Groupement de composantes connexes à l'aide de chemins minimaux
Grouping connected components using minimal path techniques

Laurent Cohen Thomas Deschamps

Cérémade, Université Paris Dauphine, place du Maréchal de Lattre de Tassigny 75775 Paris

cohen@ceremade.dauphine.fr

Résumé

On s'intéresse à la recherche d'un ensemble de contours dans une image 2D ou 3D. On considère le problème du groupement perceptuel et de la complétion de contours, lorsque la donnée est un ensemble non structuré de régions dans une image. Nous présentons une nouvelle méthode pour trouver des courbes complètes à partir d'un ensemble de points sur ces contours. Les contours sont définis comme des chemins minimaux reliant des composantes connexes que l'on calcule à l'aide de l'algorithme du fast marching. On détermine les chemins minimaux entre chacune de ces composantes, jusqu'à ce que l'ensemble de ces régions soient connectées. Les chemins sont obtenus à l'aide d'une descente de gradient à partir des points selle de la carte d'action minimale entre chaque couple de composantes.
Nous étendons ensuite notre méthode au 3D. La donnée est dans ce cas un ensemble de composantes connexes dans une image 3D, et on extrait les chemins minimaux qui relient entre elles ces composantes. Et en utilisant comme mesure de potentiel une fonction basée sur la réponse d'un filtre adapté à la détection de structures tubulaires, nous montrons une application de notre méthode à la recherche de ce type de structures dans des images médicales 3D.

Mots Clef

Groupement perceptuel, contours actifs, chemins minimaux, fast-marching, Ensembles de niveaux, minimisation d'´energie, imagerie m´edicale 3D.

Abstract

We address the problem of finding a set of contour curves in a 2D or 3D image. We consider the problem of perceptual grouping and contour completion, where the data is an unstructured set of regions in the image. A new method to find complete curves from a set of edge points is presented. Contours are found as minimal paths between connected components, using the fast marching algorithm. We find the minimal paths between each of these components, until the complete set of these "regions" is connected. The paths are obtained using backpropagation from the saddle points to both components. We then extend this technique to 3D. The data is a set of connected components in a 3D image. We find 3D minimal paths that link together these components. Using a potential based on vessel detection, we illustrate the capability of our approach to reconstruct tree structures in a 3D medical image dataset.

1 Introduction

On s'int´eresse au groupement perceptuel, et à trouver un ensemble de contours dans une image, à l'aide de courbes d'´energie minimale. Depuis leur introduction, les contours actifs [8] ont ´et´e utilis´es dans de nombreux travaux pour trouver les contours d'un objet dans une image, par l'interm´ediaire d'une minimisation d'´energie. Afin d'obtenir un ensemble de contours dans une image, on a besoin d'initialiser un grand nombre de contours. La formulation du problème à l'aide des contours acitfs par Ensemble de Niveau [11], [2], permet des changements de topologie: elle permet ainsi d'obtenir une multitude de contours à partir d'un seul. Mais ces m´ethodes ne donnent pas de r´esultats satisfaisants, car lorsque la donn´ee est incomplète, les contours peuvent se propager et fuire là où la donn´ee est manquante, se s´eparant en une multitude de contours ind´esirables, lorsqu'on cherche à obtenir un unique contour. Ce problème relève du groupement perceptuel, où l'initialisation est un ensemble de contours à compl´eter. Par exemple, dans une image comme celle de la figure 1, où est repr´esent´ee une forme incomplète, le système de vision humain complète facilement les donn´ees manquantes et extrait la courbe complète. Le groupement perceptuel est un problème qui à d´eja fait l'objet de nombreux travaux par le pass´e. Il a ´et´e reformul´e r´ecemment à l'aide de m´ethodes de minimisation d'´energie dans [13, 7, 14]. Ces m´ethodes associent une mesure de repr´esentativit´e à chaque composante d'une courbe, ou à chaque point de l'image. La d´efinition d'une mesure de repr´esentativit´e est alors bas´ee indirectement sur une ´energie de r´egularisation du second ordre du chemin contenant un de ces points, comme pour les snakes ([8]). Dans ce cas, les courbes r´esultantes sont obtenues dans un deuxième temps comme des lignes de crètes de la mesure de repr´esentativit´e, après seuillage. C'est cette relation ´etroite entre les courbes d'´energie minimale, comme les snakes, et la compl´etion de contours qui nous amène ici à vouloir d´etecter un ensemble de contours pour faire la compl´etion sur une image, en les d´efinissant comme un ensemble de courbes d'´energies minimales.

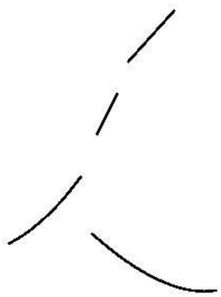

FIGURE 1 – *Exemple de régions connexes à relier*

Pour trouver les minima globaux pour les contours actifs, les auteurs de [3] ont utilis´e des chemins minimaux, introduits auparavant dans [10, 9]. L'obj´ectif ´etait d'´eviter de trouver des minima locaux, et de simplifier l'´etape d'initialisation classique et fastidieuse des snakes [1], en la remplaçant par la localisation des deux extremit´es du contour recherch´e. Le sch´ema num´erique a l'avantage d'être consistant (voir [3]) et efficace, à l'aide de l'algorithme du *Fast-Marching* introduit dans [12]. Dans cet article, nous proposons d'utiliser cette approche de chemins minimaux dans le but d'extraire un ensemble de contours à partir d'un ensemble de points dans une image, Afin de trouver l'ensemble des contours les plus repr´esentatifs dans l'image, nous cherchons les chemins minimaux entre couples de composantes connexes. Cette approche est ensuite ´etendue à la compl´etion de structures tubulaires dans des images 2D et 3D. Le problème est dans ce cas de compl´eter un objet, à partir de composantes connexes pr´e-d´et´ect´ees de cet objet .

Pour le groupement perceptuel, le potentiel P à minimiser le long des chemins est souvent une image de points de bords, formant des contours incomplets, comme sur la figure 1. Ces points de bords sont repr´esent´es par une image binaire, avec des faibles valeurs de potentiels le long des bords, et des fortes valeurs dans le fond de l'image. Le potentiel peut aussi être d´efini comme une fonction du gradient de l'image elle-même, $P = g(\|\nabla I\|)$, comme pour les contours actifs classiques. Il peut aussi être une image de niveaux de gris comme dans [3], ou une fonction plus complexe des niveaux de gris. Dans nos exemples de structures vasculaires en 2D et en 3D, on se servira d'un potentiel bas´e sur une mesure du Hessien de l'image [6].

Le plan de l'article est le suivant: Nous faisons tout d'abord un r´esum´e des chemins minimaux et du *Fast-Marching* en 2D et 3D, dans la section 2. Puis nous montrons en section 3 comment trouver un ensemble de courbes à partir d'un ensemble non structur´e de points. En regrouppant les points en composantes connexes, nous proposons une m´ethode pour trouver les couples de composantes connexes reli´ees entre elles par des chemins minimaux. Nous ´etendons ensuite cette m´etode au 3D, et nous montrons notamment des exemples sur des images m´edicales.

2 Chemins minimaux 2D et 3D

2.1 Minimum Global pour les contours actifs

Dans cette section sont d´etaill´ees les id´ees de base de la m´ethode introduite dans [3] pour trouver le minimum global de l'´energie d'un contour actif à l'aide des chemins minimaux. L'´energie à minimiser est similaire à celle des modèles d´eformables classiques ([8]), et elle combine des termes de lissage et des termes d'attraction dans des zones de l'image (Potentiel P):

$$E(C)=\int_\Omega\left\{w_1\|C'(s)\|^2+w_2\|C''(s)\|^2+P(C(s))\right\}ds \quad (1)$$

où $C(s)$ repr´esente une courbe dans une image 2D et Ω est son domaine de d´efinition. Les auteurs de [3] ont reli´e ce problème avec la formulation de type Contour actifs par Ensemble de niveau [12]. En particulier, sa formulation d'Euler est ´equivalente à celle des contours actifs g´eod´esiques[2]. La m´ethode introduite dans [3] am´eliore le processus de minimisation de l'´energie car le problème est devenu une recherche d'un minimum global.

2.2 Formulation du problème

Comme dans [3], nous sommes amen´es à minimiser

$$E(C) = \int_{\Omega=[0,L]} \{w + P(C(s))\} ds, \quad (2)$$

où s est l'abscisse curviligne ($\|C'(s)\| = 1$). La r´egularisation de ce modèle est maintenant faite par la constante $w > 0$ (voir [3] pour les d´etails). Etant donn´e un potentiel $P \geq 0$, l'´energie est comme une distance pond´er´ee par $\tilde{P} = P + w$. L'action minimale $\mathcal{U}$ est d´efinie comme l'´energie minimale int´egr´ee le long d'un chemin entre un point de d´epart p_0 et n'importe quel point p par:

$$\mathcal{U}(p) = \inf_{\mathcal{A}_{p_0,p}} E(C) = \inf_{\mathcal{A}_{p_0,p}} \left\{ \int_\Omega \tilde{P}(C(s))ds \right\} \quad (3)$$

où $\mathcal{A}_{p_0,p}$ est l'ensemble des chemins admissibles entre p_0 et p. Le chemin minimal entre p_0 n'importe quel point p_1 de l'image peut être facilement d´eduit du calcul de l'action, par une simple d´escente de gradient sur $\mathcal{U}$ depuis p jusqu'à p_0.

2.3 Résolution du Fast Marching

Afin de calculer $\mathcal{U}$, une ´equation de propagation de front li´ee à l'´equation (3) est r´esolue:

$$\frac{\partial C}{\partial t} = \frac{1}{\tilde{P}} \vec{n}. \quad (4)$$

Cette ´equation fait ´evoluer un front, en partant d'un cercle infinit´esimal autour de p_0 jusqu'à ce que chaque point de l'image soit visit´e et se voit donner une valeur pour $\mathcal{U}$. La valeur de $\mathcal{U}(p)$ est le temps t auquel le front passe par p.

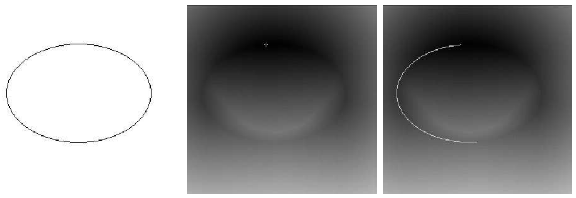

FIGURE 2 – *Trouver le chemin minimal entre deux points. Partie gauche: le potentiel est minimal sur l'ellipse. Milieu: l'action minimale ou distance pondérée au point de départ. Partie droite: chemin minimal par rétro-propagation, à partir du second point.*

La technique du *Fast Marching*, introduite dans [12], fut utilisée dans [3], en se basant sur le fait que $\mathcal{U}$ satisfait l'équation *Eikonal*:

$$\|\nabla \mathcal{U}\| = \tilde{P} \quad \text{et} \quad \mathcal{U}(p_0) = 0. \tag{5}$$

Les schémas de différences finies classiques ont tendance à diverger et sont instables. Un schéma décentré amont, proposé dans [12], permet de stabiliser le schéma numérique en ne dérivant que dans une seule direction:

$$\begin{aligned}(\max\{u - \mathcal{U}_{i-1,j}, u - \mathcal{U}_{i+1,j}, 0\})^2 &+ \\ (\max\{u - \mathcal{U}_{i,j-1}, u - \mathcal{U}_{i,j+1}, 0\})^2 &= \tilde{P}^2_{i,j},\end{aligned} \tag{6}$$

et donnant la solution de viscosité correcte u pour $\mathcal{U}_{i,j}$. L'amélioration apportée par le *Fast Marching* est d'introduire un ordre de sélection des points de la grille. Cet ordre est basé sur le fait que l'information se propage vers l'extérieur dans une seule direction, car l'action peut seulement augmenter du fait de l'équation quadratique (6).
Cette technique de ne considérer à chaque pas que l'ensemble nécessaire de points de la grille fut originellement introduit pour la construction de chemins de longueur minimale dans un graphe entre deux noeuds du graphe [5]. L'algorithme est détaillé dans la Table 1. Un exemple est montré sur la Figure 2.2. Le *Fast Marching* sélectionne à chaque itération le point *Trial* avec la valeur d'action minimum. Pour calculer cette valeur, on doit résoudre l'équation (7) pour chaque point *Trial*, comme précisé en section 2.4.

2.4 Algorithme du schéma 2D décentré amont

Remarquez que pour résoudre l'équation (6), seuls les points *Alive* sont considérés. Considérant les voisins du point (i,j) en 4-connexité, on note $\{A_1, A_2\}$ et $\{B_1, B_2\}$ les deux couples de voisins opposés tels que nous puissions les ordonner comme $\mathcal{U}(A_1) \leq \mathcal{U}(A_2)$, $\mathcal{U}(B_1) \leq \mathcal{U}(B_2)$, et $\mathcal{U}(A_1) \leq \mathcal{U}(B_1)$. Considérant que nous avons $u \geq \mathcal{U}(B_1) \geq \mathcal{U}(A_1)$, on en dérive l'équation

$$(u - \mathcal{U}(A_1))^2 + (u - \mathcal{U}(B_1))^2 = \tilde{P}^2_{i,j} \tag{7}$$

Basé sur le test du discriminant Δ de l'équation (7), un ou deux voisins servent à la résoudre:

1. Si $\tilde{P}_{i,j} > \mathcal{U}(B_1) - \mathcal{U}(A_1)$, le solution de l'équation (7) est $u = \frac{\mathcal{U}(B_1)+\mathcal{U}(A_1)+\sqrt{2\tilde{P}^2_{i,j}-(\mathcal{U}(B_1)-\mathcal{U}(A_1))^2}}{2}$.
2. sinon $u = \mathcal{U}(A_1) + \tilde{P}_{i,j}$.

Algorithme du Fast Marching en 2D

- Définitions:
  - points *Alive*: points où la valeur de l'action $\mathcal{U}$ est minimale et ne changera pas;
  - points *Trial*: prochains points de la grille à être examinés (voisins en 4-connexité). un estimé U de $\mathcal{U}$ a été calculé en ces points à l'aide de l'équation (6) seulement à partir de points *Alive*;
  - points *Far*: tous les autres points de la grille, et qui n'ont pas d'estimation de U;
- Initialisation:
  - points *Alive*: le point de départ p_0, $U(p_0) = \mathcal{U}(p_0) = 0$;
  - points *Trial*: réduit aux 4 voisins p de p_0 avec comme valeur intiale $U(p) = \tilde{P}(p)$ ($\mathcal{U}(p) = \infty$);
  - points *Far*: tous les autres points, $\mathcal{U} = U = \infty$;
- Boucle:
  - Soit $p = (i_{min}, j_{min})$ le point *Trial* avec l'action U la plus faible;
  - on l'enlève des points *Trial* pour le mettre avec les points *Alive*;
  - Pour chaque voisin (i,j) de (i_{min}, j_{min}):
    - Si (i,j) est *Far*, on l'ajoute à l'ensemble *Trial*;
    - Si (i,j) est *Trial*, on met à jour $U_{i,j}$ avec l'équation (6).

TABLEAU 1 – *Algorithme du Fast Marching*

2.5 Chemins minimaux en 3D

Une extention 3D du *Fast Marching* a été presentée dans [4]. De manière similaire, l'action minimale $\mathcal{U}$ est définie par

$$\mathcal{U}(p) = \inf_{\mathcal{A}_{p_0,p}} \left\{ \int_\Omega \tilde{P}(C(s))ds \right\} \tag{8}$$

où $\mathcal{A}_{p_0,p}$ est maintenant l'ensemble des chemins 3D admissibles entre p_0 et p. Etant donné un point de départ p_0 pour calculer $\mathcal{U}$ on commence à partir d'un front circulaire infinitésimal autour de p_0. Le schéma numérique 2D de l'equation (6) est étendu au 3D, pour donner:

$$\begin{aligned}(\max\{u - \mathcal{U}_{i-1,j,k}, u - \mathcal{U}_{i+1,j,k}, 0\})^2 &+ \\ (\max\{u - \mathcal{U}_{i,j-1,k}, u - \mathcal{U}_{i,j+1,k}, 0\})^2 &+ \\ (\max\{u - \mathcal{U}_{i,j,k-1}, u - \mathcal{U}_{i,j,k+1}, 0\})^2 &= \tilde{P}^2_{i,j,k}\end{aligned} \tag{9}$$

donnant la solution de viscosité correcte u pour $\mathcal{U}_{i,j,k}$.

3 Extraire des Contours à partir d'un ensemble de Composantes Connexes R_k

3.1 Chemin Minimal entre deux Régions

La méthode de [3], détaillée dans la section précédente, permet de trouver un chemin minimal entre deux points.

C'est une extension facile de définir un chemin minimal entre deux régions d'une image. Prenons deux régions connexes d'une image, R_0 et R_1, et considérons R_0 comme la région de départ et R_1 comme un ensemble de points d'arrivée. Le problème est donc de trouver un chemin qui minimise une énergie sur l'ensemble des chemins qui ont un point de départ dans R_0 et un point d'arrivée dans R_1. L'action minimale est maintenant définie par

$$\mathcal{U}(p) = \inf_{\mathcal{A}_{R_0,p}} E(C) = \inf_{p_0 \in R_0} \inf_{\mathcal{A}_{p_0,p}} E(C) \qquad (10)$$

où $\mathcal{A}_{R_0,p}$ est l'ensemble des chemins commençant avec un point de R_0 et se terminant en p. L'action minimale peut être calculée de la même manière qu'avant dans la table 1, avec l'ensemble *Alive* initialisé par l'ensemble des points de R_0, avec $\mathcal{U} = 0$ et l'ensemble *Trial* étant l'ensemble des voisins en 4-connexité des points de R_0 qui ne sont pas dans R_0. La rétro-propagation par descente de gradient sur $\mathcal{U}$ à partir de n'importe quel point p de l'image donnera le chemin minimal qui joint ce point à la région R_0.

De manière à trouver le chemin minimal entre les régions R_1 et R_0, on détermine un point $p_1 \in R_1$ tel que $\mathcal{U}(p_1) = \min_{p \in R_1} \mathcal{U}(p)$. Ensuite, on rétro-propage de p_1 vers R_0 pour trouver le chemin minimal entre p_1 et R_0, qui est aussi chemin minimal entre R_1 et R_0.

3.2 Chemins minimaux entre un ensemble de composantes connexes

Nous sommes maintenant intéressés par trouver une multitude de contours dans une image. Supposons que nous ayons un ensemble initial de contours, provenants d'une étape préliminaire de détection. On note R_k les différentes composantes connexes de ces contours.

On se propose de trouver les contours comme l'ensemble des chemins minimaux qui relient des paires de régions parmis les R_k. Si nous savons quelles paires de régions doivent être reliées entre elles, trouver l'ensemble des contours serait une application triviale de la section précédente. Le problème qui nous intéresse ici est aussi de trouver quelles paires de régions doivent être connectées entre elles. Comme l'ensemble des contours R_k est donné de manière non structurée, on ne sait pas par avance comment les régions sont connectées. C'est le probleme principal que l'on s'efforcera de résoudre à l'aide des chemins minimaux.

3.3 Méthode

Notre approche est similaire à calculer la carte de distance à un ensemble de régions, et leur diagramme de Voronoi. Dans notre cas, nous utilisons une distance pondérée, définie à travers un potentiel P. Cette distance est obtenue comme la carte d'action minimale, vis-à-vis de P, avec des valeurs d'action nulles à tous les points des régions R_k. Au lieu de calculer une carte d'action minimale pour chaque paire de régions, comme dans la section 3.1, on a juste besoin de calculer une carte d'action minimale pour trouver tous les chemins. En même temps que la carte d'action est calculée, on détermine les paires de régions qui doivent être reliées entre elles. Cette méthode est basée sur la détection des points de rencontre des fronts de propagation. Il s'agit des *points selles* de l'action minimale $\mathcal{U}$. Dans la section 2, on montrait que le calcul de l'action minimale peut être vu comme la propagation d'un front à l'aide de l'équation (4). Comme l'action minimale est calculée avec le *Fast-Marching*, les ensembles de niveaux de $\mathcal{U}$ donnent l'évolution du front. Pendant l'exécution du *Fast-Marching*, la frontière de l'ensemble des points *Alive* donne la positoin du front. Dans la section précédente, nous avions un seul front évoluant à partir de la région de départ R_0. Comme tous les points p des régions R_k sont données avec $\mathcal{U}(p) = 0$, on a maintenant un front évoluant à partir de chaque région de départ R_k. Par la suite, on désignera par rencontre de fronts aussi bien la position du point où les fronts de deux régions différentes se rencontrent, que le premier point *Alive* dans l'algorithme discret qui connecte deux composantes connexes (voir Figures 3 et 4).

On utilise le fait que étant données deux régions R_1 et R_2, le point selle S où les deux fronts partant de chaque région se rencontre peut être utilisée pour trouver les chemin minimal entre R_1 et R_2. En fait, le chemin minimal entre deux régions doit passer par le point de rencontre S. Ce point est le milieu (en terme d'énergie) du chemin minimal entre R_1 et R_2. La rétro-propagation de S vers R_1 et de S vers R_2 donne les deux moitié du chemin.

3.4 Notations et definitions

Voici quelques définitions utiles pour la suite.

X étant un ensemble de points de l'image, $\mathcal{U}_X$ est l'action minimale obtenue par *Fast-Marching* avec le potentiel $\tilde{P}$ et les points de départ $\{p, p \in X\}$. Cela signifie que tous les points de X sont initialisés *Alive* avec une valeur d'action nulle. Tous leurs voisins en 4-connexité qui ne sont pas dans X sont des points *Trial*. On peut ainsi voir que $\mathcal{U}_X = \min_{p \in X} \mathcal{U}_p$. X peut être aussi bien une composante connexe qu'un ensemble de composantes connexes.

Le *label* l d'un point p est égal à l'index k de la région R_k pour p plus proche en énergie de R_k que de n'importe quelle autre région R_j. Cela signifie que l'action minimale $\mathcal{U}_{R_k}(p) \leq \mathcal{U}_{R_j}(p), \forall j \neq k$. On définit la région $L_k = \{p/l(p) = k\}$. Si $X = \cup_j R_j$, on a $\mathcal{U}_X = \mathcal{U}_{R_k}$ sur L_k et le calcul de $\mathcal{U}_X$ est le même que le calcul simultané de chaque $\mathcal{U}_{R_k}$ sur chaque région L_k. Il s'agit des fronts simultanés partant de chaque R_k.

Un *point selle* $S(R_i, R_j)$ entre R_i et R_j est le premier point où le front partant de R_i pour calculer $\mathcal{U}_{R_i}$ rencontre le front partant de R_j pour calculer $\mathcal{U}_{R_j}$; en ce point, $\mathcal{U}_{R_i}$ et $\mathcal{U}_{R_j}$ sont égales et il s'agit de la plus petite valeur pour laquelle elles sont égales..

Deux régions différentes parmis les R_k seront dites *régions connectées*, si elles sont sélectionnées pour être reliées entre elles. Connecter des régions, c'est sélectionner un ensemble de points selles. Alors les régions R_i et R_j sont des *régions connectées* si leur *point-selle* est parmis les points sélectionnés.

Un *cycle* est une séquence de différentes régions $R_k, 1 \leq$

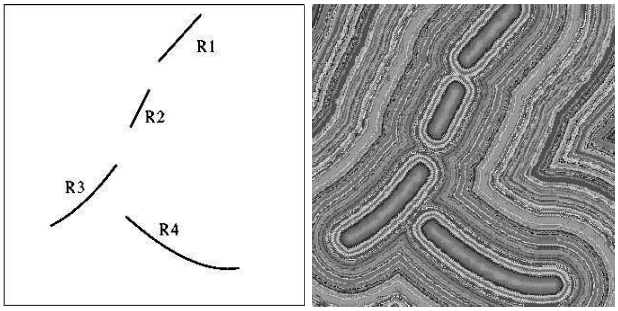

FIGURE 3 – *Carte d'Action Minimale pour les 4 régions de l'exemple de la figure 1. On utilise une table de couleurs aléatoires pour montrer les ensembles de niveaux.*

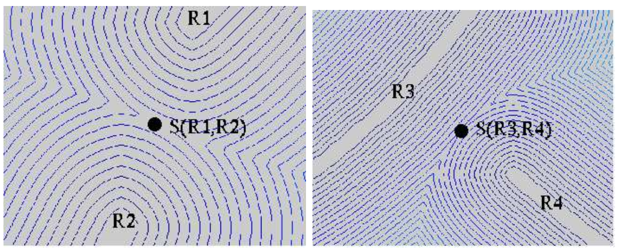

FIGURE 4 – *Zoom sur les points selles entre les regions.*

$k \leq K$, telles que pour $1 \leq k \leq K-1$, R_k et R_{k+1} sont des *régions connectées* et R_k et R_1 sont aussi des *régions connectées.*

3.5 Trouver et sélectionner les Points Selles

Le but principal de notre méthode est d'obtenir tous les chemins significatifs qui relient les régions. Chaque région ne doit pas être connectée à toutes les autres régions, mais uniquement à celles qui sont proches, au sens de l'énergie. Il y a plein de possibilités pour décider quelles régions doivent se connecter, qui dépendent du type de données, et d'application. Dans certains cas, le but pourrait être de détecter les courbes fermées et de ne pas créer d'embranchements. Le critère serait alors de contraindre une région à se relier avec au plus deux régions, dans le but de créer des *cycles*. Dans notre contexte, nous sommes intéressés par détecter des structures arborescentes, et nous voulons éviter les *cycles*. Par conséquent, le critère pour que deux régions R_i et R_j se connectent est que leurs fronts en se rencontrant ne créent pas de *cycle*.

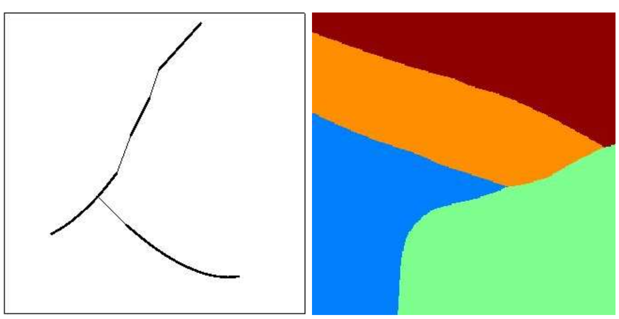

FIGURE 5 – *Exemple avec 4 régions. A gauche, on voit les chemins minimaux obtenus par rétro-propagation à partir des trois points selles de chacune des régions d'où le front vient; A droite, le diagramme de Voronoi généré.*

On peut voir sur la figure 4 un zoom sur le point selle détecté entre les régions R_1 et R_2, ainsi qu'entre R_3 et R_4. Une fois qu'un *point selle* $S(R_i, R_j)$ est trouvé et sélectionné, la rétro-propagation relativement à l'énergie $\mathcal{U}$ doit être faite dans les directions de R_i et de R_j pour extraire les deux parties du chemin entre elles. On voit sur la Figure 5 la rétro-propagation de chacun des trois *points selles* sélectionnés automatiquement. Ils relient R_1 à R_2, R_2 à R_3 et R_3 à R_4. Sur un point selle, le gradient de U est nul, mais les directions de descente vers chaque région sont opposées. Pour chaque rétro-propagation la direction de descente est celle relative à la région associée. Cela signifie que pour estimer la direction du gradient pour R_i, on met l'action de tous les points du voisinage qui ne sont pas dans L_i artificiellement à zéro. Cela permet de trouver la bonne direction pour la descente de gradient vers R_i. D'autre part, ces rétro-propagations doivent être faites uniquement pour les *points selles* sélectionnés. Dans l'algorithme du *Fast-Marching*, il existe une façon simple de trouver les *points selles* et de mettre à jour les *régions connectées*.

Par définition, la région L_k associée avec une région R_k est l'ensemble des points p de l'image tels que l'énergie minimale $\mathcal{U}_{R_k}(p)$ vers R_k est plus faible que toutes les énergies $\mathcal{U}_{R_j}(p)$ vers les autres régions R_j. L'ensemble de ces régions L_k recouvre totalement image, et forme le diagramme de Voronoi de cette image (voir figure 5). Tous les *points selles* sont à la frontière entre les régions L_k. Pour un point p de la frontière entre L_j et L_k, on a $\mathcal{U}_{R_k}(p) = \mathcal{U}_{R_j}(p)$. Le *point selle* $S(R_k, R_j)$ est le point de la frontière de valeur minimale pour $\mathcal{U}_{R_k}(p) = \mathcal{U}_{R_j}(p)$. Cela nous donne un critère pour trouver les *points selles* pendant l'exécution du *Fast-Marching*.

Chaque fois que deux fronts venant de R_k et R_j se rencontre pour la permière fois, on détermine le point de rencontre $S(R_k, R_j)$. Cela signifie que ce dont nous avons besoin pour chaque point de l'image, c'est de savoir d'où il vient. Il est facile de garder en mémoire cette origine en créant une carte d'indices, mise à jour chaque fois qu'un point *Alive* est créé dans l'algorithme. Chaque point de la région R_k commence avec le label k. Chaque fois qu'un point devient *Alive*, il prend le label des points qui ont participé au calcul de son action dans l'équation (6). Dans cette équation , le calcul de $U_{i,j}$ dépend seulement de au maximum deux des quatre pixels considérés. Ces deux pixels, dits A_1 et B_1, doivent avoir le même *label*, sauf si (i, j) se trouve sur la frontière enter les deux labels. Si A_1 et B_1 sont tous deux *Alive* et avec des labels différents k et l, cela signifie que les régions R_k et R_l se rencontre en ce

Chemin Minimaux entre les Régions R_k

- Initialisation:
  - Étant donnés les R_k
  - $\forall k, \forall p \in R_k, V(p) = 0; l(p) = k;$ p est *alive*.
  - $\forall p \notin \cup_k R_k, V(p) = \infty; l(p) = -1;$ p est *far* excepté les voisins en 4-connexité de R_k qui sont *Trial* avec U estimé à l'aide de l'équation 6.
- Boucle pour le calcul de $V = \mathcal{U}_{\cup_k R_k}$:
  - Soit $p = (i_{min}, j_{min})$ le point *Trial* avec l'action U la plus faible;
  - On l'enlève des points *Trial* pour le mettre dans les points *Alive* avec V(p) = U(p);
  - on met a jour $l(p)$ avec le même index que le point A_1 dans l'équation (6). Si $R(A_1) \neq R(B_1)$ et que nous sommes sans le cas 1 de la section 2.4 où deux points sont utilisés et s'il s'agit de la première fois que les régions des labels $l(A_1)$ et $l(B_1)$ se rencontrent, $S(R_{l(A_1)}, R_{l(B_1)}) = p$ est défini comme le *point selle* entre $R_{l(A_1)}$ et $R_{l(B_1)}$. Si ajouter un lien entre ces régions ne crée pas de *cycle*, elles sont considérées comme *régions connectées* et $S(R_{l(A_1)}, R_{l(B_1)}) = p$ est sélectionné,
    Pour chaque voisin (i, j) de (i_{min}, j_{min}):
    - Si (i, j) est *Far*, on l'ajoute aux points *Trial*;
    - Si (i, j) est *Trial*, on met à jour l'action $U_{i,j}$.
- On extrait tous les chemins entre les *régions connectées* sélectionnées par rétro-propagation de chaque coté de leurs *point selles* (see Section 3.5).

TABLEAU 2 – *Algorithme de la Section 3*

point. Si cela arrive pour la première fois, le point courant est défini comme étant le *point selle* $S(R_k, R_l)$ entre ces deux régions. Un point de la frontière entre R_k et R_l prend le label du voisin avec l'action la plus faible. A la frontière entre deux labels, l'étiquetage est donc voxélique, et l'erreur reste donc faible et peu importante dans notre contexte

3.6 Algorithme

L'algorithme de cette section est décrit dans la table 2 et est illustré par les figures 3 à 5. Quand il y a un grand nombre de régions R_k, cela ne modifie pas beaucoup les temps de calculs de la carte d'action minimale, mais cela rend plus complexe la manipulation des *régions connectées* et des *points selles*, ainsi que la détection des *cycles*.
Notre méthode pour détecter les cycles est la suivante. Étant donné un point selle trouvé pour les régions R_i et R_j. On teste s'il existe déjà un lien entre ces régions; cela revient à chercher une suite de régions distinctes $R_k, 1 \leq k \leq K$, avec $R_1 = R_i$ et $R_K = R_j$, telles que pour $1 \leq k \leq K-1$, R_k et R_{k+1} soient des *régions connectées*.
Ce genre de condition peut être facilement implémenté à l'aide d'un algorithme récursif. Quand les fronts de deux régions R_i et R_j se rencontrent, un tableau repertorie les connexions entre régions et permet de savoir si un lien direct ou indirect existe déjà entre elles. Etant donné N régions différentes, on remplit une matrice $M(N, N)$ de zéros, et chaque fois que deux régions R_i et R_j se rencontrent sans créer de cycle, on met $M(i, j) = M(j, i) = 1$. Lorsque deux fronts se rencontrent, on applique l'algorithme détaillé dans la table 3.

Algorithm de détection de cycle Quand une région R_i rencontre une région R_j: *Test*(i, j, M, i); *avec* *Test*(i, j, M, l);

- si $M(l, j) = 1$, on renvoit 1;
- sinon
  - count = 0;
  - for $k \in [1, N]$ avec $k \neq i, k \neq j, k \neq l$: count $+=$ *Test*(k, j, M, l);
  - on renvoit count;

TABLEAU 3 – *Détection de cycle*

Si deux régions sont déjà connectées, le point où leurs fronts se rencontrent n'est pas considéré comme un candidat valable pour la rétro-propagation. L'algorithme s'arrête automatiquement quand toutes les régions sont connectées, sans cycles.

3.7 Application

La méthode peut être appliquée à des composantes connexes pour un ensemble de points de bords, ou des points issus d'une détection préliminaire. Trouver tous les chemins à partir d'un ensemble de points est intéressant dans le cas d'un potentiel binaire, cas décrit sur la figure 3, pour le groupement perceptuel. Elle peut aussi être utilisée quand un ensemble caractéristique de points a été extrait à l'aide d'un traitement préliminaire particulier. Par exemple, sur la figure 6, on peut voir une image 2D au niveau de la hanche sur laquelle on s'interesse aux veines. Le potentiel est défini en se servant de idées développées dans [6] sur les filtres à base de Hessien (détaillées dans la section 4.2).

4 Trouver un ensemble de chemins dans une image 3D

4.1 Extension au 3D

On peut étendre notre approche à la recherche d'ensemble de chemins minimaux 3D entre régions dans des images tridimensionnelles. Toutes les définitions et les algorithmes de la section 3 ne sont pas modifiés par ce changement de dimension du problème. La principale différence est que les algorithmes comme le *Fast-Marching* utilisent de la 6-connexité, et que les images d'action minimale ainsi que les chemins sont maintenant tridimensionnels. Nous avons brievement présenté l'extension 3D du *Fast-Marching* en section 2.5; plus de détails sur les chemins minimaux dans les images 3D sont disponibles dans [4].

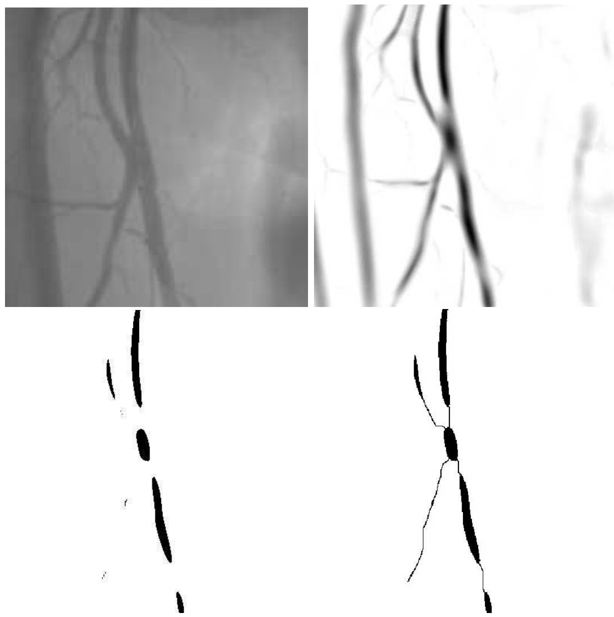

FIGURE 6 – *Image Medicale. Première ligne: image originale et potentiel pré-calculé; Deuxième ligne: à partir d'un ensemble de régions obtenues en seuillant le potentiel, notre métode permet de relier ces régions entre elles par des chemins minimaux vis-à-vis de ce potentiel.*

4.2 Application à un cas réel: un scanner MR de l'aorte

Le problème est dans ce cas de compl´eter un objet partiellement pr´e-d´etect´e. Sur la figure 7, on peut observer une image de r´esonnance magn´etique 3D de l'aorte qui pr´esente une pathologie particulière: un an´evrisme abdominal aortique. L'aorte est visible dans l'image grâce à l'injection d'un produit de contraste avant l'acquisition.
Nous proposons d'extraire à l'aide de notre m´ethode un ensemble de chemins qui donnera une repr´esentation sous forme de squelettes de la structure arborescente de l'aorte.
Notre m´ethode est bas´ee sur une pr´e-d´etection d'un ensemble de r´egions connexes qui appartiennent à l'objet concern´e.
La m´ethode utilis´ee pour extraire une information valable dans le but de construire des chemins est de calculer un filtre multi-´echelle de rehaussement des veines, à partir des travaux de [6] sur ces filtres. Si on extrait les trois valeurs propres de la matrice hessienne calcul´ee à l'´echelle σ, et qu'on les ordonne $|\lambda_1| \leq |\lambda_2| \leq |\lambda_3|$, on d´efinit une fonction de r´ehaussement des structures tubulaires de l'image par

$$\nu(s) = \begin{cases} 0, \text{ si } \lambda_2 \geq 0 \text{ ou } \lambda_3 \geq 0 \\ (1 - \exp \frac{-R_A{}^2}{2\alpha^2}) \exp \frac{-R_B{}^2}{2\beta^2} (1 - \exp \frac{-S^2}{2c^2}) \text{ sinon} \end{cases}$$

avec $R_A = \frac{|\lambda_2|}{|\lambda_3|}$, $R_B = \frac{|\lambda_1|}{\sqrt{|\lambda_2 \lambda_3|}}$, et $S = \sqrt{\lambda_1{}^2 + \lambda_2{}^2 + \lambda_3{}^2}$.
Une explication d´etaill´ee de la m´ethode et de son param´etrage peut être trouv´ee dans [6].

La r´eponse du filtre bas´e sur le Hessien est montr´ee à trois ´echelles diff´erentes $\sigma = 1, 5, 10$ sur la figure 8. La visualisation du volume est faite à l'aide d'une image de projection MIP (Maximum Intensity Projection) ou l'on projette un volume 3D sur un plan 2D en gardant l'intensit´e maximale le long de la direction de projection. En utilisant cette information calcul´ee à plusieurs ´echelles, le potentiel est d´efini comme le maximum de la r´eponse du filtre sur toutes les ´echelles (Fig. 9-gauche). On peut facilement faire un seuillage qui va nous donner un ensemble non structur´e de voxels qui appartiennent à l'aorte, comme sur la figure 9-milieu.
A partir de cet ensemble de r´egions, on applique l'algorithme d´etaill´e dans la section 3, utilisant une version 3D du *Fast-Marching* pr´esent´ee succintement en section 2.5 et plus d´etaill´ee dans [4]. On trouve ainsi l'ensemble des chemins qui relient entre elles les r´egions de d´epart dans notre image, et qui est repr´esent´e sur la figure 9-droite.

5 Conclusion

Nous avons pr´esent´e une nouvelle m´ethode qui permet d'extraire un ensemble de contours dans une image. Cette m´ethode a ´et´e appliqu´ee à un problème de groupement perceptuel pour compl´eter des informations manquantes dans des images de bords. Cette technique est bas´ee sur la recherche de chemins minimaux entre deux points [3]. Notre approche ne n´ecessite pas les points de d´epart et d'arriv´ee pour l'initialisation. Etant donn´e un ensemble non structur´e de r´egions, nous trouvons parmi elles, celles qui doivent être reli´ees par des chemins minimaux. Une fois que les points selles entre les couples de r´egions sont d´etermin´es, les chemins sont calcul´es à partir de chacun de ces points selles en direction de chacune des deux r´egions associ´ees. La somme de ces deux chemins nous donne les chemins minimaux pour chaque couple de r´egions associ´ees. L'ensemble de ces chemins nous permet de compl´eter et fermer l'ensemble initial des contours. Nous avons notamment utilis´e cette m´ethode dans le but de reconstruire des structures vasculaires et nous avons montr´e des exemples sur des images m´edicales de vascularit´es en 2D et 3D.

Références

[1] Laurent D. Cohen. On active contour models and balloons. *CVGIP:IU*, 53(2):211–218, March 1991.

[2] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *IJCV*, 22(1):61–79, 1997.

[3] Laurent D. Cohen and R. Kimmel. Global minimum for active contour models: A minimal path approach. *IJCV*, 24(1):57–78, August 1997.

[4] T. Deschamps and L. D. Cohen. Minimal paths in 3D images and application to virtual endoscopy. In *Proc. ECCV'00*, Dublin, Ireland, July 2000.

[5] E. W. Dijkstra. A note on two problems in connection with graphs. *Numerische Math.*, 1:269–271, 1959.

[6] A. Frangi and W. Niessen, Multiscale Vessel Enhancement Filtering. MICCAI'98, Cambridge.

[7] G. Guy and G. Medioni. Inferring global perceptual contours from local features. *IJCV*, 20(1/2) Oct. 1996.

[8] M. Kass, A. Witkin and D. Terzopoulos. Snakes: Active contour models. *IJCV*, 1(4):321–331, Jan. 1988.

[9] R. Kimmel, A. Amir, and A. Bruckstein. Finding shortest paths on surfaces using level sets propagation. *IEEE* PAMI-17(6):635–640, June 1995.

[10] R. Kimmel, N. Kiryati, and A. M. Bruckstein. Distance maps and weighted distance transforms. *JMIV*, 6:223–233, May 1996.

[11] R. Malladi, J. A. Sethian, and B. C. Vemuri. Shape modeling with front propagation: A level set approach. *IEEE PAMI*, 17(2):158–175, february 1995.

[12] J. A. Sethian. *Level Set Methods: Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision and Materials Sciences*. Cambridge Univ. Press, 1996.

[13] A. Shaashua and S. Ullman. Structural saliency: The detection of globally salient structures using a locally connected network. In *Proc. ICCV'88*, Dec. 1988.

[14] L. R. Williams and D. W. Jacobs. stochastic completion fields: a neural model of illusory contour shape and salience. In *Proc. ICCV'95*, June 1995.

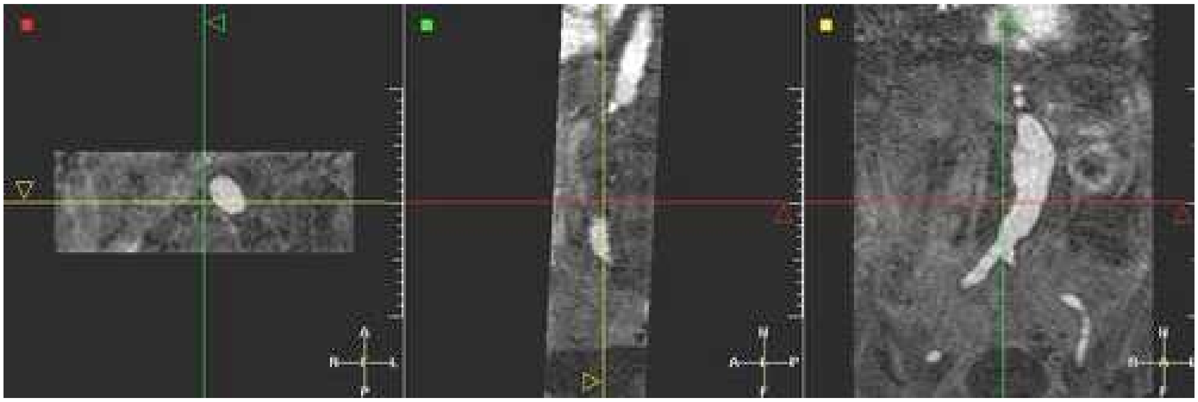

FIGURE 7 – *Trois coupes orthogonales de l'image 3D de l'aorte en resonnance magnétique*

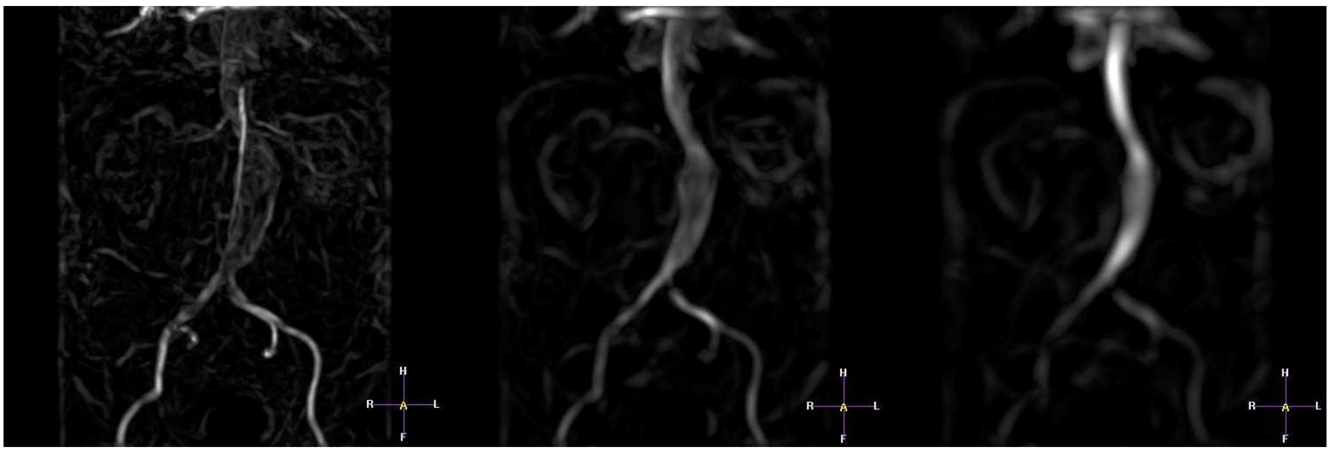

FIGURE 8 – *Détection de tube à trois échelles différentes (*$\sigma = 1, 5, 10$*) (visualisation des images 3D à l'aide de MIP)*

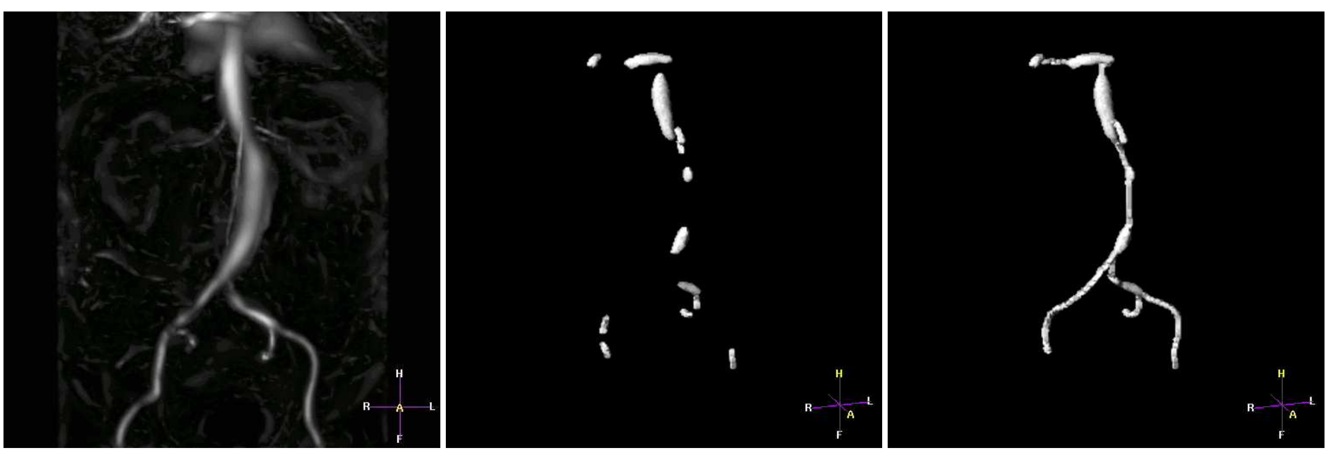

FIGURE 9 – *Groupement perceptuel dans l'aorte de la figure 7: de gauche à droite, le potentiel 3D en MIP; la pré-détection de l'aorte; l'aorte recontruite.*