

Squelettisation par blocs pour des grands volumes de données 3D : application à la microcirculation cérébrale

Skeletonization by blocks for large 3-D datasets: application to the brain microcirculation

C. Fouard^{1,2,5}
F. Cassot²

G. Malandain¹
C. Mazel⁵

S. Prohaska³
D. Asselot⁵

M. Westerhoff⁴
J.P. Marc-Vergnes²

¹ EPIDAURE, INRIA, 2004 route des lucioles, BP 93
06 902 Sophia-Antipolis Cedex, France

⁴ INDEED Visual Concept Berlin, Allemagne

² U455, INSERM Toulouse, France

³ Zuse Institute Berlin (ZIB), Allemagne

⁵ TGS Europe Bordeaux, France

Contact : Celine.Fouard@sophia.inria.fr

Résumé

Représentation compacte permettant une analyse mathématique, le squelette doit être homotope, fin et centré par rapport à l'objet qu'il représente. De nombreuses approches existent pour le calculer, et les auteurs s'attachent en particulier à présenter des algorithmes efficaces. Cependant, dans le cas de grands volumes de données, qu'il n'est pas possible de conserver dans la mémoire d'un ordinateur personnel, ces approches ne sont pas directement applicables. Nous présentons ici, dans le cadre d'une application visant à étudier la microvascularisation du cerveau, un algorithme de squelettisation travaillant localement (sur des sous-images) mais qui garde des propriétés globales (localisation). Nous montrerons les résultats obtenus sur une mosaïque d'images 3D acquises à l'aide d'un microscope confocal.

Mots Clef

Topologie discrète, axe médian, squelette, aminicissement topologique.

Abstract

Skeletons are compact representations that allow mathematical analysis of objects. A skeleton must be homotopic, thin and medial in relation to the object it represents. Numerous approaches already exist. The main focus is computational efficiency. However, when dealing with data too large to be loaded into the main memory of a personal computer, such approaches can no longer be used. We present in this article a skeletonization algorithm that processes the data locally (in sub-images) while preserving global properties (medial localisation). Our privileged application is the study of the cerebral microvascularization, and we show some results obtained on a mosaic of 3-D images acquired

by confocal microscopy.

Keywords

Digital topology, medial axis, skeleton, topological thinning.

1 Introduction

La squelettisation donne une représentation compacte des objets, même dans le cas de formes et de topologies complexes. En dimension 2, les squelettes représentent des lignes inter connectées au centre d'un objet, comme le montre la figure 1. En dimension 3,

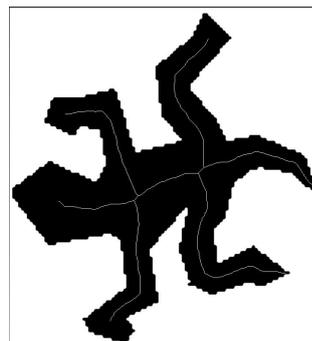


FIG. 1 – squelette d'une image 2D.

les squelettes peuvent représenter des lignes centrales comme en dimension 2 (exemple figure 2) ou bien des surfaces centrales.

Ils sont très utilisés en analyse d'image et reconnaissance de forme, car ils permettent de décrire synthétiquement non seulement la forme, mais aussi certaines propriétés mathématiques des objets, comme par exemple la longueur ou la surface. Ils sont par ailleurs utilisés en compression d'image car ils décrivent les



FIG. 2 – *Squelette linéique de vaisseaux sanguins.*

objets de façon compacte. Beaucoup d’algorithmes ont été proposés pour déterminer le squelette d’un objet [1] (un rapide état de l’art est donné dans la section 3.1), mais tous supposent implicitement que l’image à traiter peut être chargée entièrement dans la mémoire de l’ordinateur.

Cependant, la résolution des images est de plus en plus précise, et dans un même temps, la taille des images à étudier augmente, ce qui produit un accroissement de la quantité de données numériques à étudier. En effet, l’étude des réseaux en imagerie pétrolière ou médicale se déroule sur des données de plus en plus étendues et peut amener à étudier des images de plusieurs gigaoctets (cf section 2). Ces images ne peuvent être chargées en une seule fois dans la mémoire d’un ordinateur *standard*. Aussi, doit-on proposer des algorithmes adaptés pour le traitement de ces images. Si pour un grand nombre d’outils de traitement des images (opérations algébriques, opérateurs simples de morphologie mathématique, filtrage, etc) cela ne pose pas de problèmes particuliers, et où le traitement de l’image entière peut être réalisée en juxtaposant les traitements de sous-images, il n’en est pas de même pour les algorithmes de squelettisation où il faut s’assurer que ce traitement *local* par sous-images conserve les propriétés *globales* attendues du squelette (comme la localisation *au centre* des objets).

La section suivante présente les données avec lesquelles nous travaillons plus particulièrement, ainsi que les traitements que nous appliquons sur ces données. La section 3 présente l’algorithme de squelettisation retenu, qui utilise une carte de distance, puis le calcul de distance, ainsi que la squelettisation, tous deux adaptés à un traitement par sous-images. Enfin nous mon-

trons les résultats obtenus sur des images présentées dans la section 2.

2 Données de micro-vascularisation

Notre application porte sur l’analyse morphologique de la micro-circulation cérébrale qui peut être utile aux anatomistes afin de mieux connaître l’anatomie vasculaire du cerveau, ainsi qu’aux physiologistes afin de mieux appréhender, par le biais de la modélisation, ce qui se passe lors de la mesure du signal de certaines modalités d’images (comme par exemple la Tomographie à Émission de Positons (TEP), ou l’Imagerie à Résonance Magnétique fonctionnelle (IRMf)). Il est pour cela nécessaire d’obtenir des mesures quantitatives qui soient précises mais aussi significatives (par exemple statistiquement). Ces objectifs peuvent être atteints en étudiant des portions relativement étendues (pour la significativité) du cortex avec un microscope confocal (pour la précision).

Dans le cadre de la micro-vascularisation, cette étude morphométrique peut être effectuée en extrayant les lignes centrales des vaisseaux auxquelles on ajoute en chaque point le rayon de vaisseau correspondant (voir figure 4).



FIG. 3 – *Un exemple d’image volumique de microscopie confocale.*

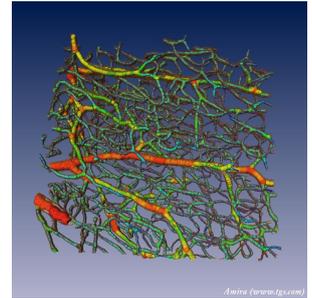


FIG. 4 – *Représentation des lignes centrales des vaisseaux additionnées des rayons correspondants.*

2.1 Acquisition des données

Nos données proviennent de la collection de Duvernoy [2]. Un cerveau a été injecté à l’encre de Chine. Une fois l’encre fixée, la matière cérébrale a été découpée en petites sections montées sur des lamelles dans le but d’observations au microscope conventionnel. Mais ces sections peuvent aussi être observées au microscope confocal à réflexion. La taille moyenne d’une image de microscope confocal est d’environ $600 \times 600 \times 150 \mu\text{m}$ (cf image 3), ce qui est beaucoup plus petit que la taille des sections originales. Afin d’acquérir une section entière, on compose une *mosaïque d’images*: la lamelle est déposée sur une table qui peut se déplacer (en translation) grâce à des vis micrométriques. Par

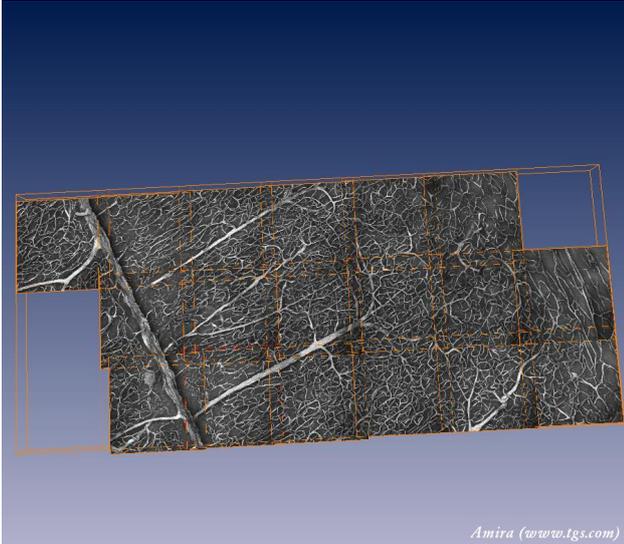


FIG. 5 – Un exemple de mosaïque d'images.

acquisitions et déplacements successifs, on peut acquérir une étendue relativement importante de la section sous la forme d'une mosaïque d'images 3D. Une telle mosaïque peut contenir jusqu'à une centaine d'images, avec pour chaque image $512 \times 512 \times 100$ voxels soit environ 25Mo. On comprend aisément que de telles mosaïques ne puissent être chargées en une seule fois dans la mémoire d'un ordinateur.

2.2 Pré-traitements

Le protocole d'acquisition a été conçu avec un recouvrement assez important (50 voxels) entre deux images successives. En effet, il s'avère que les mesures relevées sur les vis micrométriques sont sujettes à erreur. De plus ces erreurs se cumulent au fur et à mesure de l'acquisition de la mosaïque. De ce fait, si les déplacements relevés sur les vis micrométriques peuvent donner une indication sur les positionnements relatifs des images 3D formant la mosaïque, celle-ci est potentiellement imprécise.

Ce positionnement dans le plan (translations selon X et Y) est donc ré-estimé en optimisant une mesure de similarité (somme des différences au carrés) sur la zone de recouvrement¹. De plus, la table portant la lamelle peut aussi n'être pas parfaitement orthogonale par rapport à l'axe optique du microscope : ceci introduit une translation selon la profondeur (l'axe Z) entre les blocs successifs. Cette dernière translation est corrigée avec le même principe.

La mosaïque obtenue est ensuite binarisée par seuillage après filtrages (filtre médian puis convolution par une gaussienne), et opérations topologiques (suppression

1. Ceci est en pratique réalisé avec les images 2D de projection des intensités maximales, telles que celle présentées dans la figure 5, et en ne recherchant que des translations *entières* en voxels.

des cavités et des petites composantes connexes). La binarisation est pour l'instant réalisée par un simple seuillage, le seuil global pour toute la mosaïque étant choisi par l'utilisateur.

3 Squelettisation des données

3.1 Amincissement ordonné par la distance

Blum fut le premier à introduire la notion de squelette en utilisant la métaphore des feux de prairie [3] : il définit l'axe médian comme le lieu des points où les fronts de propagation s'évanouissent. Calabi et Harnett [4] définissent le squelette comme le lieu des centres des boules maximales d'un objet ; une boule étant définie comme maximale dans un objet s'il n'existe aucune autre boule incluse dans l'objet qui la contient. Kong et Rosenfeld [5] ajoutèrent la notion de préservation de la topologie à la définition du squelette. Par la suite, nous utiliserons la définition suivante, car elle conduit à un squelette ayant de bonnes propriétés pour les analyses mathématiques de l'objet :

DEFINITION 3.1 (Squelette d'un objet)

Un squelette est un sous-ensemble de l'objet ayant les propriétés suivantes :

homotopie : le squelette est topologiquement équivalent à l'objet original, c'est-à-dire qu'il a le même nombre de composantes connexes, le même nombre de trous et de cavités que l'objet original.

minceur : le squelette est topologiquement mince, c'est-à-dire qu'il a un pixel d'épaisseur, sauf aux jonctions pour lesquelles un pixel ne suffit pas à garantir l'homotopie.

localisation : le squelette est situé au centre de l'objet.

La construction du squelette peut se faire dans le domaine continu en utilisant le diagramme de Voronoï. On choisit des points discrets sur le contour continu de l'objet. Le squelette est alors un sous graphe du diagramme de Voronoï de ces points, entièrement contenu dans l'objet (cf figure 6). Cette méthode est basée sur les points du contour de l'image, ce qui représente une quantité de données petite. Le squelette obtenu est connecté et topologiquement équivalent à l'objet car la représentation des objets par leur contour conserve implicitement ces notions, il est de plus centré grâce à la définition du diagramme de Voronoï, et fin car on utilise une représentation continue. Cependant, cette méthode peut poser des problèmes lors des passages entre les domaines continus et discrets. De plus, la complexité des algorithmes et les temps de calculs sont rédhibitoires pour des images gigantesques. Enfin, outre les difficultés liées à la discrétisation des

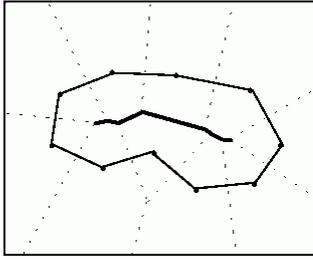


FIG. 6 – Le squelette d'un objet continu est constitué des arêtes de Voronoï qui sont complètement à l'intérieur de l'objet.

contours (échantillonnage, etc), l'extraction d'un squelette à partir d'un diagramme de Voronoï en 3-D n'est pas la simple extension du cas bidimensionnel.

On peut en revanche s'intéresser aux méthodes discrètes, généralement faciles à mettre en oeuvre et rapides. Le processus de squelettisation discret peut être basé sur :

l'amincissement (ou *thinning*): le squelette est obtenu en "épluchant" itérativement les couches frontières de l'objet. Les points *supprimables* (les points simples² et qui ne sont pas des points de fin³) sont enlevés successivement [7] ou en parallèle [8, 9, 10, 11, 12, 13] ou bien encore à l'aide d'opération morphologiques [14]. Ces méthodes conduisent à un squelette *homotope* à l'objet par construction, *mince*, géométriquement représentatif (si les points de fin ont été correctement caractérisés), mais pas nécessairement centré.

les cartes de distance : le squelette est défini comme le lieu des maxima locaux de la carte de distance [4, 15, 16]. Le principe de ces méthodes est de calculer la carte de distance à l'intérieur de l'objet, d'en trouver les maxima locaux, et de reconnecter le squelette ainsi obtenu. Le squelette résultant est *centré* par construction, *mince*, suivant le choix du seuil des maxima locaux, mais pas nécessairement homotope, suivant la méthode de reconstruction.

Des méthodes hybrides, introduites relativement récemment, permettent de bénéficier des avantages de ces deux approches [17, 18]. Ces méthodes consistent en un amincissement homotope, c'est-à-dire en la suppression itérative de *points simples* mais dans l'ordre croissant de la valeur d'une carte de distances calculées

2. Un *point simple* est un point dont la suppression préserve la topologie de l'objet (cf la caractérisation des points simples de [6]).

3. Si l'on enlève itérativement tous les points simples (*shrinking*), on conserve un objet topologiquement équivalent mais géométriquement trop simplifié: une composante connexe sans trous ni cavités se retrouvera réduite à un seul point. On définit alors une condition de *points de fin* qui sont des points qui seront aux extrémités d'une ligne ou au bord d'une surface et qui ne seront pas effacés: ils permettront donc de conserver les lignes ou les surfaces.

à l'intérieur de l'objet (on parlera de *Distance Ordered Homotopic Thinning* ou DOHT), ce qui conduit à un squelette centré.

Par construction, le squelette obtenu est homotope, car les points ne sont enlevés que si cette opération préserve la topologie de l'objet, et mince car tous les points sont détruits jusqu'à ce qu'aucun point supplémentaire ne puisse être supprimé. Afin de préserver au mieux l'invariance par rotation du squelette, nous avons ajouté à cette technique une stratégie directionnelle héritée des méthodes d'amincissement parallèle. Pour chaque distance, on considère tout d'abord les points de bords situés au nord de l'objet, puis les points de bord situés à l'est, puis ceux du bas, puis ceux de l'ouest, ceux du sud et enfin ceux du haut. Le pseudo-code suivant indique le principe de cet algorithme :

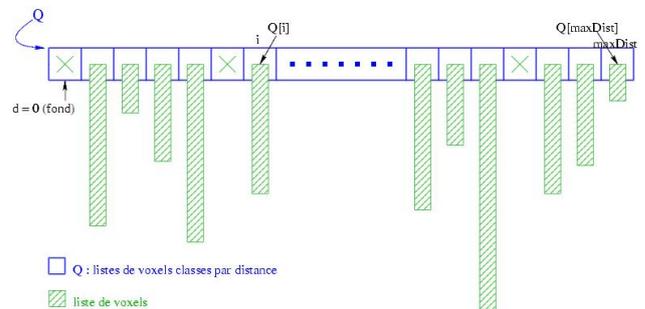


FIG. 7 – Liste des points ordonnés par la distance.

1: Distance Ordered Homotopic Thinning

2: **Entrée:** Image binaire Image[]
% Image d'entrée

3: **Entrée:** Image d'entiers DistMap[]
% Carte de distance de l'image d'entrée

4: **Sortie:** Image binaire Image[]
% Squelette de l'image d'entrée

% Ordonnement des points dans une double liste (figure 7)

5: Q = NouvelleDoubleListe(taille: maxDist)

6: **pour** chaque point p de l'image **faire**

7: **si** (p ∈ Objet) **alors**

8: Insérer(p, Q[DistMap[p]])

% Amincissement ordonné par la distance

9: **pour** dist de 1 à maxDist **faire**

10: **tant que** on détruit des points **faire**

11: **pour** chaque direction **faire**

12: **pour** tout point p ∈ Q[dist] **faire**

13: **si** (Image[p] est sur un bord de la bonne direction) **et**

 (Image[p] est supprimable) **alors**

14: Enlever(p, Q[dist])

3.2 Calcul de carte de distance sur des mosaïques d'images

Comme précisé dans la section précédente, l'algorithme DOHT utilise une carte de distance pour calculer le squelette. Une carte de distance est une image dans laquelle le niveau de gris en chaque point de l'objet correspond à sa distance par rapport à l'extérieur. Il existe de nombreuses façon de calculer une carte de distances (cf [19] pour une revue de ces méthodes). La méthode dite du chanfrein, popularisée par Borgefors [20], réalise un excellent compromis entre la précision et le temps de calcul. Cette méthode est fondée sur la propagation de distances locales entières (cf algorithme de Rosenfeld et Pflatz [21]). Il utilise des masques de chanfrein (qui sont en fait des demi-voisinages) contenant les incréments (ou coefficients) locaux pour la propagation de la distance. La difficulté de cette méthode est de déterminer des incréments entiers optimaux (réalisant la plus petite erreur par rapport à la distance euclidienne). Généralement, ce calcul est conduit pour une grille isotrope. Dans le cas des images biologiques ou médicales, où l'épaisseur de la coupe est en général supérieure à la taille du pixel dans le plan, cela supposerait d'interpoler les données pour les rendre isotropes au prix d'une augmentation de la taille de l'image à traiter. Aussi, des méthodes de calcul des incréments optimaux dans des grilles anisotropes ont été proposées [22, 23] et sont utilisées ici. Par exemple, dans notre cas, la résolution des images est de $1,22 \times 1,22 \times 3 \mu\text{m}$ par voxel. L'algorithme de calcul automatique des coefficients [23] donne les coefficients suivant :

géométrie du masque	coefficient associé
(1,0,0) et (0,1,0)	20
(0,0,1)	49
(1,1,0)	28
(1,0,1) et (0,1,1)	53
(1,1,1)	56

avec un coefficient de division de 22,053.

Le calcul d'une carte de distance utilise en entrée une image binaire où l'extérieur est initialisé à 0 et l'objet à l'infini (en pratique un très grand nombre) pour le calcul de la distance à l'intérieur de l'objet. Son principe est une mise à jour de la distance en chaque point par deux parcours de l'image : un parcours avant (ou *forward*, dans le sens vidéo direct) et un parcours arrière (ou *backward*, dans le sens vidéo inverse). Au cours de chaque parcours, la distance en un point est mise à jour en la comparant à celles des points déjà parcourus additionnées des incréments locaux correspondants (donnés par le masque de chanfrein). Typiquement, lors du parcours direct la mise à jour de la distance en un point $M(i,j,k)$ dépendra des points $N(x,y,z)$ avec $z < k$ ou ($z = k$ et $y < j$) ou ($z = k$ et $y = j$ et $x < i$).

Considérons à présent une image 3D découpée en sous-images ou blocs réguliers. De plus, un parcours avant sur chacun des blocs, eux-mêmes parcourus dans le sens vidéo direct, ne permet pas d'obtenir la dépendance sus-mentionnée pour des points n'appartenant pas au même bloc. En effet, la distance d'un point situé au bord d'un bloc pourra dépendre de points situés dans un autre bloc, et qui plus est, qui n'a pas encore été parcouru. Ceci ne garantit donc pas une bonne propagation des distances.

Afin de propager l'erreur calculée d'un bloc à l'autre, les différents blocs doivent se recouvrir partiellement. La taille de recouvrement entre les blocs doit donc correspondre à la taille du voisinage utilisé par le masque de chanfrein.

Par ailleurs, en ce qui concerne les blocs qui n'ont pas encore été parcourus, le problème peut être résolu par des parcours supplémentaires sur les blocs, par exemple avec la méthode suivante :

parcours avant

- après avoir effectué les parcours avant dans les blocs d'une ligne de blocs (parcourue dans le sens vidéo direct), cette ligne de blocs est re-parcourue dans le sens vidéo indirect et chacun des blocs (sauf le dernier) subit à nouveau un parcours avant.
- après avoir effectué les parcours avant dans les blocs d'un plan de blocs, (parcouru dans le sens vidéo direct), ce plan de blocs est re-parcouru dans le sens vidéo indirect et chacun des blocs (sauf ceux de la dernière ligne) subit à nouveau un parcours avant.

parcours arrière

C'est exactement le pendant du parcours avant.

Ceci conduit à l'algorithme exprimé dans le pseudo-code suivant :

- 1: **Transformation de chanfrein bloc par bloc**
- 2: *%Passage avant sur l'image*
- 3: **pour tout** plan de blocs d'avant en arrière **faire**
- 4: *%Passage avant sur les plans de blocs*
- 5: **pour tout** lignes de bloc de haut en bas **faire**
- 6: *%Passage avant sur la ligne de bloc*
- 7: **pour tout** bloc de la ligne, de gauche à droite **faire**
- 8: Effectuer le passage chanfrein avant sur ce bloc
- 9: *%Passage arrière sur la ligne de blocs*
- 10: **pour tout** bloc de la ligne, sauf le dernier, de droite à gauche **faire**
- 11: Effectuer le passage chanfrein avant sur ce bloc
- 12: *%Passage arrière sur le plan de blocs*
- 13: **pour tout** lignes de blocs, sauf la dernière, de bas en haut **faire**
- 14: *%Passage arrière sur la ligne de blocs*

- 15: **pour tout** bloc de la ligne, de droite à gauche **faire**
- 16: Effectuer le passage chanfrein avant sur ce bloc
- 17: *%Passage arrière sur l'image*
- 18: **pour tout** plan de blocs d'arrière en avant **faire**
- 19: *%Passage arrière sur le plan de bloc*
- 20: **pour tout** lignes de bloc de bas en haut **faire**
- 21: *%Passage arrière sur la ligne de blocs*
- 22: **pour tout** bloc de la ligne de droite à gauche **faire**
- 23: Effectuer le passage chanfrein arrière sur ce bloc
- 24: *%Passage avant sur la ligne de blocs*
- 25: **pour tout** bloc de la ligne, sauf le dernier, de gauche à droite **faire**
- 26: Effectuer le passage chanfrein arrière sur ce bloc
- 27: *%Passage avant sur le plan de blocs*
- 28: **pour tout** lignes de blocs, sauf la première, de haut en bas **faire**
- 29: *%Passage avant sur la ligne de blocs*
- 30: **pour tout** bloc de la ligne de gauche à droite **faire**
- 31: Effectuer le passage chanfrein arrière sur ce bloc

On constate que les passages les derniers blocs des lignes et les dernières lignes des blocs ne sont effectués qu'une seule fois dans le passage forward contrairement aux passages sur les autres blocs qui sont effectués deux fois. On peut donc en déduire une stratégie de découpage des blocs pour le passage forward, en choisissant les derniers blocs des lignes ainsi que les dernières lignes des blocs plus volumineuses que les autres. De même, pour le passage backward.

3.3 Squelettisation bloc par bloc

Dans cette section, nous adaptons l'algorithme de squelettisation précédant aux grandes images, en procédant bloc par bloc. Cette adaptation est guidée par les propriétés que doit impérativement conserver un squelette : l'homotopie, la localisation et la minceur.

Homotopie. Les problèmes d'homotopie se posent au bord des blocs. En effet, lorsque l'on implémente la squelettisation bloc par bloc, on n'a, à chaque instant, qu'une vue partielle de l'image. En particulier les voisinages des points au bord des blocs sont partiellement inconnus. Si l'on considère que cette partie inconnue appartient au fond, les points au bord peuvent être effacés ce qui conduit à une déconnexion des objets (figure 8). Considérer que cette partie inconnue appartient à l'objet ne résout pas non plus ce problème (une composante touchant le bord sera amincie en un point sur le bord mais une déconnexion reste possible).

La seule solution est de *geler* tous les points situés sur les bords des blocs, c'est-à-dire qu'on ne considérera comme "supprimables" que les points dont le voisinage est entièrement inclus dans le bloc. Ainsi, on évitera les déconnexions (figure 9).

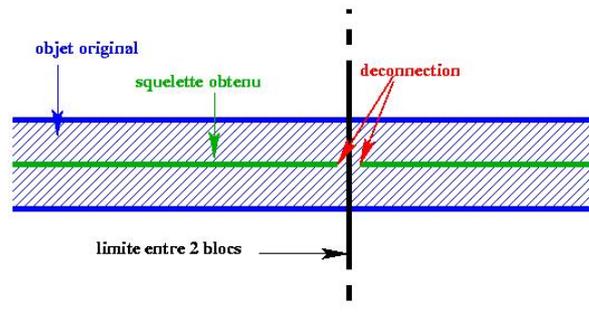


FIG. 8 – Le squelette peut-être déconnecté si l'on ne garde pas les points en bordure de bloc.

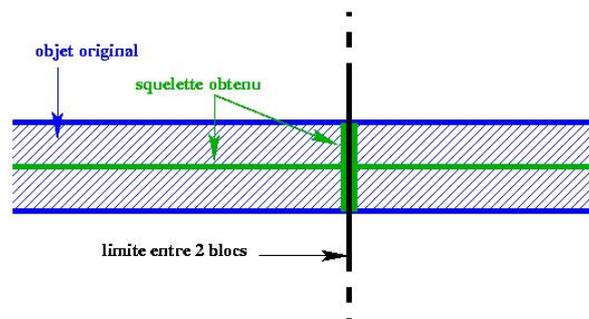


FIG. 9 – En gardant les points de bord, on garde un squelette homotope à l'objet d'origine.

On constate alors une barbule entre les deux blocs. Cet élément sera supprimé si l'on refait une passe sur la partie de l'image concernée. En effet, le bord de ces barbules correspond à des valeurs faibles de la carte de distance, il sera donc éliminé.

Localisation. Cette caractéristique du squelette est plus délicate à conserver. En effet, si l'on supprime dans un bloc tous les points à l'exception de ceux qui sont juste au bord, il peut apparaître un problème de délocalisation du squelette. Comme le montre la figure 10, si l'on procède bloc par bloc, les points *espérés* pour le squelette peuvent être supprimés, et la composante conservée homotope à l'objet sera composée des points *gelés* sur le bord. Le squelette sera donc *collé* au bord du premier bloc aminci, et non pas au centre de l'objet. Pour pallier à ce problème, on ne considère qu'un point est *supprimable* seulement si sa distance par rapport au bord du bloc est plus grande que sa distance par rapport au bord de l'objet. Cela signifie qu'un point ne peut être effacé que si sa boule maximale associée

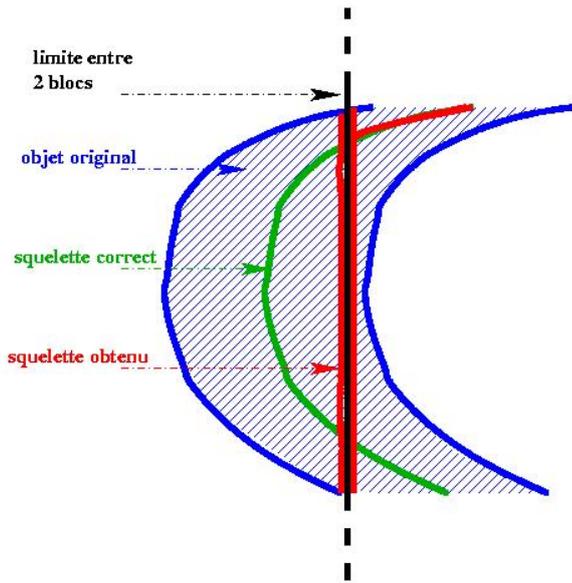


FIG. 10 – le squelette est collé au bord du premier bloc.

est entièrement incluse dans le bloc considéré : il est remarquable que cette condition permet également de *geler* les points au bord des blocs, comme requis pour la conservation de l'homotopie. La figure 11 illustre

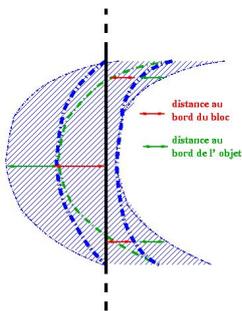


FIG. 11 – Le squelette reste au centre de l'objet.

la partie de l'objet que l'on garde après cette phase de squelettisation. Le squelette correct est entièrement contenu dans cette partie. On constate alors qu'il n'est pas forcément centré par rapport à cette partie. Cependant, si l'on applique une fois de plus une phase de squelettisation sur la région considérée, la suppression des points simples se fera dans l'ordre de la carte de distance de l'objet original. Le squelette correct reste donc situé sur les maxima de cette carte de distance, et l'algorithme de squelettisation détruira par conséquent tous les points situés autour du squelette avant d'accéder aux points du squelette. On obtiendra ainsi un squelette centré par rapport à l'objet original.

Minceur. Les conditions données dans les deux sections précédentes conduisent à un squelette homotope

et centré, mais épais. En effet, les parties de l'objet qui se trouvent au bord des blocs ne sont pas entièrement amincies. Pour obtenir un squelette fin, on ré-applique l'algorithme de squelettisation, avec les mêmes conditions, mais sur les surfaces non amincies, ce qui revient à décaler les blocs de l'image. Les figures 12 à 14 montrent un exemple 2D des différents blocs où la méthode de squelettisation précédente est appliquée :

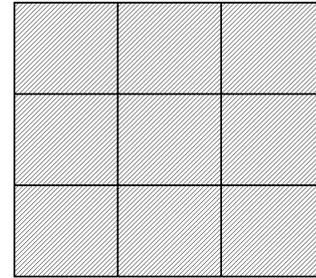


FIG. 12 – Organisation originale des blocs.

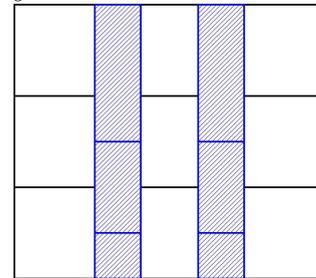


FIG. 13 – Passage sur les frontières de blocs dans la direction x .

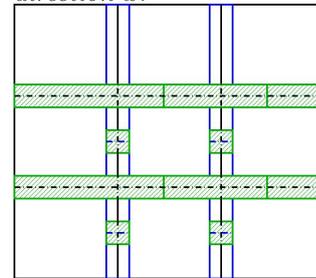


FIG. 14 – Passage sur les frontières de blocs dans la direction y .

Premier passage : on effectue le premier passage sur l'image en partitionnant simplement l'image originale en plusieurs blocs contigus. L'image 12 illustre un tel partitionnement en dimension 2.

Passage sur les frontières de blocs dans la direction x : les zones se trouvant sur la frontière des blocs n'ont pas été entièrement amincies. Pour pallier cela, on effectue à nouveau l'algorithme d'amincissement sur ces zones. On redécoupe donc l'image

en bandes chevauchant les frontières des blocs précédents. On choisit pour largeur de ces bandes 2 fois le plus grand rayon de la carte de distance. Ainsi est-on sûr d'amincir tous les points restants. De plus, on choisit une longueur des bandes légèrement différente de la longueur des blocs initiaux, évitant ainsi les problèmes aux coins des blocs. La figure 13 illustre une telle décomposition de l'image en dimension 2.

Passage sur les frontières de blocs dans la direction

on recommence l'étape précédente pour les frontières des blocs dans la direction y qui n'ont pas encore été traités. Ainsi que sur les frontières dans la direction y qui ont été produites par l'étape précédente. La figure 14 illustre la décomposition de l'image correspondante en dimension 2.

Passage sur les frontières de blocs dans la direction

on recommence l'étape précédente pour les frontières des blocs dans la direction z .

Le squelette obtenu à la fin de ces étapes est *homotope* car on a supprimé uniquement des points dont l'étude du voisinage entier nous garantissait la simplicité, *centré* car on effectue l'amincissement dans l'ordre de la carte de distance, et l'on prend soin de ne pas décaler le squelette sur le bord des blocs, et *mince* grâce à plusieurs passages sur les frontières des blocs, de sorte que chaque partie de l'objet se trouve à un moment ou à un autre situé au centre d'un bloc.

4 Résultats

Nous présentons ici les résultats obtenus sur les mosaïques d'images présentées dans la section 2.

4.1 Lignes centrales des vaisseaux

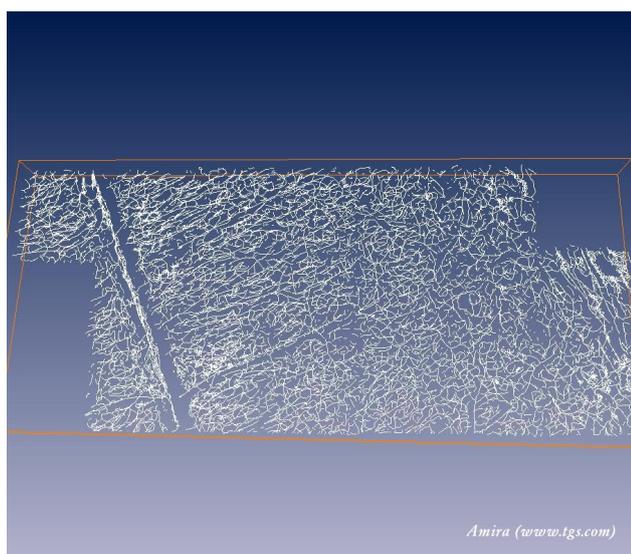


FIG. 15 – Lignes centrales obtenues à partir de la figure 5.

La figure 15 représente le réseau des lignes centrales des vaisseaux que l'on obtient lorsque l'on applique l'algorithme de squelettisation présenté sur la mosaïque d'images de la figure 5. On peut ainsi visualiser les connexions des vaisseaux.

La figure 16 met en évidence les déconnexions que l'on peut obtenir lorsque l'on applique l'algorithme DOHT indépendamment sur chacun des blocs. En effet, cette

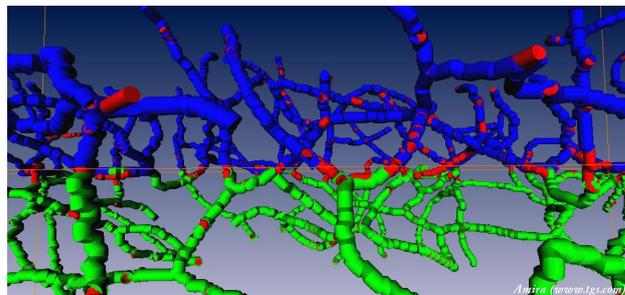


FIG. 16 – Les parties rouges du squelette montrent les déconnexions lorsque l'on ne prend pas de précaution pour la squelettisation bloc par bloc.

figure représente :

- en bleu et vert : les squelettes des images obtenus sur deux blocs voisins traités indépendamment : on traite chaque bloc comme une image à part entière.
- en rouge : le squelette obtenu avec notre méthode : les vaisseaux ne sont pas déconnectés à la frontière entre deux blocs.

4.2 Application

L'analyse morphométrique du réseau micro vasculaire cérébral nécessite par exemple l'étude du rayon moyen des vaisseaux dans différentes zones corticales. Pour calculer un histogramme de ces rayons, on modélise un vaisseau comme une suite de cylindres centrés sur les points des lignes centrales, et dont le rayon correspond à la valeur de la carte de distance en ce point. En effet, le rayon d'un vaisseau correspond à la plus petite distance entre un point central et l'extérieur du vaisseau. La figure 17 montre une telle représentation des vaisseaux sur une partie de la mosaïque.

On peut alors en déduire le rayon moyen des vaisseaux dans cette partie du cerveau, ainsi que la densité du réseau micro-vasculaire ou le nombre de jonctions entre les vaisseaux. La figure 18 montre un histogramme de la distribution des rayons, calculé à partir de la mosaïque, ainsi que le rayon moyen obtenu.

5 Conclusion

Au travers d'une application particulière, l'étude du réseau micro-vasculaire cérébral, nous avons présenté l'adaptation de quelques outils de traitement d'image à

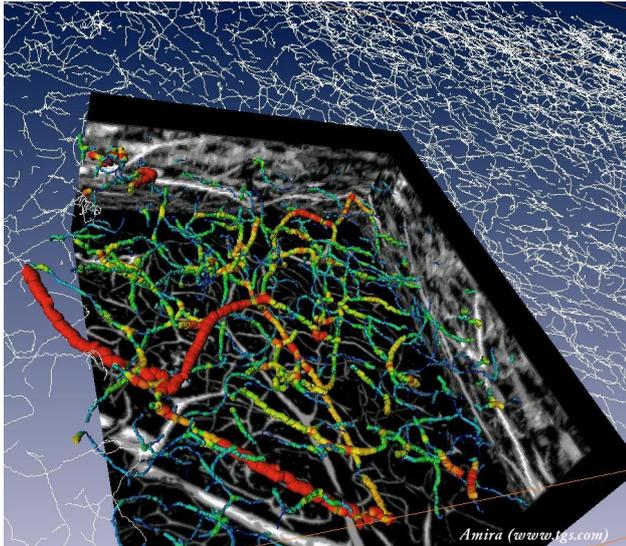


FIG. 17 – Modélisation des vaisseaux.

des grands volumes de données, dont une partie seulement (une sous-image ou bloc) est accessible à un instant donné.

En particulier, nous avons présenté une méthode de squelettisation qui permet de garantir des propriétés locales (homotopie, minceur) et globales (localisation) au sein d'une mise en œuvre par blocs. Outre les caractéristiques *classiques* de telles méthodes (points simples, point de fin, etc), notre méthode, en vérifiant que la boule maximale centrée en un point est incluse dans le bloc à traiter, permet d'éviter les effets de bords (en particulier la délocalisation) inhérents à un traitement par blocs. Dans une première passe, on traite des blocs 3D sans recouvrement (faute de ne pas pouvoir charger l'image entière), puis on traite des sous-images recouvrant les frontières entre ces blocs, la largeur de celles-ci dépendant de la taille des objets à amincir. De ce fait, les zones intérieures des blocs (les blocs sans les frontières) ne sont traités qu'une seule fois. Le surcoût algorithmique lié au traitement par blocs ne se manifeste que sur les frontières.

Nous avons également montré que, dans le cadre de notre application privilégiée, nos outils permettaient d'extraire des informations quantitatives. De telles informations présentent énormément d'intérêt pour des spécialistes (neuro-anatomistes, neuro-physiologistes) pour la description et la compréhension (au moins partielle) du cerveau. Afin de leur rendre accessibles ces outils, ils ont été intégrés dans un logiciel ergonomique. Au cours de cette étude, il est apparu que le point faible de la méthode est le seuillage global de la mosaïque. De ce fait, s'intéresser à des méthodes adaptées de segmentation nous paraît être une continuation naturelle de ces travaux. D'autres techniques d'imagerie (tomographie synchrotron) permettent aussi l'acquisition de grands volumes de données pour l'étude de

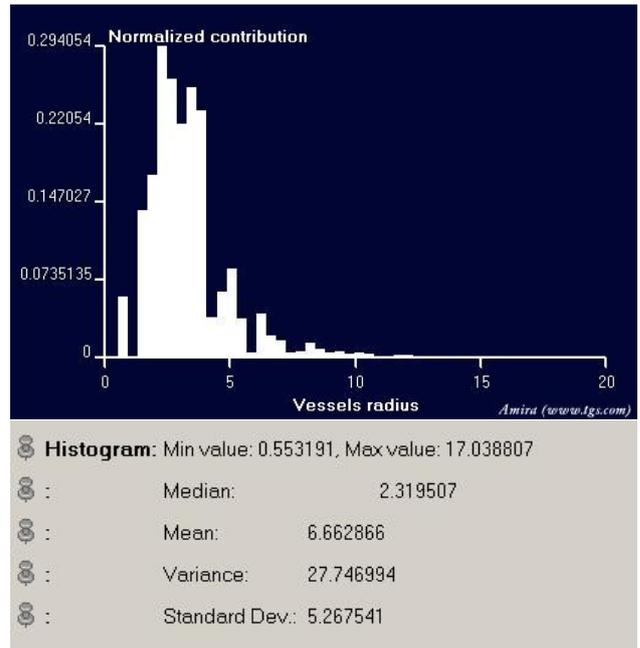


FIG. 18 – Histogramme des rayons du réseau micro vasculaire cérébral (dans la partie observée).

la micro-circulation cérébrale tout en posant d'autres problèmes de segmentation (liés aux défaut de perfusion par exemple).

Remerciements

Ce travail, mené dans le cadre du projet MicroVisu3D, a été partiellement financé par une aide du Fond de la Recherche Technologique. Le travail de C. Fouard est financé par une bourse CIFRE. Le transfert des outils présentés vers les équipes utilisatrices (INSERM U455, etc) a été rendu possible grâce à leur intégration au sein du logiciel Amira [24]. Nous tenons à remercier le Professeur Duvernoy pour avoir mis sa collection à notre disposition, ainsi que messieurs Pablo Cassot et Philippe Cochard pour l'acquisition des images.

Références

- [1] G. Borgefors, I. Nystrom, and G. Sanniti di Baja. Computing skeletons in three dimensions. *Pattern Recognition*, 32(7):1225–1236, July 1999.
- [2] Henri M Duvernoy. Cortical blood vessels of the human brain. *Brain Research Bulletin*, 1981.
- [3] H. Blum. A transformation for extracting new descriptors of shape. *Models for Perception of Speech and Visual Form*, pages 362–380, 1967.
- [4] L. Calabi and W.E. Harnett. Shape recognition, prairie fires, convex deficiencies and skeletons. *American mathematical monthly*, pages 335–342, 1968.

- [5] T.Y. Kong and A. Rosenfeld. Digital topology: Introduction and survey. *Computer Vision, Graphics, and Image Processing*, 48:357–393, 1989.
- [6] G. Bertrand and G. Malandain. A new characterization of three-dimensional simple points. *Pattern Recognition Letters*, 15(2):169–175, February 1994.
- [7] Kálmaán Palágyi, Erich Sorantin, Emese Balogh, and Attila Kuba. A sequential 3D thinning algorithm and its medical applications. In *Information Processing in Medical Imaging (IPMI 2001)*, volume 2082, pages 409–415. Springer-Verlag, June 2001.
- [8] Y.F. Tsao and K.S. Fu. A parallel thinning algorithm for 3-d pictures. *Computer Graphics and Image Processing*, 17:315–331, 1981.
- [9] W. Gong and G. Bertrand. A simple parallel 3D thinning algorithm. In *International Conference on Pattern Recognition*, pages 188–190, 1990.
- [10] Kálmaán Palágyi and Kuba Attila. Directional 3D thinning using 8 subiterations. In *Discrete Geometry for Computer Imagery (DGCI)*, volume 1568, pages 325–336. Springer, 1999.
- [11] Kálmaán Palágyi and Kuba Attila. A parallel 3D 12-subiteration thinning algorithm. *Graphical Models and Image Processing*, 61:199–221, 1999.
- [12] C.M. Ma and S.Y. Wan. Parallel thinning algorithms on 3D (18, 6) binary images. *Computer Vision and Image Understanding*, 80:364–378, 2000.
- [13] C. Lohou and G. Bertrand. A new 3D 6-subiteration thinning algorithm based on p-simple points. In Achille J.-P. Braquelaire, Jacques-Olivier Lachaud, and Anne Vialard, editors, *Discrete Geometry for Computer Imagery (DGCI)*, volume 2301, pages 102–113. Springer, 2002.
- [14] P.P. Jonker. Morphological operations on 3D and 4D images: From shape primitive detection to skeletonization. In Gunilla Borgerfors, Ingela Nyström, and Gabriella Sanniti di Baja, editors, *Discrete Geometry for Computer Imagery (DGCI)*, volume 1953, pages 371–391. Springer, 2000.
- [15] G. Malandain and S. Fernández-Vidal. Euclidean skeletons. *Image and Vision Computing*, 16(5):317–327, April 1998.
- [16] Y. Zhou, A. Kaufman, and Toga A. W. 3D skeleton and centerline generation based on an approximate minimum distance field. *International Journal of Visual Computer*, 14(7):303–314, 1998.
- [17] T. Saito and J.i. Toriwaki. A sequential thinning algorithm for three dimensional digital pictures using the euclidean distance transformation. In *9th Scandinavian Conference on Image Analysis (SCIA '95)*, pages 507–516, Uppsala, Sweden, June 6–9 1995. IAPR.
- [18] C.J. Pudney. Distance-ordered homotopic thinning: A skeletonization algorithm for 3D digital images. *Computer Vision and Image Understanding*, 72(3):404–413, December 1998.
- [19] O. Cuisenaire. *Distance Transformations: Fast Algorithm and Applications to Medical Image Processing*. PhD thesis, Université catholique de Louvain, October 1999.
- [20] G. Borgefors. Distance transformations in arbitrary dimensions. *Computer Vision, Graphics, and Image Processing*, 27:321–345, February 1984.
- [21] A. Rosenfeld and J.L. Pfaltz. Sequential operations in digital picture processing. *Journal of the Association for Computing Machinery*, 13(4):471–494, October 1966.
- [22] Ida-Maria Sintorn and Gunilla Borgefors. Weighted distance transforms for images using elongated voxel grids. In A. Braquelaire, J.-O. Lachaud, and A. Vialard, editors, *Discrete Geometry for Computer Imagery (DGCI 2002)*, volume 2301 of *LNCS*, Bordeaux, France, 2002. Springer.
- [23] C. Fouard and G. Malandain. Systematized calculation of optimal coefficients of 3-D chamfer norms. In *Proceedings of the 11th International Conference on Discrete Geometry for Computer Imagery (DGCI'03)*, LNCS, Naples, Italy, November 2003. Springer.
- [24] Zuse Institute Berlin (ZIB) and Indeed - Visual Concepts, Berlin. *Amira 3.0 - Programmer's Guide*, November 2002. <http://amira.zib.de>.