenir à la coupe minimum puisqu'ils deviendront moins

	Nom de l'image	Taille	Graph cuts standards		Réduction		$\rho^*$ (%)	$r^*$
			Temps	Mémoire	Temps	Mémoire		
2D	étoile-mer-c	481 × 321	0.22	27.59	0.27	8.31	76.76	1
	livre	3012 × 2048	9.19	1105.92	8.94	94.68	91.45	1
	symbole-viking	660 × 740	0.71	87.41	0.70	4.01	94.62	1
2D+t	interview-c	320 × 240 × 203	<b>PM</b>	8079.36	38.27	950.25	88.74	1
	avion-c	492 × 276 × 180	<b>PM</b>	12677.12	55.95	1198.08	90.28	1
	discours-c	370 × 276 × 190	<b>PM</b>	10065.92	49.85	1423.36	85.23	1
	cellule-fluo-c	478 × 396 × 121	<b>PM</b>	11868.16	61.68	1546.24	85.94	1
3D	ct-thorax	245 × 245 × 151	<b>PM</b>	4689.92	22.95	950.25	81.54	1
	cellules-souris	230 × 230 × 57	10.90	1546.24	16.04	1423.36	20.88	1
	cerveau+bruit-3%	181 × 217 × 181	<b>PM</b>	3676.16	19.30	950.25	78.81	1

TAB. 2: Comparaison temps de calcul (secs) / mémoire (Mo) entre notre méthode et les graph cuts standards pour la segmentation d'images avec un modèle de Boykov/Jolly en connexité 1. Les images dont le nom est suffixé par « c » sont en couleur tandis que les autres sont en niveaux de gris. Le message **PM** (« Problème Mémoire ») indique que le graphe n'a pas pu être alloué.  $r^*$  correspond au rayon de la fenêtre pour lequel  $\rho$  est maximum (ici noté  $\rho^*$ ).



FIG. 2: Résultats obtenus pour les images décrites dans le Tableau 2 avec un modèle Boykov/Jolly en connexité 1. Les segmentations (ligne 2 et 4) ainsi que les graines (lignes 1 et 3) sont superposées par transparence aux images originales.

	$p \in A_{\sigma_a}$	$p \notin A_{\sigma_a}$
$E_p(u_p = 0)$	$-\log \left[ Pr(I(p)   p \in \mathcal{O}) \times \exp\left(-\left(\frac{d(p, \mathcal{O})}{\sigma_a}\right)^2\right)\right]$	$+\infty$
$E_p(u_p = 1)$	$-\log Pr(I(p)   p \in \mathcal{B})$	0

TAB. 3: Définition du terme de région pour le modèle GCSSLP.

coûteux à couper. Dans les expériences présentées, on fixe  $\sigma = 0.2$  pour toutes les images excepté pour T8 où  $\sigma = 2$  et T7 où  $\sigma = 0.05$ . L'ajustement automatique de ce paramètre est laissé comme perspective de travail.

Dans un premier temps, nous comparons les segmentations obtenues avec les vérités terrain fournies par un expert. Le Tableau 4 montrent les différences entre les deux segmentations selon plusieurs mesures d'évaluation<sup>1</sup>. Pour toutes les images, nous obtenons un coefficient de Dice qui reste supérieur à 70%.

Notre méthode est aussi évaluée qualitativement. La Figure 5 montre les segmentations obtenues pour des coupes uniformément réparties sur l'axe  $z$  pour les tumeurs T1, T8 et T9 (voir Figure 3). Afin d'illustrer la propagation des graines, les graines de la Figure 4 correspondent également à des coupes uniformément réparties sur  $z$  mais pour différentes hauteurs. La propagation évite ainsi de marquer chaque image successive du volume et permet donc un gain supplémentaire par rapport à l'existant. Comparé à la vérité terrain, la tumeur T1 est très proche tandis que les segmentations obtenues pour les tumeurs T8 et T9 diffèrent légèrement. Cette situation illustre bien la difficulté à segmenter de tels objet.

Les performances entre notre méthode et les graph cuts standards sont aussi comparées dans le Tableau 5 pour segmenter les tumeurs en utilisant les mêmes graines et les mêmes paramètres que précédemment. Les temps d'exécution sont mesurés de la même manière qu'en section 3. La proportion de graines objet par rapport au volume de la tumeur de la vérité terrain est indiqué. Cela fournit une mesure objective de la quantité d'interaction requise pour placer les graines. Les résultats montrent que notre méthode utilise de 7 à 500 fois moins de mémoire que les graph cuts standards tout en obtenant la même segmentation. Notons qu'une faible quantité de graines est nécessaire à la segmentation.

Généralement, le temps d'exécution dépend à la fois de la taille de la tumeur et de la capacité de l'utilisateur à positionner correctement les graines. La précision du résultat dépend directement de l'emplacement des graines mais peut être corrigé rapidement. Le calcul de la carte de distance, la réduction du graphe et le calcul de la coupe minimum prennent quelques secondes. La méthode proposée démontre donc sa capacité à segmenter précisément des tumeurs pulmonaires de manière interactive.

## Références

- [1] Y. Boykov and M-P. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images. In *ICCV*, volume 1, pages 105–112, 2001.
- [2] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for

energy minimization in vision. *IEEE Transactions on PAMI*, 26(9) :1124–1137, 2004.

- [3] C. Cigla and A.A. Alatan. Region-based image segmentation via graph cuts. In *ICIP*, pages 2272–2275, 2008.
- [4] A. Delong and Y. Boykov. A scalable graph-cut algorithm for N-D grids. In *CVPR*, pages 1–8, 2008.
- [5] L. Dice. Measure of the amount of ecological association between species. *Ecology*, 26(3) :297–302, 1945.
- [6] D. M. Greig, B. T. Porteous, and A. H. Seheult. Exact maximum a posteriori estimation for binary images. *Journal of the Royal Statistical Society*, 51(2) :271–279, 1989.
- [7] L. Ikonen. Pixel queue algorithm for geodesic distance transforms. In *DGCI*, volume 3429, pages 228–239, April 2005.
- [8] P. Kohli, V. Lempitsky, and C. Rother. Uncertainty driven multi-scale optimization. *Pattern Recognition*, 6376 :242–251, 2010.
- [9] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts ? *IEEE Transactions on PAMI*, 26(2) :147–159, 2004.
- [10] V. Lempitsky and Y. Boykov. Global optimization for shape fitting. In *CVPR*, pages 1–8, 2007.
- [11] N. Lermé, F. Malgouyres, and L. Létochart. Reducing graphs in graph cut segmentation. In *ICIP*, pages 3045–3048, 2010.
- [12] N. Lermé, F. Malgouyres, and J-M. Rocchisani. Fast and memory efficient segmentation of lung tumors using graph cuts. In *MICCAI – Third International Workshop on Pulmonary Image Analysis*, 2010.
- [13] Y. Li, J. Sun, CK. Tang, and HY. Shum. Lazy Snapping. *ACM Transactions on Graphics*, 23(3) :303–308, 2004.
- [14] H. Lombaert, Y.Y. Sun, L. Grady, and C.Y. Xu. A multilevel banded graph cuts method for fast image segmentation. In *ICCV*, volume 1, pages 259–265, 2005.
- [15] A.K. Sinop and L. Grady. Accurate banded graph cut segmentation of thin structures using laplacian pyramids. In *MICCAI*, volume 9, pages 896–903, 2006.
- [16] P. Strandmark and F. Kahl. Parallel and distributed graph cuts by dual decomposition. In *CVPR*, pages 2085–2092, 2010.
- [17] C. N. Vasconcelos, A. Sá, P. C. Carvalho, and M. Gattass. Using quadtrees for energy minimization via graph cuts. In *VMV*, pages 81–90, 2007.
- [18] X. Ye, G. Beddoe, and G. Slabaugh. Graph cut-based automatic segmentation of lung nodules using shape, intensity, and spatial features. In *Workshop on Pulmonary Image Analysis, MICCAI*, 2009.

<sup>1</sup>Une vue détaillée de ces mesures est disponible à l'adresse suivante : <http://lts08.bigr.nl/about.php>

Tumeur	Dice Co-efficient (%)	Volume Overlap (%)	Volume Différence (%)	Average Surface Distance	RMS Surface Distance	Maximum Surface Distance
T1	90.97	83.45	7.39	0.86	0.92	4.42
T2	80.95	67.99	4.98	1.25	1.54	6.63
T3	72.95	57.42	15.76	1.26	1.50	6.87
T4	71.33	55.44	42.31	3.30	4.01	14.34
T5	80.53	67.41	29.22	3.63	4.55	16.56
T6	86.63	76.42	18.02	1.30	1.49	5.90
T7	82.49	70.21	22.28	1.34	1.56	5.16
T8	89.25	80.59	9.59	1.20	1.47	9.32
T9	72.66	57.07	34.17	1.75	2.09	7.36
T10	74.04	58.79	41.09	4.97	5.55	15.99
Moyenne	80.18	67.47	22.48	2.08	2.46	9.25

TAB. 4: Comparaison entre notre les segmentations obtenues par notre méthode et les vérités terrain fournies par l'expert.

Tumeur	Taille	Graph cuts standards		Notre méthode		Quantité de graines objet (%)
		Temps	Mémoire	Temps	Mémoire	
T1	512 × 512 × 50	4.08	472.34	2.00	24.71	2.52
T2	512 × 512 × 50	4.89	573.05	2.71	83.42	2.71
T3	512 × 512 × 50	4.90	580.78	2.87	83.42	2.47
T4	512 × 512 × 50	5.34	729.72	2.14	37.07	12.99
T5	512 × 512 × 50	5.36	737.41	2.10	36.37	10.22
T6	512 × 512 × 50	10.18	1476.30	3.80	37.07	3.15
T7	512 × 512 × 50	5.16	544.74	3.21	83.42	8.86
T8	512 × 512 × 316	<b>PM</b>	43091.25	68.93	83.42	2.45
T9	512 × 512 × 50	4.24	496.79	2.39	37.07	8.01
T10	512 × 512 × 70	10.36	1151.74	5.42	125.13	9.19

TAB. 5: Temps d'exécution (secs) et consommation mémoire (Mo) des graph cuts avec et sans réduction. Le message **PM** (« Problème Mémoire ») indique que le graphe n'a pas pu être alloué.



FIG. 3: Contexte global des tumeurs pulmonaires T1 (gauche), T8 (milieu) et T9 (droite). Les vérités terrain sont superposées aux images originales en rouge clair.

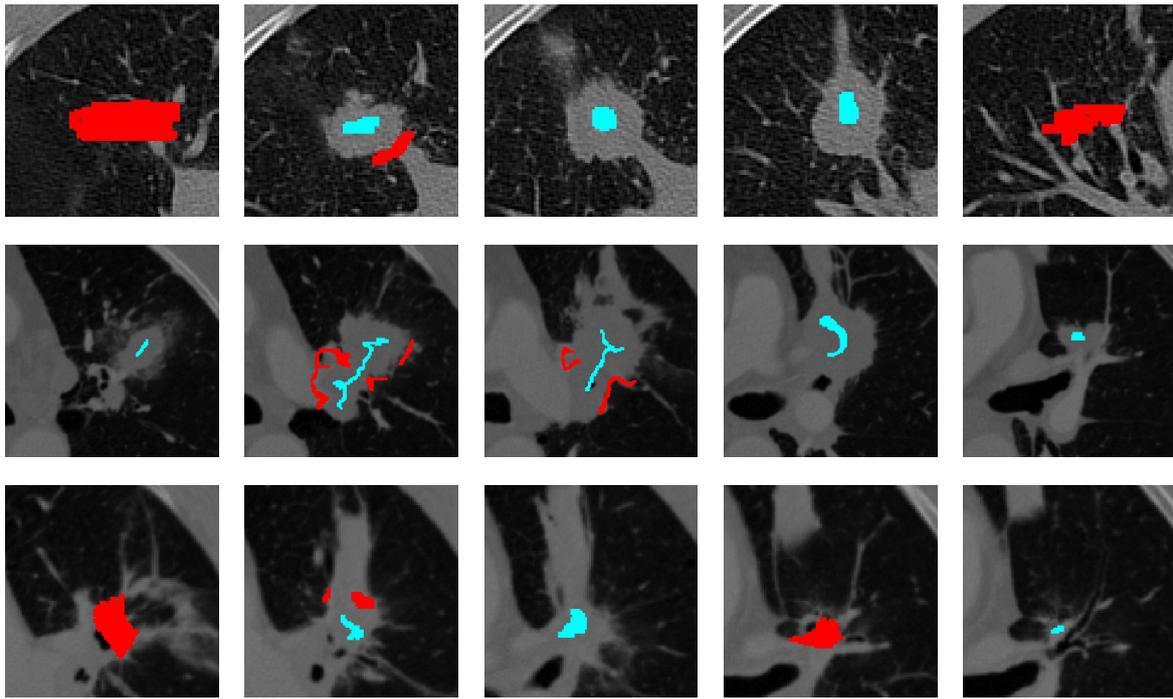


FIG. 4: Emplacement des graines pour la segmentation des tumeurs T1 (ligne du haut), T8 (ligne du milieu) et T9 (ligne du bas). Les graines objet (cyan) et les graines du fonds (rouge) sont superposées à l'image originale pour des coupes successives.

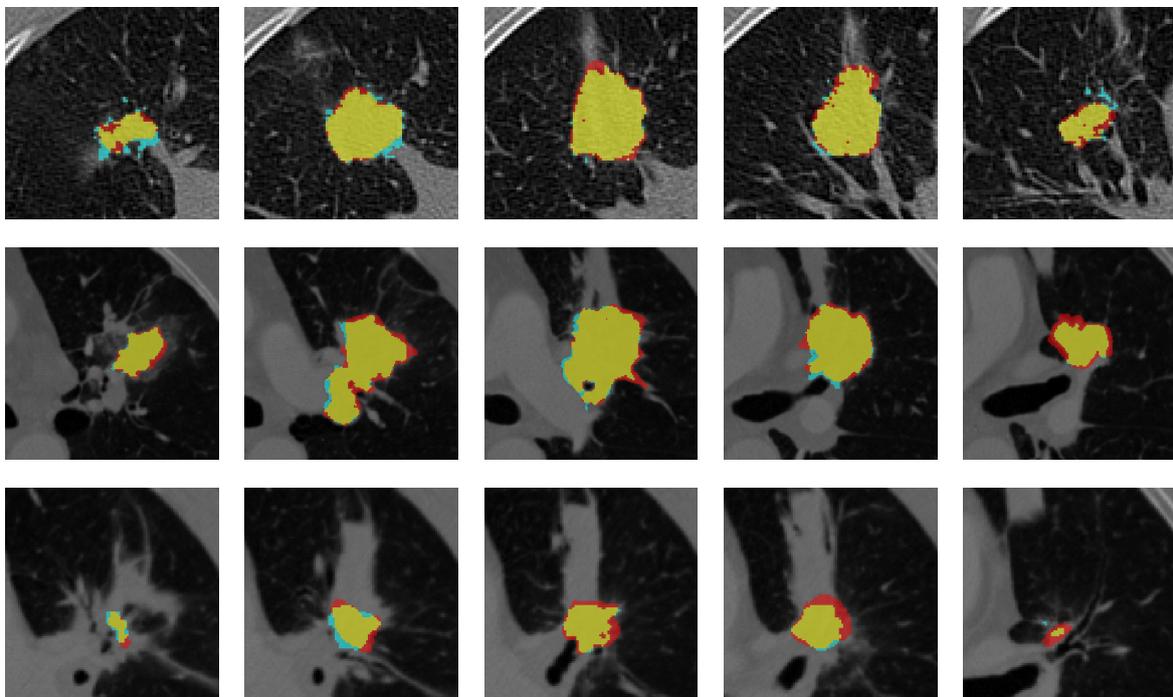


FIG. 5: Segmentation de tumeurs pulmonaires T1 (ligne du haut), T8 (ligne du milieu) and T9 (ligne du bas), pour des coupes ne contenant aucun points des graines. La vérité terrain (rouge) et la segmentation (cyan) sont superposées sur l'image originale. L'intersection est de couleur jaune.