

Segmentation tridimensionnelle hiérarchique par triangulation de surface

A Hierarchical Approach with Triangulated Surfaces for 3D Data Segmentation

par Jacques-Olivier LACHAUD, Annick MONTANVERT

LIP, ENS-Lyon, URA CNRS 1398
46, allée d'Italie, 69364 LYON Cedex 7
tel : 72.72.84.32 fax : 72.72.80.80
e-mail : {jolachau, montanv}@lip.ens-lyon.fr

résumé et mots clés

Ce travail présente un algorithme de segmentation d'images tridimensionnelles par utilisation de surfaces triangulées et de pyramides. Une triangulation de surface dynamique, dotée d'une modélisation physique et capable de changer sa topologie, va, en se déformant suivant certaines contraintes, segmenter l'image en ses constituants. Afin d'accélérer le processus, un algorithme de construction de pyramide de facteur de réduction quelconque permet de transformer l'image en un ensemble d'images de résolution progressive. Cette hiérarchisation, couplée à un modèle capable d'adapter la précision de sa maille à la résolution de son espace de travail, permet d'estimer très rapidement les formes générales contenues dans une image. Une fois ceci fait, le modèle recherche les détails de plus en plus petits en s'appuyant successivement sur les différents niveaux de la pyramide.

Segmentation tridimensionnelle, modèle déformable, triangulation de surface, topologie variable, pyramide tridimensionnelle.

abstract and key words

This article presents a new algorithm for segmenting three-dimensional images. It is based on a dynamic triangulated surface and on a pyramidal representation. The triangulated surface, which follows a physical modelization and which can as well modify its geometry as its topology, segments images into their components by altering its shape according to internal and external constraints. In order to speed up the whole process, an algorithm of pyramid building with any reduction factor allows us to transform the image into a set of images with progressive resolutions. This organization into a hierarchy, combined with a model that can adapt its mesh refinement to the resolution of the workspace, authorizes a fast estimation of the general forms included in the image. After that, the model searches for finer and finer details while relying successively on the different levels of the pyramid.

Three-dimensional segmentation, deformable model, three-dimensional pyramid, complex topology, triangulated surface, multi-scale.

1. introduction

La segmentation d'images bidimensionnelles cherche à imiter le processus de la vision humaine. L'apparition de données tridimensionnelles dans les domaines médicaux, biologiques ou géologiques a étendu la détection des formes à la reconstruction de volumes complexes, opération difficile à réaliser intuitivement pour l'homme. Ce type de données peut provenir de microscopie confocale, d'IRM ou de tomographie.

L'analyse d'images 3D et la détection des objets qui les composent, posent un grand nombre de problèmes supplémentaires comme la gestion des objets à topologie complexe ou le coût élevé des calculs nécessaires à un espace 3D. Par ailleurs, les moyens d'acquisition introduisent pour l'instant des bruits dans les données. Les segmentations directes [16][20] sont donc souvent insuffisantes et le recours à un modèle déformable est alors indispensable.

Le premier objectif a été d'évaluer l'existant afin de mettre au point un modèle déformable, semi-discret, à topologie variable. Pour dépasser le cadre des segmentations classiques et accélérer

le traitement, notre deuxième objectif a été d'associer à l'image un champ scalaire continu et d'introduire la notion de pyramides d'images tridimensionnelles. Nous avons enfin testé notre modèle et nous avons mesuré le gain fourni par l'utilisation d'une approche hiérarchique.

2. surfaces déformables

2.1. description des modèles déformables

De nombreuses modélisations de surfaces portent le nom de modèle déformable car elles suivent toutes un principe général : il s'agit de déformer un objet (le modèle) jusqu'à ce qu'il minimise une fonction d'énergie. La formulation de la minimisation peut se faire de différentes manières :

– soit directement : la fonction d'énergie est explicite et calculable. Un outil de recherche de minimum, comme la résolution par les moindres carrés, est employé à ce moment-là pour obtenir les paramètres du modèle.

– soit indirectement : on exerce des forces sur le modèle et on attend le point d'équilibre stable. Une fois ce point stable atteint, les paramètres courants minimisent la fonction d'énergie associée au système de contraintes.

Ainsi, dans le cas bidimensionnel, le *snake* (ou contour actif) est une courbe déformable, permettant de segmenter une image par minimisation d'énergie [9] [5]. Afin de s'affranchir d'une seule minimisation globale, il réalise des séries de minimisations locales jusqu'à obtention d'un état global stable. Les moindres carrés sont alors utilisés localement. Le *snake* est généralisable à d'autres dimensions et sert de base à la plupart des modèles tridimensionnels existants. Une extension du *snake* bidimensionnel gérant les changements de topologie peut être trouvée dans [18].

Dans l'espace, la modélisation par quadriques, superquadriques [24] [1] et hyperquadriques [10] est très répandue et permet de segmenter l'image en déterminant à la fois des paramètres globaux et des déformations locales. Dans [24] ils appliquent un ensemble de forces dépendantes de l'image et des propriétés du modèle pour l'orienter vers la solution désirée. Dans [1] ils déterminent dans un premier temps des paramètres globaux par moindres carrés, puis ils affinent le résultat par ajustement d'une boîte de contrôle. Ces deux approches nécessitent néanmoins de travailler sur des objets homologues à une sphère.

La minimisation d'énergie peut aussi être effectuée par une méthode de type éléments finis. Ainsi certaines extensions tridimensionnelles des *snakes* [6] utilisent ces outils de résolution. Là encore, les objets représentés ne peuvent modifier leur topologie.

On peut aussi modéliser le volume à l'aide de fonctions implicites (ou blobs) [26]. Ces blobs actifs fonctionnent similairement aux snakes 2D, dans le sens où il s'agit à chaque fois d'une

minimisation locale des paramètres du modèle. Le principal problème est le coût du processus de résolution d'équations différentielles non linéaires.

Une approche similaire est celle des propagations de fronts [17]. La forme est définie comme étant une fonction implicite de \mathbb{R}^4 dans \mathbb{R} . L'avantage est de pouvoir disposer d'une topologie quelconque sur \mathbb{R}^3 . Par contre, cela se fait au détriment de la complexité de calcul, le recours à une dimension supérieure étant onéreux.

Une dernière approche plus classique, est d'utiliser une maille déformable (mesh) et d'appliquer des forces en ses sommets. La maille peut être une triangulation de surface [19] [11], une spline cubique [15], ou toute autre structure. Dans tous les cas, il faut appliquer des contraintes locales pour conserver une certaine homogénéité au modèle. Le problème est alors de gérer les ruptures de topologie. Pour les splines, [14] apporte une solution partielle à ce problème.

2.2. triangulation de surface adaptative

L'objectif étant de segmenter une image 3D, il apparaît nécessaire de choisir notre modèle en fonction de deux critères essentiels, la rapidité d'obtention et la qualité des résultats.

Après examen des différents modèles, il nous est apparu que la dernière approche, la triangulation de surface, constituait le meilleur compromis. C'est en effet une extension directe du modèle des *snakes* bidimensionnels. Ses avantages sont clairs : simplicité, rapidité ($\mathcal{O}(n^2)$ sommets pour une image de n voxels d'arête), visualisation aisée. Un atout supplémentaire réside dans la facilité d'extraction des caractéristiques de l'objet : calculs de la surface, du volume engendré, de moments d'informations topologiques. Son champ d'application en est donc élargi et couvre des domaines très variés : segmentation, représentation de volumes, CAO, ...

Afin de travailler avec un modèle déformable performant, nous l'avons doté à la fois d'une modélisation physique (voir Section 2.3) et d'une structure capable de conserver sa cohérence au fil des évolutions (voir Section 5).

2.3. modélisation physique

La triangulation (réunion finie de triangles) est considérée comme un système dynamique de particules (les sommets des triangles) interagissant entre elles et avec leur environnement. L'interaction entre particules se traduit en pratique par l'interaction entre une particule et ses proches voisines. Un système dynamique de particules est également modélisé dans [22], mais il ne s'appuie pas sur les liens d'une surface triangulée pour trouver rapidement les particules en réaction.

Nous allons pour notre part utiliser la géométrie de la surface triangulée pour déterminer, pour chaque sommet, les sommets

voisins avec lesquels il va réagir, i.e. ses voisins directs. Ceux-ci permettent de définir deux forces internes : la force de résistance à la courbure \mathcal{F}_c (qui lisse les formes) et la force d'élasticité de la surface \mathcal{F}_e (lorsqu'on tire d'un côté du volume, toute la surface est étirée).

Si S est un sommet de la surface, on note S_g le barycentre des voisins (notés S^i) de S , N_S le nombre de voisins de S , S_n la normale à la surface au point S . De plus d_G désigne la longueur moyenne des arêtes de la surface.

$$\forall S, \vec{\mathcal{F}}_c(S) = \lambda_c \cdot \left[\overrightarrow{SS_g} - \frac{1}{N_S} \sum_{i=0}^{N_S-1} \overrightarrow{SS^i} \right] \quad (1)$$

où λ_c est le coefficient de tension de surface.

$$\forall S, \vec{\mathcal{F}}_e(S) = \sum_{i=0}^{N_S-1} \lambda_e \cdot \left(\|\overrightarrow{SS^i}\| - d_G \right) \cdot \frac{\overrightarrow{SS^i}}{\|\overrightarrow{SS^i}\|} \quad (2)$$

où λ_e est le coefficient de rigidité.

Ces deux forces respectent le principe d'action/réaction. \mathcal{F}_c attire les sommets vers leur plan tangent et représente l'énergie de tension sur la surface; ainsi une forme homologue à une sphère se transformera peu à peu en une sphère si elle est soumise à l'action de cette seule force. \mathcal{F}_e régularise les longueurs des arêtes sur toute la surface et traduit l'énergie de liaison. Ces forces participent à la régularité de la surface triangulée du point de vue de sa forme (force \mathcal{F}_c) et du point de vue du maillage (force \mathcal{F}_e).

Notre but est la segmentation de formes dans une image tridimensionnelle. Afin de traduire l'influence de l'image, notée I et fonction de \mathbb{Z}^3 dans $[0, 1]$, sous forme de contraintes sur notre modèle, nous allons considérer notre espace de travail comme étant un champ scalaire continu de \mathbb{R}^3 . Cela définit une fonction potentiel de \mathbb{R}^3 dans $[0, 1]$ que l'on note Π_I . La transformation de l'image 3D en un champ scalaire continu est décrite dans la Section 3.1.

Cette transformation va nous permettre de définir deux forces externes \mathcal{F}_i et \mathcal{F}_{di} , de fonctionnalités différentes, issues du champ scalaire.

$$\forall S, \vec{\mathcal{F}}_i(S) = \lambda_i \delta \left(\Pi_0 - \Pi_I(\vec{S}) \right) \vec{S}_n \quad (3)$$

où λ_i est le coefficient d'attraction vers une isopotentielle et Π_0 est l'isopotentielle recherchée. Cette force est normalisée par δ (voir Section 5.2), δ représentant l'invariant contrôlant la précision de la triangulation (i.e. la longueur des arêtes).

Notons $(\nabla_i I, \nabla_j I, \nabla_k I)(m, n, p)$ le gradient au point (m, n, p) de l'image discrète I . Si $\Pi_F(\vec{X})$ est l'interpolation d'une image discrète F au point \vec{X} , alors on peut définir un vecteur v_{di} égal à $\Pi_{\nabla_i I}(\vec{S})\vec{i} + \Pi_{\nabla_j I}(\vec{S})\vec{j} + \Pi_{\nabla_k I}(\vec{S})\vec{k}$.

$$\forall S, \vec{\mathcal{F}}_{di}(S) = \delta \left((\lambda_{di} - \rho_{di})(v_{di} \cdot \vec{S}_n) \vec{S}_n + \rho_{di} v_{di} \right) \quad (4)$$

où λ_{di} (resp. ρ_{di}) est le coefficient d'attraction suivant le gradient selon \vec{S}_n (resp. \vec{S}_n^\perp).

– La force \mathcal{F}_i permet de rechercher l'isopotentielle de valeur Π_0 . Son principe est de gonfler ou dégonfler localement la surface tant que celle-ci ne suit pas cette isopotentielle. Il faut donner une valeur positive au coefficient λ_i si le potentiel tend vers 1 à l'infini, négative s'il s'annule.

– La force \mathcal{F}_{di} est plus classique, puisqu'il s'agit d'un simple pistage des gradients. Le coefficient λ_{di} module cette force suivant la normale à la surface, ρ_{di} suivant le plan tangent. Il n'y a donc pas de recherche ou de reconstruction de contours, comme le fait [20], mais seulement une utilisation directe du gradient.

L'algorithme de déplacement de la surface se résume à une itération des pas suivants :

- (i) Calcul des forces internes et des forces déduites de l'image en chacun des sommets,
- (ii) Rééchantillonnage de l'échelle de temps pour borner les déplacements,
- (iii) Application de la Loi Fondamentale de la Dynamique discrétisée en chacun des sommets,
- (iv) Déplacements effectifs des sommets.

On définit le critère d'arrêt de l'algorithme en mesurant l'énergie cinétique correspondant à un déplacement normal à la surface (tout mouvement tangent à la surface n'est qu'un glissement des sommets sur la surface et ne traduit pas un changement réel de forme). Il faut aussi noter que ce processus ne traduit que le déplacement géométrique (i.e. coordonnées des sommets) de la surface et non ses modifications topologiques (i.e. liens entre les sommets).

3. espace de travail image

3.1. transformation en fonction potentielle continue

On note $I(i, j, k)$ notre image discrète à segmenter, où $i = 0 \dots M-1$, $j = 0 \dots N-1$, $k = 0 \dots P-1$. On note aussi sa taille réelle $\mu \times \nu \times \pi$, représentent le volume réellement occupé par l'image discrète dans l'espace.

On détermine alors un potentiel continu $\Pi(x, y, z)$, avec $x \in [0, \mu]$, $y \in [0, \nu]$, $z \in [0, \pi]$, en effectuant l'interpolation trilineaire de $I()$ dans l'espace réel correspondant. $\Pi(x, y, z)$ est donc continu, non dérivable et prend les valeurs de $I()$ aux points $(i*\mu/M, j*\nu/N, k*\pi/P)$ avec $i = 0 \dots M-1$, $j = 0 \dots N-1$, $k = 0 \dots P-1$.

Comme le champ scalaire n'est pas dérivable, le gradient ne peut être directement déduit de l'interpolation. Nous préférons interpoler le gradient discret calculé à partir de l'image 3D discrète.

Malheureusement cette transformation en champ continu provoque une certaine perte d'information sur les contours, et les gradients en ces points sont nettement atténués. Une solution serait d'utiliser une interpolation de degré supérieur (2 ou 3). Par contre, quel que soit le degré d'interpolation choisi, le champ scalaire ainsi obtenu permet d'interpréter l'image comme un empilement d'isopotentielle (similaire aux courbes de niveaux d'une carte topographique par exemple, mais en 3D). Ceci est nécessaire à la cohérence des contraintes issues de l'image définies dans la Section 2.3 et résume ainsi la recherche de formes et de contours à une simple recherche des isopotentielle correspondantes.

3.2. utilisation de pyramides

3.2.1. motivation

L'application telle quelle du processus défini précédemment n'est pas totalement satisfaisante. Le problème est que l'influence du potentiel image est localisée autour des sommets et elle n'a ainsi de sens que si la triangulation est de précision supérieure ou égale à la précision de la discrétisation de l'image 3D.

Deux approches permettent de résoudre ce problème :

- La première est de gérer une surface triangulée de précision comparable à celle de l'image qu'elle segmente. La surface est alors cohérente avec la fréquence du domaine dans lequel elle travaille. Son défaut est de gérer une surface très précise et d'être alors très coûteuse en temps de calcul.

- La seconde est de gérer une surface moins finement triangulée, mais de s'intéresser à une région plus large de l'image autour des sommets. Si on applique ce principe directement, le gain obtenu par un nombre moindre de sommets ne compense pas la perte due au parcours de l'image autour de chaque sommet.

Pour profiter des avantages de chacune des deux solutions, il suffit de calculer une seule fois l'influence à différentes échelles des zones de l'image. Cette solution mixte est fournie par le calcul d'une pyramide d'images 3D, dans laquelle les réductions correspondent à différentes précisions de la triangulation. Le modèle se servira des calculs effectués à des niveaux de précision inférieurs pour initialiser le modèle à la précision supérieure.

3.2.2. pyramides en traitement d'images

Les informations présentes dans une image sont liées à la résolution avec laquelle l'image est considérée. Aussi l'analyse en multi-résolution est devenue courante, car elle permet de structurer le contenu d'une image en le hiérarchisant; on la trouve aussi bien pour traiter des objets que des images.

Les représentations pyramidales d'images, proposées par [23] ont été les premières à définir et exploiter la réduction d'une image. Différents buts peuvent cependant être visés, dont le calcul rapide de paramètres, la compression, la décomposition du signal image, la segmentation, etc [4].

Les pyramides de décomposition en fréquence telles que proposées par [2][3] fournissent des images suivant les résolutions décroissantes qui sont comparables à la perception visuelle d'une image lorsqu'elle est éloignée de l'observateur. L'application d'un noyau gaussien filtre les hautes fréquences de l'image, et permet alors de la sous-échantillonner, fournissant l'image du niveau supérieur. Dans la pratique, un seul opérateur effectue la combinaison filtrage/sous-échantillonnage. Ce processus définit la pyramide gaussienne; le noyau gaussien a pour avantage de ne pas créer de faux contour. De taille 5×5 , il réduit la bande de fréquence d'une octave, d'où une réduction de la fréquence d'échantillonnage du même facteur.

Pour tirer parti des représentations pyramidales, nous avons besoin de les définir pour des images 3D à voxels non cubiques, et en relation avec notre modèle de représentation de surface. Ces points sont abordés dans la Section 4.3.

3.3. problèmes induits

Notre segmentation est donc basée sur une triangulation de surface dynamique travaillant dans un espace multi-résolution. Sa mise en œuvre introduit un certain nombre de problèmes nouveaux :

- Choix de la pyramide. Les pyramides habituelles ne sont pas forcément adaptées au problème de la segmentation volumique. Nous avons développé des pyramides à facteur de réduction quelconque. La Section 4.1 résume ces développements.

- Conservation de l'adéquation constante entre résolution de l'image courante et la précision de la triangulation. Une approche par pyramide introduit des variations de l'échelle de représentation de la structure. La Section 4.2 montre les moyens de la gérer.

- Maintien de cohérence d'une triangulation soumise à des déformations non prévisibles. Il faut alors s'intéresser aux variations de la géométrie de la surface mais aussi à des modifications plus fondamentales, les ruptures de topologie (voir Section 5.4). La Section 5.1 montre quant à elle les outils mathématiques dont on dispose pour contrôler les déformations.

4. pyramide 3D et surface

4.1. pyramides d'images 3D à réduction quelconque

Pour la segmentation 3D, nous avons besoin que notre modèle de surface suive au mieux les zones de discontinuité en enserrant les régions homogènes.

Une gestion en multi-résolution de l'image 3D va nous fournir un accès à l'image allant de l'essentiel au détail (*coarse-to-fine*) qui,

couplé avec l'évolution de notre modèle de surface, va accélérer le processus de segmentation (point très sensible en 3D), et assurer la convergence du processus vers une solution proche d'une solution « visuelle ».

Classiquement, les niveaux successifs d'une pyramide gaussienne sont calculés par application d'un noyau gaussien de 5 pixels de côté. Ceci assure un filtrage à faible coût et sans translation de phase, associé à un facteur de réduction de 2 par dimension de l'image [8].

Soit G_0 l'image initiale de voxels 3D, base de la pyramide, le calcul de G_{h+1} (image de niveau $h+1$ de la pyramide) en fonction de G_h (image de niveau h de la pyramide) est donné par la formule de convolution discrète :

$$G_{h+1} \begin{pmatrix} i' \\ j' \\ k' \end{pmatrix} = \sum_{m,n,p} \omega \begin{pmatrix} m \\ n \\ p \end{pmatrix} \cdot G_h \begin{pmatrix} 2i' + m \\ 2j' + n \\ 2k' + p \end{pmatrix} \quad (5)$$

où ω est un noyau de convolution gaussien de taille 5 et vaut $\left(\frac{1}{16}[1 \ 4 \ 6 \ 4 \ 1]\right)^3$.

Dans notre cas, deux contraintes essentielles sont à prendre en compte :

- les voxels ne sont pas forcément cubiques (les fréquences d'échantillonnage sont fonction du mécanisme d'acquisition de l'image de voxels et ne sont pas forcément identiques – par exemple d'un rapport de 1 à 5 en microscopie confocale);
- le facteur de réduction du sous-échantillonnage doit être adapté à la représentation de la surface.

Aussi la formulation précédente n'est pas directement exploitable.

Rendre l'espace des voxels isotropes pour appliquer de manière cohérente les opérateurs de convolution serait une perte de place (la résolution deviendrait le ppm des fréquences d'échantillonnage). En définissant un espace de travail réel, en correspondance avec la structure discrète de stockage des données initiales, nous allons réaliser les opérations de convolution de manière efficace.

Notre but est de construire une liste d'images discrètes, G_0, \dots, G_{max} , qui représente la pyramide. G_0 est l'image initiale donnée pour segmentation, de taille $M \times N \times P$; c'est l'image qui possède le plus d'informations. G_{max} est l'image qui ne contiendra que les informations de plus basses fréquences. Nous noterons M_h, N_h, P_h , les tailles des images discrètes G_h au niveau h . Soient $I_h = M_h \times N_h \times P_h$, pour $h = 0 \dots max$. Avec ces notations, G_h est une fonction de I_h dans $[0, 1]$.

Nous travaillons dans l'espace défini par l'image réelle, noté I_R , de dimensions $[0, \mu] \times [0, \nu] \times [0, \pi]$. Comme chaque image G_h représente à une échelle différente la même image réelle, chacune de ces images a la même taille réelle μ, ν, π . Le plongement d'un voxel (i, j, k) de G_h dans l'espace réel I_R est donné par la transformation T_h suivante :

$$T_h : \begin{array}{l} I_h \longrightarrow I_R \\ (i, j, k) \longrightarrow \left(i \frac{\mu}{M_h}, j \frac{\nu}{N_h}, k \frac{\pi}{P_h} \right) \end{array} \quad (6)$$

Il s'agit donc d'un plongement conservant les proportions réelles de l'image 3D dans le parallélépipède de l'image réelle.

On appelle unité de l'espace réel, et on note U la valeur $\min(\mu/M, \nu/N, \pi/P)$ (la Section 4.3 explique ce choix). C'est donc la plus petite distance entre les plongements de 2 voxels dans l'image réelle; ce sera l'unité de déplacement dans cette image suivant les 3 directions. Dans le cas d'une image isotrope, on a $U_h = \mu/M_h = \nu/N_h = \pi/P_h$.

Pour construire les niveaux successifs de la pyramide, il faut connaître le facteur de réduction. Ce volume continu nous permet de définir une réduction d'un facteur quelconque, contrairement aux formalisations purement discrètes. Notons cependant que [2][21] abordent le problème de la construction de pyramides discrètes pour des facteurs de réduction rationnels.

La seule contrainte qui reste imposée pour respecter la cohérence de l'opération filtrage/sous-échantillonnage est que le filtrage des hautes fréquences autorise bien la valeur du facteur de réduction. Nous avons choisi de conserver un noyau de convolution 5×5 (en dim. 2), aussi le facteur de réduction par dimension, noté T , doit être inférieur à 2.

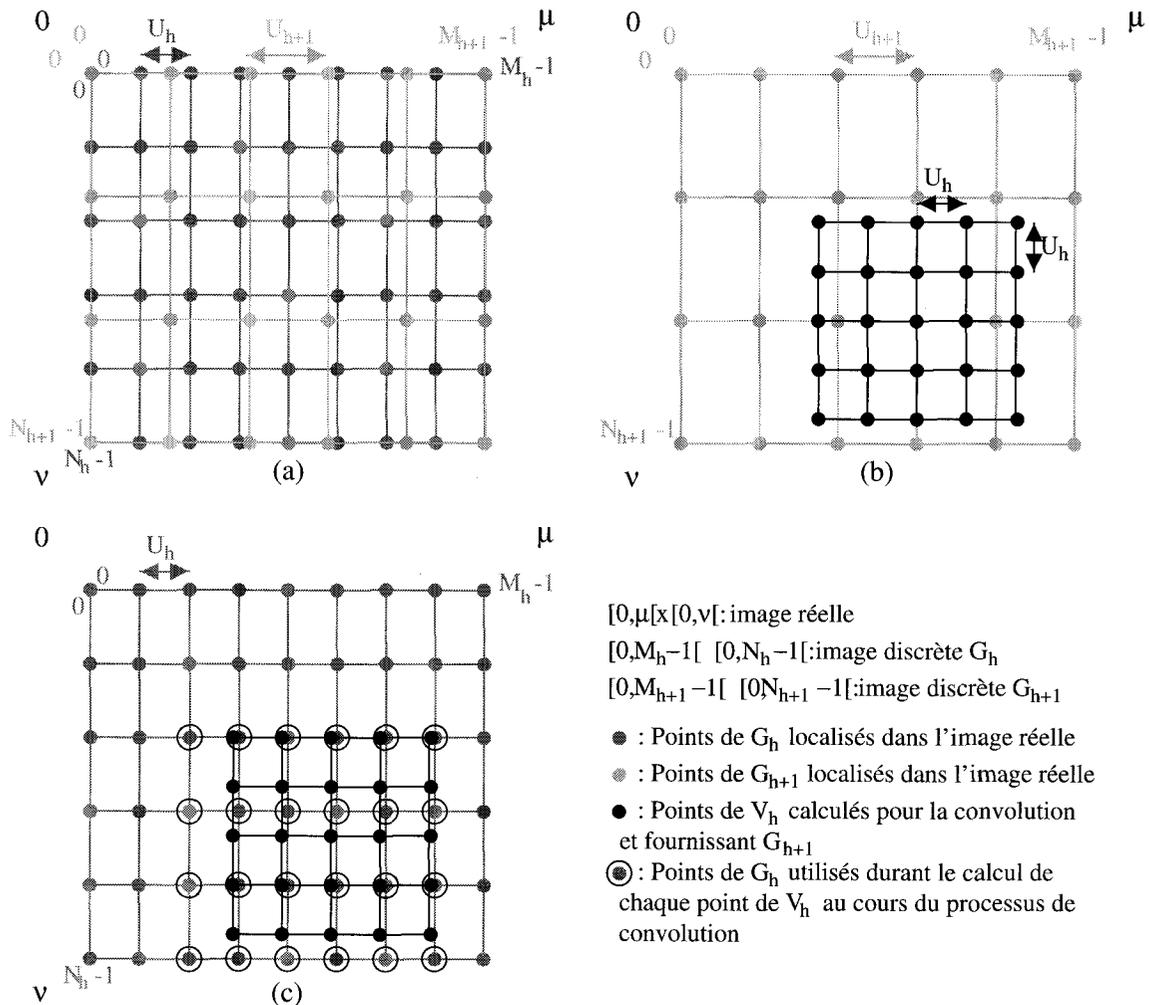
Notons V_0 la base de notre pyramide d'images réelles 3D. V_0 est donnée comme indiqué précédemment par plongement des données discrètes (cf. Section 3.1). Ainsi $V_0 = \Pi_{G_0}$. V_h est le niveau h de cette pyramide. V_{h+1} est calculé à partir de V_h . Le nombre et la localisation des points à calculer sont donnés par le facteur de réduction T , et leurs valeurs vont être obtenues par interpolation puis convolution à partir de points de V_h . Leur stockage après calcul sur l'image réelle se fera dans un tableau de voxels, assimilé à la pyramide discrète (G_i) au niveau $h+1$.

A une image réelle V_h correspondent ses dimensions discrètes M_h, N_h, P_h et son unité de mesure U_h ; ses dimensions réelles restent inchangées (μ, ν, π) . Ces caractéristiques sont définies de manière récursive par :

$$\begin{array}{l} M_0 = M \qquad N_0 = N \\ P_0 = P \qquad U_0 = \min(\mu/M, \nu/N, \pi/P) \\ M_{h+1} = \left\lfloor \frac{M_h}{T} \right\rfloor \qquad N_{h+1} = \left\lfloor \frac{N_h}{T} \right\rfloor \\ P_{h+1} = \left\lfloor \frac{P_h}{T} \right\rfloor \qquad U_{h+1} = U_h \cdot T \end{array} \quad (7)$$

Soit R , de coordonnées (i', j', k') , un point des données discrètes de G_{h+1} ; son plongement $R_{V_{h+1}}$ dans l'image réelle V_{h+1} a pour coordonnées $(i' \cdot \mu/M_{h+1}, j' \cdot \nu/N_{h+1}, k' \cdot \pi/P_{h+1})$ (voir figure 1a).

Pour calculer la valeur de R , l'opération de convolution est définie sur des points de V_h . Le point central a même position dans V_h que dans V_{h+1} . La localisation des 24 autres points est donnée



- [0,μ[x[0,ν[: image réelle
- [0,M_h-1[[0,N_h-1[: image discrète G_h
- [0,M_{h+1}-1[[0,N_{h+1}-1[: image discrète G_{h+1}
- : Points de G_h localisés dans l'image réelle
- : Points de G_{h+1} localisés dans l'image réelle
- : Points de V_h calculés pour la convolution et fournissant G_{h+1}
- ⊙ : Points de G_h utilisés durant le calcul de chaque point de V_h au cours du processus de convolution

Figure 1. – Visualisation du calcul de la convolution : (a) les deux niveaux G_h et G_{h+1} superposés – (b) calcul du niveau G_{h+1} avec la localisation des points du masque de convolution – (c) application du masque sur le niveau G_h et visualisation des points discrets de G_h utilisés pour le calcul.

par utilisation de l'unité de mesure U_h pour discrétiser V_h autour du point $R_{V_{h+1}}$ (voir figure 1b). D'où la formule de convolution :

$$G_{h+1} \begin{pmatrix} i' \\ j' \\ k' \end{pmatrix} = \sum_{m,n,p} \omega \begin{pmatrix} m \\ n \\ p \end{pmatrix} \cdot V_h \left(\mathcal{T}_{h+1} \begin{pmatrix} i' \\ j' \\ k' \end{pmatrix} + \begin{pmatrix} m \\ n \\ p \end{pmatrix} U_h \right) \quad (8)$$

G_{h+1} définit alors V_{h+1} de manière implicite ($V_{h+1} = \Pi_{G_{h+1}}$).

De par le facteur de réduction quelconque, les 25 points sollicités pour la convolution ne coïncideront pas en général avec des points donnés dans G_h (voir figure 1c). De plus il n'y aura en général pas de recouvrement entre les points sollicités pour 2 convolutions voisines.

Aussi chaque point de V_h nécessaire au calcul des convolutions est interpolé à partir des 8 points des données de V_h (donc stockés dans G_h) qui forment le parallélépipède contenant ce point (voir Section 3.1 et aussi la figure 1c).

De par son caractère séparable, le noyau de convolution gaussien 5^3 est appliqué successivement suivant les 3 dimensions. Un bref calcul donne, si T est le facteur de réduction, 5 la taille du noyau de convolution, t' le temps d'accès à la valeur d'un point, t_1 le temps mis par l'algorithme classique et t_2 le temps de l'algorithme optimisé :

$$\begin{aligned} \frac{t_1}{t'} &= 5^3 M_{h+1} N_{h+1} P_{h+1} \\ \frac{t_2}{t'} &= 5 M_{h+1} (N_h P_h + N_{h+1} P_h + N_{h+1} P_{h+1}) \end{aligned}$$

$$\text{D'où } \frac{t_2}{t_1} = \frac{1}{25} (T^2 + T + 1) \quad (9)$$

L'algorithme utilisé est donc plus rapide pour tout facteur T entre 0 et $(-1 + \sqrt{97})/2$.

L'algorithme de construction de la pyramide 3D est donné dans l'Annexe A.

4.2. adéquation avec la surface triangulée

L'interaction surface/image ne se traduisant qu'en termes de contraintes calculées localement autour de chaque sommet, il faut que la finesse de la maille des triangles corresponde avec la résolution de l'image. Les arêtes de notre modèle ne doivent donc être ni trop grandes, pour ne pas manquer des contours de haute fréquence, ni trop petites, car celles-ci ne représenteraient plus alors que des décompositions d'un seul contour sur les voxels.

L'introduction d'un invariant géométrique global sur la surface, noté δ , et décrit précisément dans la Section 5.2, construit les bornes aux longueurs de chaque arête. Cet invariant est notamment utile pour simplifier la résolution des problèmes de cohérence au sein des polyédrisations déformables. Si d_{min} est la plus petite longueur d'arête autorisée et d_{max} est la plus grande longueur d'arête autorisée, alors :

$$d_{min} = \delta \text{ et } d_{max} = 2,5 \delta \quad (10)$$

Si l'on se place dans le cas d'une image 3D isotrope (les résolutions sur les trois axes sont identiques), et si on rappelle que U_h , précédemment défini dans la Section 4.1, est la dimension réelle d'une arête des voxels (cubiques pour une image isotrope) au niveau h de la pyramide, on déduit la relation liant la finesse δ_h de la triangulation à ce niveau de la pyramide et l'unité U_h de l'image. Le cas non-isotrope est résolu dans la Section 4.3.

D'après la figure 2, il faut $d_{min} \leq U_h$ pour qu'une arête puisse représenter les contours formés par deux voxels 6-connexes (distants de U_h) et $d_{max} \geq \sqrt{3} U_h$ pour qu'une arête puisse représenter les contours formés par deux voxels exclusivement 26-connexes (distants de $\sqrt{3} U_h$). On déduit aisément :

$$\frac{\sqrt{3}}{2,5} \leq \frac{U_h}{\delta_h} \leq 1 \text{ avec } U_h = \min\left(\frac{\mu}{M_h}, \frac{\nu}{N_h}, \frac{\pi}{P_h}\right) \quad (11)$$

Ainsi, quel que soit U_h , on peut trouver un invariant δ_h qui satisfasse (11).

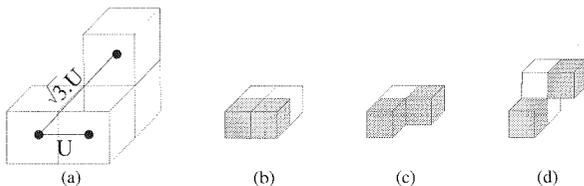


Figure 2. – Adéquation longueur d'une arête et taille d'un voxel : (a) longueurs d'arêtes idéales, (b) exemple de contour 6-connexe, (c) exemple de contour 18-connexe, (d) exemple de contour 26-connexe.

Malheureusement, on ne peut construire une surface de δ donné qu'au moment de son initialisation. Par la suite, la modification de cet invariant est soumise à des contraintes géométriques.

Or, notre surface doit travailler dans une pyramide d'images, et, par ce fait, elle doit pouvoir affiner sa précision à chaque descente

dans la pyramide (voir figure 3). On va donc élaborer un processus capable d'affiner notre triangulation d'un facteur K . Celui-ci nous fournira le facteur de réduction de la pyramide.

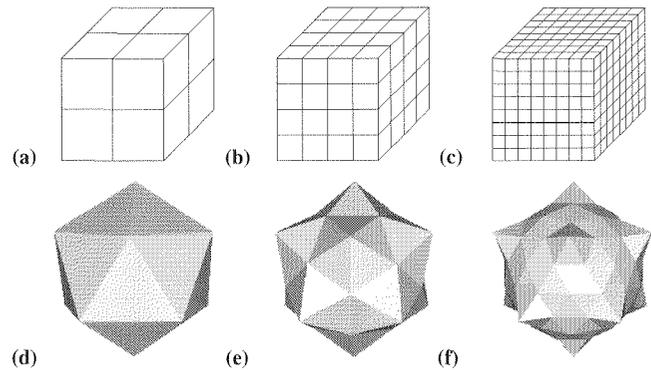


Figure 3. – Résolution de l'image pyramidale et précision de la triangulation : (d) évolue dans l'image (a), (e) dans (b), (f) dans (c).

Processus d'affinement de la triangulation (ou division globale de la surface) (voir figure 4) :

(i) Dans un premier temps, on crée un nouveau sommet au milieu de chaque facette du modèle. Celui-ci est connecté aux trois sommets qui délimitent sa facette.

(ii) Dans un deuxième temps, afin de régulariser les longueurs des arêtes de façon systématique, on remplace chaque arête de l'ancienne triangulation par l'autre diagonale du quadrilatère défini par cette arête.

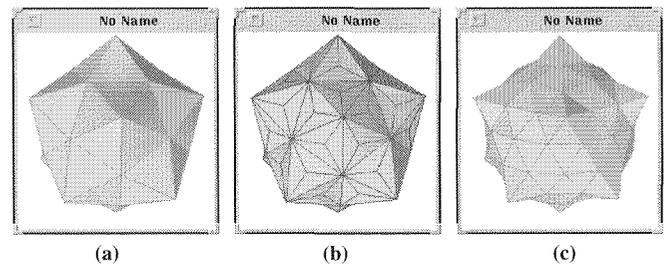


Figure 4. – Exemple de division globale sur un polyèdre à 60 facettes : (a) avant la division globale, (b) après la première phase, (c) après la seconde phase.

Un tel processus réduit la longueur d'arête moyenne à $1/\sqrt{3}$ l'ancienne valeur. Le facteur de réduction à appliquer à l'invariant est donc aussi de $\sqrt{3}$, d'où $K = \sqrt{3}$. Afin que l'inégalité (11) soit respectée à l'initialisation et à chacun des niveaux successifs dans la pyramide, on choisit un facteur de réduction identique pour la pyramide (donc $T = K = \sqrt{3}$) et, pour $h = 0 \dots h_{max} - 1$, on définit :

$$\begin{aligned} \delta_{h_{max}} &= \delta_{init}, \text{ et } \delta_h = \delta_{h+1}/K \\ U_0 &= U, \text{ et } U_{h+1} = U_h T \end{aligned} \quad (12)$$

A l'initialisation, on crée une bulle d'invariant compatible avec (11) au niveau h_{max} . Ensuite, le processus décrit par (12) garantit une bonne adéquation surface/image quels que soient l'itération ou le niveau courant dans la pyramide.

4.3. gestion de l'anisotropie des images

Nous devons gérer différemment la non-isotropie des images suivant l'étape de calcul : construction de la pyramide ou segmentation de l'image à un niveau de la pyramide.

4.3.1. anisotropie lors de la construction de la pyramide

Il est indispensable que le masque de convolution appliqué lors de cette construction soit isotrope dans l'espace réel où est plongée l'image. Si nous n'opérons pas ainsi, la pyramide aura tendance à conserver les contours situés dans une direction où la résolution de l'image est importante et à lisser exagérément ceux situés dans une direction dont la résolution est proportionnellement faible. C'est pourquoi nous avons introduit l'unité U (et U_h pour chaque niveau de la pyramide) dans la Section 4.1.

Le filtre appliqué dans le cas d'une image anisotrope est donc défini à partir de l'unité U_h de valeur $\min(\mu/M_h, \nu/N_h, \pi/P_h)$ et la convolution est réalisée de la même manière que dans la Section 4.1.

4.3.2. anisotropie lors de la segmentation

Lors du processus de segmentation, il faut que la signification des arêtes soit identique pour toutes les directions de l'espace. Si nous nous contentons de travailler dans l'image réelle dotée de ses dimensions (μ, ν, π) , les arêtes perdent leur cohérence par rapport à la précision des informations. D'un autre point de vue, travailler dans une image réelle, où les voxels sont bien cubiques, modifie les contraintes à appliquer; la modélisation physique perd de son réalisme. Nous avons dégagé trois manières d'appréhender ce problème :

(i) La surface se déforme dans l'espace réel de dimensions (μ, ν, π) et suit la modélisation physique. La cohérence triangulation-voxel n'est alors assurée que sur les axes de forte résolution.

(ii) La surface se déforme dans un espace réel obtenu par transformation affine de l'espace (μ, ν, π) . Cet espace a les mêmes proportions que l'image discrète qu'il interpole et est donc de dimensions $(\lambda M_h, \lambda N_h, \lambda P_h)$. Les forces internes ont alors un comportement un peu différent de celui qu'elles auraient dans un espace physique réel.

(iii) La surface se déforme dans l'espace réel de dimensions (μ, ν, π) , que l'on munit d'une métrique non isotrope. Celle-ci est définie à partir de l'image discrète courante (M_h, N_h, P_h) et permet de conserver une cohérence triangulation/voxel (même non cubique) tout en respectant la modélisation physique.

La première méthode donne de bons résultats si l'anisotropie est faible, la deuxième fournit de bons résultats même si l'anisotropie est plus importante. La dernière, plus difficile à mettre en œuvre, notamment pour les opérations sur les vecteurs, est une solution

applicable dans tous les cas. Cependant elle est loin d'être indispensable dans le cadre d'une segmentation, où la modélisation physique n'est pas sensée simuler un comportement très précis et ne constitue qu'un support pour orienter le résultat. Leur mise en œuvre est présentée dans la Section 6.

5. évolution de la surface

Notre triangulation doit représenter des volumes sous forme de surfaces. Or l'espace des surfaces fermées orientables est en bijection avec l'espace des volumes réels. Nous allons donc restreindre notre définition de la triangulation aux surfaces fermées et orientées. De plus, notre surface triangulée étant dynamique, nous devons adapter sa géométrie et sa topologie aux déplacements de ses sommets, afin de conserver sa cohérence et de représenter constamment un volume réel.

5.1. contrôle de la cohérence

Les triangulations de surface, livrées à elles-mêmes, ont tendance à s'auto-intersecter ou à s'entrecroiser. Afin d'éviter ces problèmes, [11] introduit un invariant δ qui limite les tailles minimales et maximales de chaque arête (cf. Section 5.2). Cela permet de gérer simplement les modifications géométriques locales et facilite la détection des collisions entre surfaces. Pour les modifications topologiques, nous nous appuyons sur la caractéristique d'Euler-Poincaré (cf. Section 5.3) pour vérifier la cohérence des changements de topologie.

5.2. rappels sur l'invariant δ

On définit la surface triangulée orientée et fermée τ par :

- (i) τ est une réunion finie de triangles;
- (ii) Soient deux triangles T_i et T_j de τ , $i \neq j$; leur intersection est soit vide, soit réduite à un sommet commun, soit réduite à une arête commune;
- (iii) Toute arête est contenue dans exactement deux triangles distincts.

Notre surface peut donc être composée de plusieurs composantes connexes fermées et orientées.

On introduit alors des contraintes géométriques globales basées sur un invariant δ :

$$\forall (U, V) \text{ sommets voisins, } \delta \leq \left\| \overrightarrow{UV} \right\| \quad (13)$$

$$\forall (U, V) \text{ sommets voisins, } \left\| \overrightarrow{UV} \right\| \leq 2,5 \delta \quad (14)$$

$$\forall (U, V) \text{ sommets non voisins, } \frac{2,5}{\sqrt{3}} \leq \left\| \overrightarrow{UV} \right\| \quad (15)$$

(13) et (14) permettent de travailler avec une triangulation régulière et la préservation de ces deux contraintes sera décrite dans la Section 5.4. (15) détecte les collisions entre deux parties localement distinctes de la surface. Son utilisation dans les changements de topologie est décrite dans la Section 5.4. Les constantes numériques utilisées dans ces deux équations ont été choisies pour garantir l'indépendance des contraintes ainsi définies.

5.3. caractéristique d'Euler-Poincaré

La théorie des surfaces permet de classer les surfaces simplement, en se basant sur des invariants topologiques. Ainsi [7] dégage trois invariants topologiques fondamentaux différents pour une surface \mathcal{S} : le nombre de courbes frontières $\beta(\mathcal{S})$, la caractéristique d'Euler-Poincaré $\chi(\mathcal{S})$, et son orientabilité $q(\mathcal{S})$. L'intérêt est que ces invariants sont indépendants du pavage de \mathcal{S} . En d'autres termes, le degré d'affinement d'une surface n'influe pas sur ses invariants topologiques.

Dans notre cas, on dispose d'informations supplémentaires : notre surface est fermée, donc $\beta(\mathcal{S})$ est nul, et est orientable, donc $q(\mathcal{S})$ est aussi nulle. Toutes nos surfaces triangulées peuvent donc être classifiées suivant ce seul critère $\chi(\mathcal{S})$. Or $\chi(\mathcal{S})$ est simplement calculable pour une surface triangulée. En effet, si $s(\mathcal{S})$ est le nombre de sommets de \mathcal{S} , $a(\mathcal{S})$ est le nombre d'arêtes de \mathcal{S} , $f(\mathcal{S})$ est le nombre de facettes de \mathcal{S} , il est défini par :

$$\forall \mathcal{S}, \chi(\mathcal{S}) = s(\mathcal{S}) - a(\mathcal{S}) + f(\mathcal{S}) \quad (16)$$

Sachant que, pour une surface triangulée fermée, on a la relation $2a(\mathcal{S}) = 3f(\mathcal{S})$; on en déduit :

$$\forall \mathcal{S}, \chi(\mathcal{S}) = s(\mathcal{S}) - \frac{1}{3}a(\mathcal{S}) \quad (17)$$

La figure 5 résume les quatre changements de topologie principaux pour une surface fermée et orientable. On rappelle les variations de $\chi(\mathcal{S})$ engendrées par ces changements. On considère de plus que notre surface \mathcal{S} peut contenir plusieurs composantes connexes. Sa caractéristique $\chi(\mathcal{S})$ est alors la somme des caractéristiques de chacune des composantes connexes de \mathcal{S} . On ne s'intéresse enfin qu'à la variation de la caractéristique de la surface complète.

– Fusion axiale : elle se produit lorsque deux éléments de surface, localement non voisins, entrent en collision. D'après [7], il se produit par construction un découpage de deux morceaux de surfaces qui définissent deux contours C_1 et C_2 , puis un pont est créé entre ces deux contours et les réunit en un seul contour C , enfin ce dernier contour est refermé par un couvercle. Par ce processus, soit un trou est créé dans la surface, soit deux

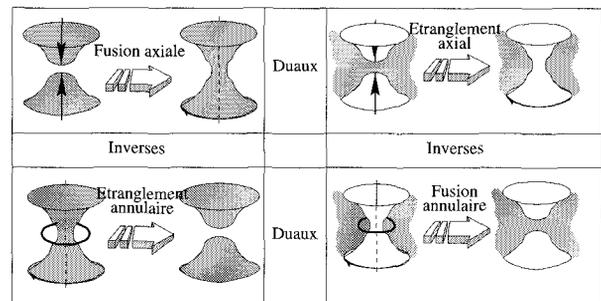


Figure 5. – Tableau récapitulatif des changements de topologie d'une surface fermée, orientée, de \mathbb{R}^3 .

composantes connexes sont réunies en une seule. La surface obtenue \mathcal{S}_1 possède une caractéristique $\chi(\mathcal{S}_1)$ telle que :

$$\chi(\mathcal{S}_1) = \chi(\mathcal{S}) - 2 \quad (18)$$

– Etranglement annulaire : il apparaît lorsqu'une partie de la surface est trop étroite et se replie sur elle-même. La surface est sectionnée à cet endroit précis. On peut modéliser le processus en remarquant que l'on découpe un morceau de surface, créant un contour C , puis on enlève un pont sur ce contour, créant par là-même deux contours C_1 et C_2 , enfin ces deux contours sont refermés par deux couvercles. Un tel processus élimine un trou ou crée une nouvelle composante connexe. La surface obtenue \mathcal{S}_1 possède une caractéristique $\chi(\mathcal{S}_1)$ telle que :

$$\chi(\mathcal{S}_1) = \chi(\mathcal{S}) + 2 \quad (19)$$

– Etranglement axial : il est dual de la fusion axiale (en inversant intérieur et extérieur). La variation de χ est donc identique à (18).

– Fusion annulaire : elle est duale de l'étranglement annulaire (en inversant intérieur et extérieur). La variation de χ est donc identique à (19).

Il existe deux autres changements de topologie, indépendants des quatre précédents : l'apparition de matière et la disparition de matière. La première crée une composante connexe et suit ainsi (19). La deuxième élimine une composante connexe et suit ainsi (18). De tout ce qui précède, il est relativement facile de déduire par récurrence que, si $N(\mathcal{S})$ est le nombre de composantes connexes de \mathcal{S} et $H(\mathcal{S})$ son nombre de trous (topologiques), alors :

$$\forall \mathcal{S}, \chi(\mathcal{S}) = 2(N(\mathcal{S}) - H(\mathcal{S})) \quad (20)$$

5.4. mise en œuvre

La surface triangulée va se déformer sous l'action de différentes forces. Les contraintes définies par l'invariant risquent donc d'être violées. Il faut alors modifier la géométrie de la surface afin d'éliminer les sommets, arêtes et facettes problématiques. Le processus de vérification de ces contraintes est effectué après chaque déplacement des sommets. Les problèmes sont alors

résolus immédiatement. Une description de cette mise en œuvre est dans [12]. Seuls ses apports sont rappelés ici.

Les transformations topologiques respectant les contraintes de la section précédente (notamment les équations (18), (19) et (20)) ont été implémentées (voir [12]). Les bornes imposées à la longueur des arêtes par l'invariant δ autorisent un contrôle rapide et un traitement efficace de ces transformations. Toutes les lois de changement de géométrie ou de topologie de la surface sont cohérentes et transforment notre surface fermée et orientée en une surface de même nature.

Nous initialisons la surface triangulée avec un icosaèdre englobant l'image réelle 3D. On le divise globalement ensuite jusqu'à obtenir une surface respectant nos contraintes géométriques. Ensuite la surface peut évoluer librement.

Les transformations topologiques sont validées sur un exemple synthétique : le « fromage » de Sierpinski, qui est une structure volumique fractale. La figure 6 représente l'évolution de la surface et ses transformations topologiques dans le cadre d'une segmentation classique avec un petit icosaèdre initialisé dans un coin du volume. La figure 7 montre au contraire l'évolution de la surface avec une approche multi-échelle et une initialisation englobant l'image. Les transformations topologiques effectuées sont très différentes même si la surface résultante est de même topologie : 1 composante connexe et 1409 trous. Ainsi, la surface ne segmente que les 5 trous principaux sur l'image de niveau 3 et celle de niveau 2 de la pyramide (car le facteur de réduction est de $\sqrt{3}$ et non de 3 entre chaque niveau). Au niveau 1, la surface détermine les 81 premiers trous du volume. Enfin la surface trouve

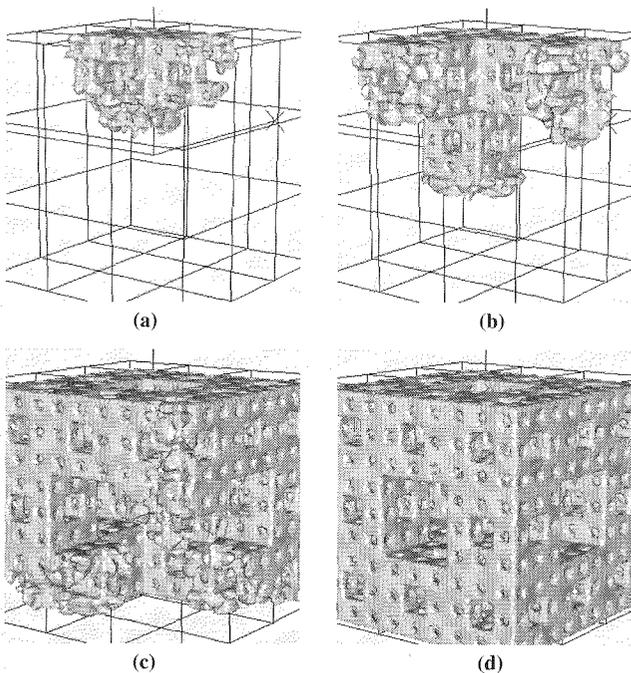


Figure 6. – Evolution de la surface dans une segmentation sans pyramide d'un «fromage» de Sierpinski : (a) itération 140 sur l'image G_0 , (b) itération 240 sur l'image G_0 , (c) itération 440 sur l'image G_0 , (d) itération 640 sur l'image G_0 .

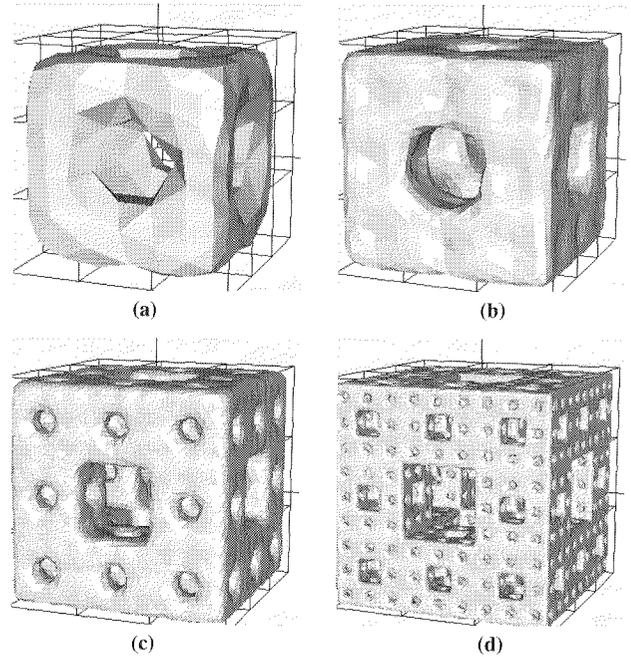


Figure 7. – Evolution de la surface dans une segmentation avec pyramide d'un «fromage» de Sierpinski : (a) itération 100 sur l'image G_3 , (b) itération 175 sur l'image G_2 , (c) itération 275 sur l'image G_1 , (d) itération 525 sur l'image G_0 .

les 1409 premiers trous du volume au niveau 0. De plus, l'image ne contenant qu'une seule composante connexe, le résultat est indépendant à la fois de l'initialisation et de l'approche utilisée. Le gain en temps offert par l'approche pyramidale est quant à lui décrit dans la Section 6 sur des exemples concrets de segmentation.

6. applications

Afin de valider notre méthode, nous l'avons testée sur différents types d'images biomédicales tridimensionnelles : tomographies à rayon X, angiographie IRM, microscopie confocale, coupes congelées, etc. Nous présentons ici deux types d'images qui mettent en évidence les caractéristiques de notre approche.

Nous allons tester notre modèle sur une image volumique de crâne humain, issue d'une tomographie à rayon X¹, de dimensions $256 \times 256 \times 68$ et de taille réelle $1,0 \times 1,0 \times 1,0625$. Cette image est assez fortement anisotrope et nous nous servons de la deuxième méthode de la Section 4.3.2 pour résoudre ce problème. La segmentation se fera avec les paramètres physiques suivants : une tension de surface relativement faible pour obtenir un léger lissage ($\lambda_c = 0,05$), aucune contrainte sur la régularité du maillage ($\lambda_e = 0,0$). La segmentation sera faite par recherche d'une

1. Nos remerciements à Yves Usson (CHU Grenoble – TIMC IMAG) pour la base de données volumique.

isopotentielle ($\lambda_i = -1,0$ et $\Pi_0 = 0,08$). Dans ce cadre nous n'utilisons pas l'autre force d'interaction avec l'image ($\lambda_{di} = 0,0$ et $\rho_{di} = 0,0$). Ces paramètres sont communs à la plupart des images que nous avons testées.

On dote de plus notre processus d'une heuristique qui accélère le traitement des sommets en les parcourant d'autant plus souvent qu'ils se déplacent. On lance deux fois le processus de segmentation :

– D'abord en travaillant directement avec l'image volumique. La figure 8 montre l'évolution de la surface au fil de ses déformations : la surface va lentement coller sur le crâne et pénétrer dedans pour segmenter son intérieur. Il lui faut plus de 400 itérations pour coller parfaitement à l'intérieur du crâne. L'initialisation avec une précision de triangulation adéquate pour cette image étant trop coûteuse en temps, nous avons seulement segmenté l'image de niveau 1 correspondante, ce qui justifie d'autant plus l'utilisation de pyramides comme celles que nous proposons.

– Ensuite en utilisant la pyramide élaborée à partir de cette image volumique avec un facteur de réduction de $\sqrt{3}$. Le processus converge complètement à chaque niveau dans la pyramide avant de descendre d'un autre niveau. La figure 9 représente l'évolution de la surface dans une pyramide d'images. La surface, d'abord grossière, ressemble rapidement à la forme recherchée et s'appuie sur la segmentation d'un niveau pour esquisser l'objet au niveau suivant. A noter que la numérotation des itérations est indépendante du niveau dans la pyramide.

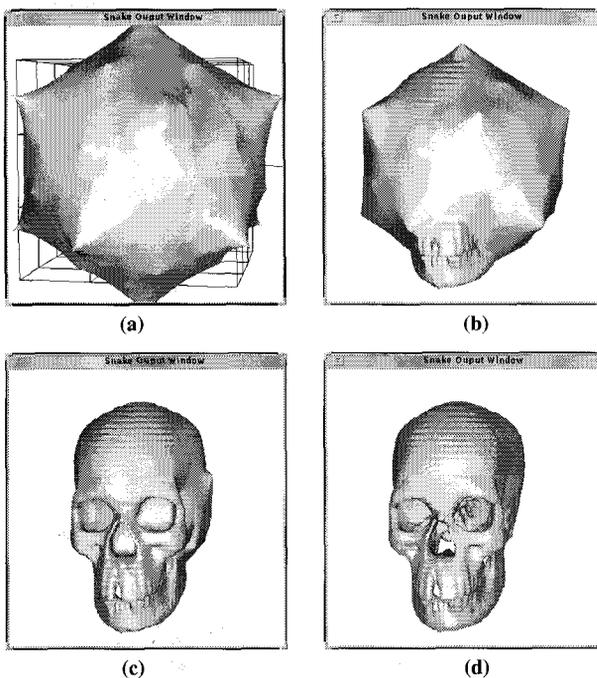


Figure 8. – Evolution de la surface dans une segmentation sans pyramide : (a) itération 0 sur l'image G_1 , (b) itération 100 sur l'image G_1 , (c) itération 250 sur l'image G_1 , (d) itération 700 sur l'image G_1 .

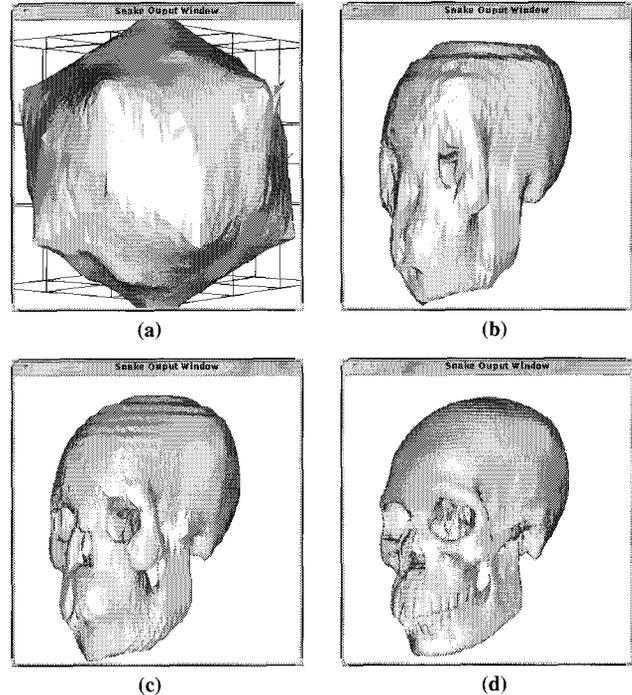


Figure 9. – Evolution de la surface dans une segmentation avec pyramide : (a) itération 0 sur l'image G_3 , (b) itération 225 sur l'image G_3 , (c) itération 333 sur l'image G_2 , (d) itération 461 sur l'image G_1 .

La figure 10 analyse le comportement des deux algorithmes. Le comportement de l'algorithme classique de segmentation est assez simple : la courbe de l'énergie cinétique montre la convergence lente de la segmentation (figure 10b), le nombre de sommets (figure 10d) et la longueur moyenne des arêtes (figure 10c) subissent peu de modifications, le coût en temps (figure 10a) diminue un peu du fait de l'emploi de l'heuristique. Le comportement de l'algorithme de segmentation multi-résolution révèle les 4 niveaux de pyramides utilisés : le nombre de sommets (figure 10d) et la longueur moyenne des arêtes (figure 10c) indiquent que la triangulation a changé de niveau aux itérations 226, 334 et 462. L'évolution de l'énergie cinétique (figure 10b) montre les convergences à chacune des étapes, et le graphe des durées de chaque itération (figure 10a) explique la vitesse d'obtention du résultat. Notons que les irrégularités sur les courbes de coût en temps sont dues à la résolution des ruptures de topologie qui, pour l'instant, nécessitent un autre parcours complet de la surface pour examiner à nouveau les contraintes géométriques.

Afin de mesurer la robustesse de l'approche pyramidale, nous avons testé notre modèle sur une angiographie IRM à contraste de phase². Ce type d'image présente la particularité de ne posséder que peu d'informations de basses fréquences, les données représentant surtout des vaisseaux d'irrigation du cerveau. Les premières images de la pyramide sont donc peu significatives et la surface n'obtient qu'une initialisation éloignée de la forme finale. Malgré cela, comme le montrent les images de

2. Nos remerciements à l'UMDS Image Processing Group, London, pour l'image angiographique.

Segmentation tridimensionnelle hiérarchique

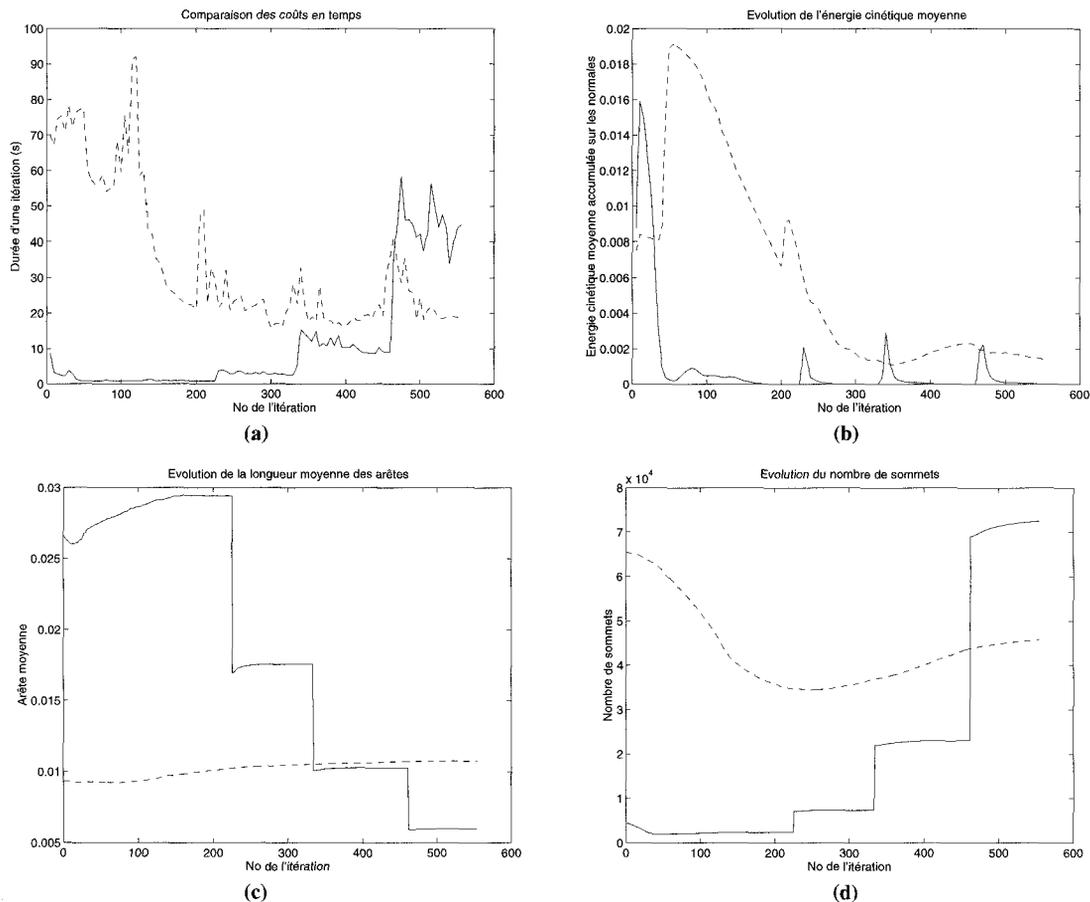


Figure 10. – Statistiques sur la segmentation d'un crâne et comparaison entre l'approche sans pyramide (en tirets) et l'approche avec pyramide (en continu) : (a) durée de chaque itération, (b) évolution de la moyenne de l'énergie cinétique normale à la surface, (c) longueur d'arête moyenne à chaque itération, (d) nombre de sommets de la surface à chaque itération.

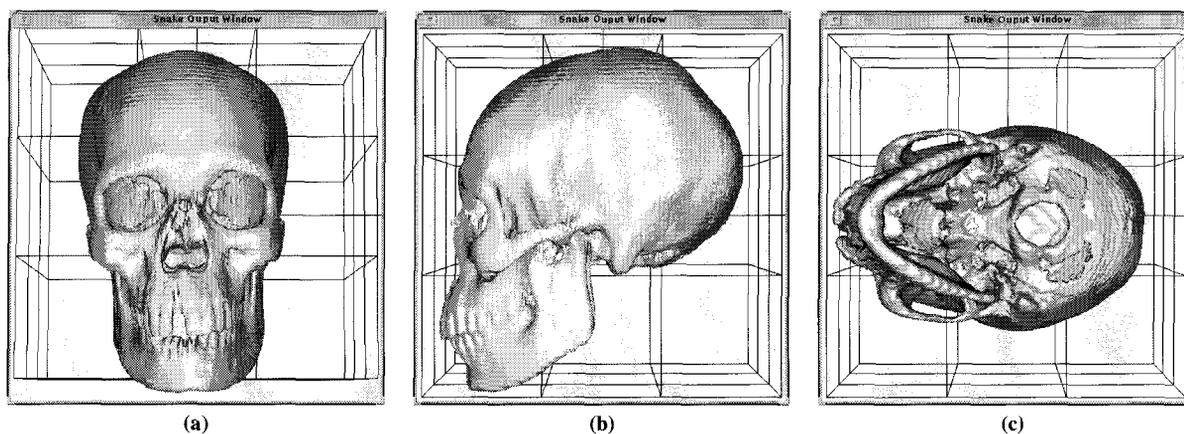


Figure 11. – Résultats finaux de la surface pour la segmentation du crâne (82473 sommets, 247551 arêtes, 165034 facettes) : (a) vue de face, (b) vue de côté, (c) vue de dessous.

la figure 12, la surface parvient à remonter le long des vaisseaux pour reconstituer le système d'irrigation du cerveau de manière aussi précise qu'une approche non-hiérarchique. Bien entendu, sur ce type d'image, le gain en temps offert par l'approche pyra-

midale est beaucoup moins significatif. Une étude préalable sur l'information fréquentielle d'une image 3D est donc essentielle pour juger de l'opportunité de l'utilisation d'une approche multi-résolution.

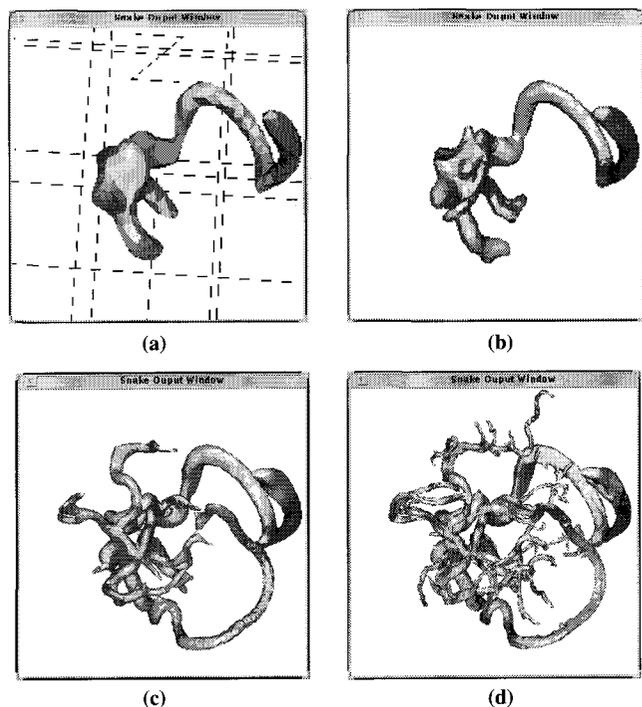


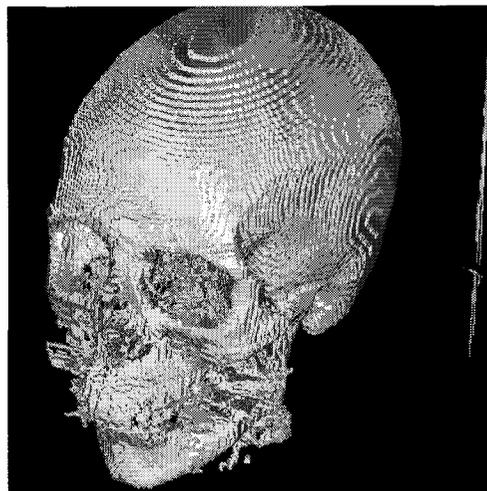
Figure 12. – Evolution de la surface lors de la segmentation d'une angiographie avec utilisation d'une pyramide : (a) après convergence sur l'image G_3 , (b) après convergence sur l'image G_2 , après convergence sur l'image G_1 , (d) résultat final.

7. conclusion

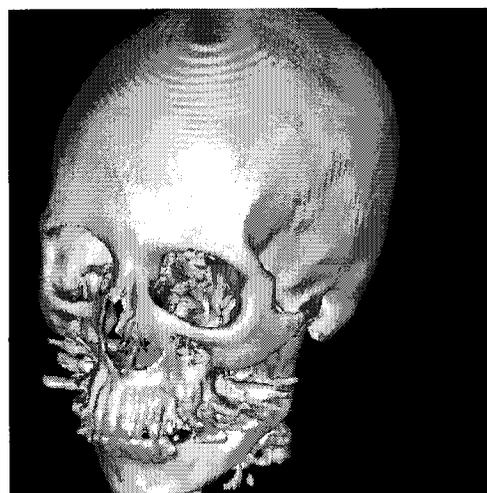
Nous avons développé un algorithme de segmentation performant au moyen d'une triangulation de surface déformable évoluant dans une pyramide d'images tridimensionnelles. Les résultats obtenus montrent que leur qualité est identique à celle d'autres algorithmes; les tests démontrent quant à eux l'efficacité et la rapidité de notre méthode.

Nous avons de plus testé l'utilisation de notre approche dans une chaîne de traitement d'images : notre application récupère le résultat d'un algorithme Marching-Cubes [16]³ et élimine les nombreux bruits et artefacts produits par ce type de segmentation « directe ». Ainsi la figure 13a montre la surface issue d'une version étendue du Marching-Cubes [13], qui garantit notamment la fermeture et l'orientabilité de l'isosurface générée. La figure 13b montre cette surface déformée après quelques itérations par notre modèle, grâce à l'introduction de contraintes internes. Avant déformation, la surface possédait 354 composantes connexes et 958 trous; après déformation et stabilisation, la surface est composée de 45 composantes connexes et 181 trous. L'introduction de contraintes internes a permis ainsi l'élimination d'artefacts peu stables du point de vue physique, i.e. des bruits peu significatifs.

3. Nos remerciements à Jean-Marc Nicod et Serge Miguet (LIP - ENS Lyon) pour l'implémentation de cet algorithme.



(a)



(b)

Figure 13. – Enchaînement d'un processus de type Marching-Cubes avec notre modèle : (a) après Marching-Cubes, (b) après déformations sous l'action de contraintes internes.

Quelques points peuvent cependant être améliorés :

- On peut guider la convergence par le biais de nouvelles contraintes, par exemple en introduisant des forces attractives générées soit par des points « sûrs » de l'objet, soit par des arêtes détectées par un pré-traitement. Ainsi nous travaillons sur l'utilisation d'ondelettes pour détecter les forts contrastes et les singularités des images ce qui autoriserait une vitesse de convergence accrue ainsi qu'une meilleure détection des zones instables sur la surface.
- Ensuite, afin d'élargir le champ d'application de notre modèle, on peut compléter la formulation physique du modèle en lui ajoutant des paramètres physiques globaux (vitesse globale, vecteur rotation instantanée, ...) [24]. On conserve bien entendu les composantes locales pour représenter les déformations.
- Enfin, pour améliorer la vitesse générale de l'évolution, on peut accélérer la détection des changements de topologie en utilisant

des algorithmes optimaux comme celui présenté dans [25]. Une hiérarchisation plus contraignante des sommets permettrait ainsi de réduire le coût de la détection des auto-intersections (actuellement en $\mathcal{O}(n \log(n))$ ou n est le nombre de sommets).

Ces différentes extensions proposées constituent autant de domaines à explorer.

A construction de la pyramide tridimensionnelle

Soit I l'image de travail de dimensions $M \times N \times P$ et de taille réelle $\mu \times \nu \times \pi$. On note T le facteur de réduction et T_{max} la réduction maximale demandée.

- $h \leftarrow 0, G_0 \leftarrow I, M_0 \leftarrow M, N_0 \leftarrow N, P_0 \leftarrow P$.
- Tant que $T^h < T_{max}$ faire,
 - calcul de $M_{h+1}, N_{h+1}, P_{h+1}$.
 - convolution de V_h , déduit de G_h , suivant le premier axe par le filtre $[1 \ 4 \ 6 \ 4 \ 1]/16$ et le résultat est placé dans G'_h de dimensions $M_{h+1} \times N_h \times P_h$.
 - convolution de V'_h , déduit de G'_h , suivant le deuxième axe par le filtre $[1 \ 4 \ 6 \ 4 \ 1]/16$ et le résultat est placé dans G''_h de dimensions $M_{h+1} \times N_{h+1} \times P_h$.
 - convolution de V''_h , déduit de G''_h , suivant le troisième axe par le filtre $[1 \ 4 \ 6 \ 4 \ 1]/16$ et le résultat est placé dans G_{h+1} de dimensions $M_{h+1} \times N_{h+1} \times P_{h+1}$.
 - libération de la mémoire occupée par G'_h et G''_h ,
 - $h \leftarrow h + 1$,
- Fin Tant que.

BIBLIOGRAPHIE

- [1] E. Bardinet and L. D. Cohen and N. Ayache, "Fitting 3D Data Using Superquadrics and Free-Form Deformations", In *12th International Conference on Pattern Recognition*, Volume 1, pages 79-83, IAPR, October 1994.
- [2] P. J. Burt, "Fast filter transforms for image processing", *Computer Graphics and Image Processing*, 16 : 20-51, January 1981.
- [3] P. J. Burt and E. H. Adelson, "The Laplacian pyramid as a compact image code", *IEEE Transactions on Communications*, 31 : 532-540, April 1983.
- [4] A. Chéhikian, "Algorithmes optimaux pour la génération de pyramides passes-bas et laplaciennes", *Traitement du Signal*, 9 : 297-308, January 1992.
- [5] L.D. Cohen, "Note on active contour models and balloons", *CVGIP*, 53 (2) : 211-218, March 1991.
- [6] L.D. Chen and I. Cohen, "Finite Element Methods for Active Contour Models and Balloons for 2D and 3D Images", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11), November 1993.
- [7] H.B. Griffiths, "Surfaces", *Cambridge University Press, January, 1976*.
- [8] J. M. Jolion and A. Rosenfeld, "A pyramid framework for early vision", Kluwer, January 1994.
- [9] M. Kass and A. Witkin and D. Terzopoulos, "Snakes : active contour models". In *1st Conference on Computer Vision*, Londres, June 1987.
- [10] S. Kumar and D. Goldgof, "A Robust Technique for the Estimation of the Deformable Hyperquadrics from Images". In *12th International Conference on Pattern Recognition*, volume 1, pages 74-78. IAPR, october 1994.
- [11] J-O. Lachaud and E. Bainville, "A discrete adaptative model following topological modifications of volumes". In *4th Discrete Geometry for Computer Imagery*, 183-194, Septembre 1994.
- [12] J-O. Lachaud and A. Montanvert, "Volumic Segmentation using Hierarchical Representation and Triangulated Surface". Research report 95-37, LIP - ENS Lyon, France, November 1995.
- [13] J.O. Lachaud, "Topologically Defined Isosurfaces". Research Report 96-20, LIP-ENS Lyon, France, 1996.
- [14] F. Leitner and P. Cinquin, "Complex topology 3D objects segmentation", In *Advances in Intelligent Robotics Systems*, volume 1609 of *SPIE*, Boston, November 1991.
- [15] F. Leitner and I. Marquand and P. Cinquin, "Dynamic Segmentation : finding the edge with snake-splines". In *Curves and Surfaces*, pages 279-284. Academic Press, January 1990.
- [16] W. E. Lorensen and H. E. Cline, "Marching Cubes : A High Resolution 3D Surface Construction Algorithm", *Computer Graphics*, 21 : 163-169, January 1987.
- [17] R. Malladi and J. A. Sethian and B. C. Vemuri, "Shape Modelling with Front Propagation : A Level Set Approach", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(2) : 158-174, February 1995.
- [18] T. Mc Inerney and D. Terzopoulos, "Medical Image Segmentation Using Topologically Adaptable Snakes". In *Computer Vision, Virtual Reality and Robotics in Medicine*, pages 92-101, Nice, 1995. Springer-Verlag.
- [19] J.V. Miller and D.E. Breen and W.E. Lorensen and R.M. O'Barnes and M.J. Wozny, "Geometrically deformed models : A method for extracting closed geometric models from volume data". *Computer Graphics*, 25(4), July 1991.
- [20] O. Monga and S. Benayoun, "Using differential geometry in R4 to extract typical features in 3D density images". In *11th International Conference on Pattern Recognition*, volume 1, pages 379-382. IAPR, September 1992.
- [21] S. Peleg and O. Federbusch, "Custom Made Pyramids". *Pyramidal Systems for Computer Vision*, F 25, January 1986.
- [22] R. Szeliski and D. Tonnesen, "Surface Modeling with oriented Particle Systems". Technical Report CRL-91-14, DEC Cambridge Research Lab., December 1991.
- [23] S. Tanimoto and T. Pavlidis, "A hierarchical data structure for picture processing". *Computer Graphics and Image Processing*, 4 : 104-119, June 1975.
- [24] D. Terzopoulos and A. Witkin, "Deformable Models : Physically based models with rigid and deformable components", *IEEE Computer Graphics and Applications*, 8(6) : 41-51, November 1988.
- [25] P. Volino and N. Magnenat-Thalmann, "Efficient self-collision detection on smoothly discretized surface animations using geometrical shape regularity". In *Eurographics '94*, volume 13(3), September 1994.
- [26] R.T. Whitaker, "Volumetric deformable models : active blobs". in *VBC*, volume 2359 of *SPIE*, pages 122-134, march 1994.

Manuscrit reçu le 10 Juillet 1996.

LES AUTEURS

Jacques-Olivier LACHAUD



Jacques-Olivier LACHAUD est Ingénieur ENSIMAG. Il prépare un Doctorat en Informatique au LIP (Laboratoire de l'Informatique du Parallélisme) de l'Ecole Normale Supérieure de Lyon. Ses travaux de recherche portent sur les représentations de surfaces et de volumes par des modèles dynamiques, pour la segmentation et le suivi d'objets en imagerie 3D. Il travaille sur les propriétés topologiques des surfaces, les méthodes par triangulation, et à l'exploitation de ses résultats sur des données biomédicales.

Annick MONTANVERT



Annick MONTANVERT est Professeur d'Informatique à l'Ecole Normale Supérieure de Lyon, et effectue ses recherches au LIP (Laboratoire de l'Informatique du Parallélisme). Ses travaux portent sur l'analyse d'images, avec une attention particulière pour la nature discrète des données. Les domaines étudiés en imagerie 2D et 3D sont la représentation de formes, la segmentation d'images, les approches hiérarchiques et le codage d'images.