

Un modèle volumique déformable hiérarchique et un calcul de distance k-d efficace pour la segmentation d'images 3-d

Eric Bittar, Stéphane Lavallée

GMCAO - TIMC / IMAG Institut A. Bonniot, Université Joseph Fourier,
Faculté de Médecine de Grenoble, 38 706 La Tronche, France.
Eric.Bittar@imag.fr, Stephane.Lavallee@imag.fr

RÉSUMÉ

Nous présentons un modèle volumique déformable qui permet la segmentation automatique d'images médicales tridimensionnelles. Notre travail combine un modèle octree-spline déformable hiérarchique et régularisé, avec le calcul rapide d'une distance généralisée prenant en compte des caractéristiques différentielles extraites automatiquement. Nous montrons des résultats de déformation d'objets de synthèse (ellipsoïdes), et des tests de segmentation d'images TDM 3D (SCANNER X) de vertèbres à l'aide d'un modèle.

ABSTRACT

We present a volumetric deformable model for automatic segmentation of three dimensional medical images. We combine a hierarchical deformable octree-spline model with a fast calculation of a generalized distance which uses differential features automatically extracted on the images. The results we show include deformation of synthetic objects (ellipsoids) and segmentation tests of 3D CT-SCAN images of a vertebra with a deformable model.

1 Introduction

L'extraction d'un objet dans des images médicales 3D est une opération difficile. L'utilisation d'un modèle déformable permet d'inférer un signal dans les zones les plus bruitées. Dans la suite de l'apparition des courbes puis des surfaces splines, sont apparus des modèles déformables de toutes sortes qui ont apporté des fonctionnalités supplémentaires aux modèles existants. Les principales nouveautés sont la mémoire de forme, la déformation volumique et hiérarchique, et l'utilisation de différents types d'information pour guider la déformation.

Nous décrivons dans cet article un modèle qui combine ces aspects. La déformation, qui est obtenue par un octree-spline [18] est volumique et hiérarchique. Elle est définie sur un modèle initial qui constitue la forme de référence que l'on cherche, par hypothèse à préserver. La minimisation d'une distance généralisée entre les éléments du modèle et ceux de l'image de données permet de prendre en compte les caractéristiques différentielles des images pour le recalage. Nous détaillons dans la section 4 la méthode de calcul du point le plus proche que nous avons adoptée pour calculer cette distance. Nous présentons ensuite en section 5 des expérimentations sur des objets de synthèse et des résultats de segmentation d'images TDM 3D par un modèle déformable de vertèbre.

2 Bibliographie

De nouvelles méthodes de segmentation d'images tridimensionnelles sont apparues récemment. Elles utilisent des déformations volumiques et non plus surfaciques, et peuvent se classer selon deux critères : la représentation de la transfor-

mation et les caractéristiques que l'on extrait sur les images 3D pour les recalcer. L'introduction de points possédant des caractéristiques dans la boucle de traitement nécessite un algorithme efficace de calcul d'une distance généralisée. Nous présenterons donc également les méthodes de ce domaine.

2.1 Représentations de la transformation

Un vaste éventail de transformations sont utilisées pour positionner et déformer un modèle. On note d'abord les transformations globales, affines (à 12 degrés de liberté), quadratiques, ou d'ordre supérieur [16, 9]. Viennent ensuite les transformations locales. Feldmar [7] propose d'utiliser des transformations affines par morceaux. D'autres transformations sont construites sur des grilles régulières de vecteurs. La transformation est en général calculée en multi-résolution [1, 3]. La transformation peut aussi être calculée sur une base spline. De nombreux auteurs utilisent les splines plaques-minces de Duchon pour leur optimalité [2, 6, 10]. Nous avons choisi d'utiliser les grilles hiérarchiques de type octree-spline [18].

2.2 Caractéristiques

La mesure qui guide le recalage peut, elle aussi, être de nature variée. Certains auteurs appartiennent simplement des points qui sont désignés interactivement, ou bien extraits semi-automatiquement ou automatiquement. En considérant que les points forment un ensemble dans chacune des deux images, d'autres auteurs utilisent la distance à cet ensemble [18]. D'autres auteurs encore calculent des structures géométriques caractéristiques ayant de bonnes propriétés de stabilité pour réduire la dimension des objets à recalcer [17]. Certains auteurs se réfèrent à la théorie de l'entropie et utilisent l'information

mutuelle [10, 16] pour un recalage multimodalité. Bajcsy utilise la corrélation [1]. Le gradient et la courbure sont également utilisés. Ils peuvent être combinés, sous la forme du produit de la norme du gradient par la courbure moyenne [5], ou pour former une distance généralisée [7].

2.3 Calcul de distance

Un moyen de calculer rapidement la distance d'un point à un ensemble de points de dimension k (que l'on appelle modèle), est de précalculer cette distance pour un certain échantillonnage de l'espace.

Plusieurs solutions ont été présentées en 3D. Mangin utilise une approximation à l'aide d'une distance du chanfrein adaptative [13]. Brunie, Lavallée et Szeliski [4, 11] créent un octree-spline adapté au modèle. Ils précalculent la distance au modèle en chaque sommet des cubes de l'octree. Ensuite la distance d'un point quelconque au modèle est approximée par l'interpolation linéaire des distances des 8 sommets du cube auquel appartient le point. Cette structure est très efficace en terme de calcul de distance, puisque le coût ne dépend pas du nombre de points du modèle.

Ces approches se généralisent mal pour des dimensions plus importantes. Une distance discrète comme un octree nécessitent un nombre de distances précalculées et stockées qui dépendent exponentiellement de la dimension.

D'autres types d'approches, évaluées dans [12] peuvent s'utiliser pour des dimensions supérieures à 3. Elles sont basées sur la recherche du point le plus proche dans le modèle.

3 Présentation de la méthode

Nous considérons deux images 3D : un volume de données et un volume modèle. De ces images sont extraits deux ensembles de points k-d, le modèle et les données. Le modèle une fois constitué servira à segmenter de multiples jeux de données. Les caractéristiques sont extraites des deux volumes par un filtre de Canny-Deriche-Monga [14]. Les points de contour de l'objet auquel nous nous intéressons sont définis de manière semi-automatique une fois pour toutes dans le volume modèle. Le modèle est formé par l'ensemble de ces points, associés de leurs caractéristiques. Les données sont constituées de points du volume de données et de leurs caractéristiques. Les points extrémaux dans la direction de leur gradient sont sélectionnés, et seuillés à l'aide d'un seuillage par hystérésis pour ne garder que les points de gradient suffisamment fort. Notre objectif est de réaliser la segmentation du volume de données en déformant le modèle pour le mettre en correspondance avec les données.

3.1 La déformation par Octree-Spline

Comme dans [18], nous utilisons un ensemble de déformations qui ont un nombre de degrés de liberté croissant. La première est la transformation rigide, puis nous appliquons des déformations polynômiales globales, avant d'utiliser des déformations splines locales. La déformation est régularisée en utilisant une combinaison linéaire de stabilisateurs d'ordre 0 et 1.

3.2 Minimisation d'une énergie

La déformation doit minimiser une fonction de coût. Cette fonction comprend un terme correspondant à la somme des carrés des distances généralisées des points de donnée au modèle, et un terme de régularisation.

Nous utilisons un algorithme itératif de minimisation dont le coeur est la méthode de Levenberg-Marquardt, combinée avec une descente de gradients conjugués préconditionnés qui opère sur une représentation de l'octree-spline dans des bases hiérarchiques. Cet algorithme utilise les dérivées des distances par rapport aux paramètres de la déformation.

4 La distance généralisée

Pour chaque point de donnée, il faut calculer à chaque itération la distance k-d au modèle. C'est la distance au point le plus proche parmi les n points du modèle. A chaque point de coordonnées x, y, z sont associées ses caractéristiques différentielles pour limiter les faux appariements. Les caractéristiques que nous avons retenues sont la direction et la norme du gradient de l'image, que nous regroupons dans les 3 composantes g_x, g_y, g_z du gradient. Au total, chaque point est représenté par $k = 6$ composantes, x, y, z, g_x, g_y, g_z .

4.1 Les arbres k-d

Un arbre k-d est une généralisation de l'arbre binaire, dont la construction est en $O(n \log n)$ opérations. La taille d'un arbre k-d est en $O(n)$. La complexité moyenne de la recherche du point le plus proche dans l'arbre k-d pour un point est en $O(\log n)$.

Détaillons la construction d'un arbre k-d pour un ensemble de n points $x^i, i = 1..n$ de dimension k . Les points sont triés suivant chacune de leurs coordonnées, et la racine est la médiane des points, suivant le premier axe. Les fils de la racine seront les médianes respectives des deux sous ensembles de points selon le deuxième axe. Et ainsi de suite chaque noeud non terminal divise l'espace en deux demi-espaces par un hyperplan qui est orthogonal à un des k axes de coordonnée.

Un noeud de l'arbre est donc représenté par l'indice d'un point et une dimension de l'espace. L'arbre est construit en considérant les k dimensions de manière circulaire.

4.2 Recherche dans un arbre k-d

Les premiers algorithmes de recherche [15] dans un arbre k-d utilisaient une boîte de taille fixe pour réduire l'espace de recherche par une méthode de type aiguiller-et-borner (branch-and-bound). Nous utilisons une amélioration proposée par Friedman [8] dans laquelle la dimension de la boîte se réduit avec les points trouvés au cours du parcours.

Voici la procédure de recherche du point le plus proche du point P dans un noeud v d'un arbre k-d. d_{min} est la distance au point le plus proche P_{pp} trouvé jusqu'à présent. dv est la dimension du noeud v . Pv est le point par lequel passe le noeud v . $P[dv]$ est la coordonnée d de P . $Pv[dv]$ est la coordonnée dv de Pv . fg est le fils gauche de v . fd est le fils droit de v .

On remarque qu'un premier test qui est une simple comparaison unidimensionnelle permet d'éviter le coûteux calcul de la distance k-d si les composantes de Pv et P sur l'axe dv sont éloignées de plus de $dmin$.

Si ensuite la distance k-d entre P et Pv est inférieure à $dmin$, c'est que Pv est le plus proche des points que l'on a trouvé pour le moment.

```

procedure cherche_noeud(v, P, Ppp, dmin)
  si (|Pv[dv]-P[dv]| <= dmin) {
    si (distance_kd(P,Pv) < dmin){
      Ppp =Pv
      dmin=distance_kd(P,Pv)
    }
  }
  si ((P[dv]-dmin<Pv[dv]) et que fg est non vide)
    cherche_noeud(fg,p,Ppp,dmin)
  si ((Pv[dv]-dmin<P[dv]) et que fd est non vide)
    cherche_noeud(fd,p,Ppp,dmin)
}

```

Comme le parcours de l'arbre se fait en profondeur d'abord, on commence en fait par l'exploration du fils dans le demi-espace duquel se trouve le point P . En effet une mise à jour de la valeur de $dmin$ lors de l'exploration de cette première branche peut éviter l'exploration de l'autre demi-espace.

4.3 Recherche itérée dans un arbre k-d

Comme notre algorithme de déformation est itératif, nous devons calculer à chaque itération le point le plus proche dans le modèle de chacun des points de donnée. Nous utilisons la distance au précédent point le plus proche $dmin = distance_{kd}(P, Ppp_{Prec})$ pour borner initialement la recherche dans l'arbre k-d. L'algorithme ainsi initialisé garantit de trouver le point le plus proche.

Nous faisons en plus un deuxième choix, qui permet d'accélérer la recherche mais ne permet plus de garantir l'obtention exacte du point le plus proche. Si d'une itération à l'autre un point de donnée P ne s'est pas trop déplacé, on peut considérer que son point le plus proche reste le même. On ne recalcule réellement le point le plus proche que si la distance entre P , et sa position précédente à laquelle on a effectivement calculé le point le plus proche dépasse un certain seuil.

4.4 Distance fine

Une autre distance est utilisée pour obtenir une convergence plus fine dans un état proche du minimum. Il s'agit de la distance $3D$ au plan passant par le point le plus proche, et dont la normale est le vecteur gradient en ce point. En utilisant cette distance, les points de donnée ne sont plus appariés avec des points du modèle, mais avec la surface du modèle, approximée par les points et leurs normales. Ceci confère plus de régularité à la distance, au voisinage de la surface.

5 Résultats expérimentaux

La première série de figures montre un recalage de deux ellipsoïdes de synthèse. La figure 1 présente les deux nuages après recalage rigide. On remarque (figure 2) qu'avec une distance $3D$, la convergence n'est pas complète, alors qu'avec

notre distance $6D$, les deux nuages se recouvrent bien (figure 3).

Les figures suivantes montrent une segmentation de vertèbre. Le nuage modèle est représenté par 26368 points, il a été segmenté à la main sur une image scanner. Les points de données ont été extraits automatiquement sur une autre image scanner comportant 40 coupes 256×256 , générant 4005 points. Ils sont représentés par un segment de droite qui part d'un point de donnée et qui aboutit sur le point du modèle le plus proche. La figure 4 présente les points après le recalage rigide et la figure 5 montre le résultat après convergence, qui a pris 6min34 sur un Pentium 133.

6 Conclusion

Nous avons présenté une méthode de segmentation 3-d consistant à inférer un modèle sur l'image traitée par déformation volumique de ce modèle. Cette approche volumique est a priori plus robuste que les méthodes de surfaces déformables. Son implémentation utilise un maillage de déformation adaptatif, hiérarchique et régularisé, ainsi qu'une minimisation de distances généralisées entre des points du modèle munis de caractéristiques différentielles et des images données munis de ces mêmes caractéristiques. Nous avons présenté une méthode efficace pour le calcul de ces distances en utilisant des arbres k-d. Nous montrons des résultats sur données de synthèse et des images médicales de type scanner.

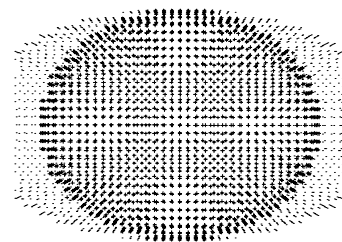


FIG. 1 — Position après recalage rigide

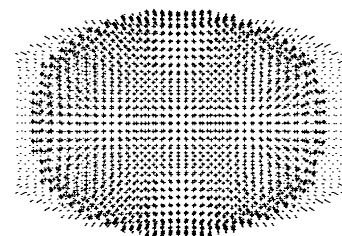


FIG. 2 — Avec une distance 3D

Références

- [1] R. Bajcsy and S. Kovacic. Multiresolution elastic matching. *Computer Vision, Graphics, and Image Processing*, 46 :1–21, 1989.
- [2] F.L. Bookstein and W.D.K. Green. Edge information at landmarks in medical images. In *SPIE vol. 1808*

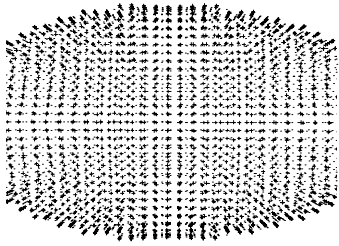


FIG. 3 — Avec une distance 6D

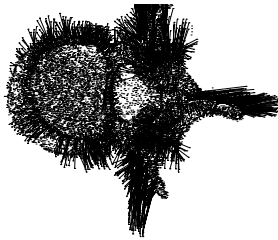


FIG. 4 — Position après recalage rigide

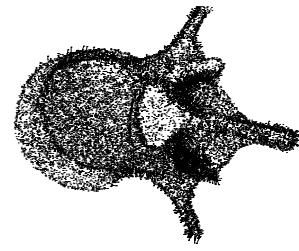


FIG. 5 — Après recalage élastique

Visualization in Biomedical Computing, pages 242–258, 1992.

- [3] M. Bro-Nielsen and K. Gramkow. Fast Fluid Registration of Medical Images. In Springer, editor, *Visualization in Biomedical Computing*, pages 267–276, September 1996.
- [4] L. Brunie. *Fusion d'images médicales multi-modales : application à l'étude tridimensionnelle dynamique de la colonne vertébrale (in french)*. PhD thesis, Grenoble University, december 1992.
- [5] D.L. Collins, G. Le Goualher, R. Venugopal, A. Caramanos, A.C. Evans, and C. Barillot. Cortical Constraints for Non-Linear Cortical Registration. In K. Hoehne and R. Kikinis, editors, *Visualization and Biomedical Computing*, pages 307–316. Springer-Verlag, September 1996.
- [6] A.C. Evans, D.L. Collins, P. Neelin, and T.S. Marrett. Correlative analysis of three-dimensional brain images. In R. Taylor, S. Lavallee, G. Burdea, and R. Mosges, editors, *Computer-integrated surgery : technology and clinical applications*, pages 99–114. MIT Press, Cambridge, MA, 1996.
- [7] Jacques. Feldmar. *Recalage rigide, non rigide et projectif d'images médicales tridimensionnelles*. PhD thesis, Ecole Polytechnique, november 1995.
- [8] J. H. Friedman, J.L. Bentley, and R. A. Finkel. An algorithm for finding best matches in logarithmic expected time. *ACM Trans. Math. Software*, 3(3) :209–226, Sept. 1977.
- [9] J.J. Jacq and C. Roux. Automatic registration of 3D images using a simple genetic algorithm with a stochastic performance function. In *IEEE Engineering in Medicine and Biology Society Proceedings*, pages 126–127, San Diego, CA, october 1993.
- [10] B. Kim, J.L. Boes, K.A. Frey, and Meyer C.R. Mutual Information for Automated Multimodal Image Warping. In Springer, editor, *Visualization in Biomedical Computing*, pages 349–354, September 1996.
- [11] S. Lavallee, R. Szeliski, and L. Brunie. Anatomy-based registration of 3-D medical images, range images, X-ray projections, 3-D models using Octree-Splines. In R. Taylor, S. Lavallee, G. Burdea, and R. Mosges, editors, *Computer Integrated Surgery*, pages 115–143. MIT Press, Cambridge, MA, 1996.
- [12] S. Lubiarsz and P. Lockwook. Evaluation of fast algorithms for finding the nearest neighbor. In *ICASSP97*, volume 2, pages 1491–1494, Munich, April 1997.
- [13] J.F. Mangin. *Mise en correspondance d'images médicales 3D multi-modalités multi-individus pour la corrélation anatomo-fonctionnelle cérébrale*. PhD thesis, Ecole Nationale Supérieure des Télécommunications, march 1995.
- [14] O. Monga, R. Deriche, G. Malandain, and J. P. Cocque-rez. Recursive filtering and edge closing : Two primary tools for 3D edge detection. In *First European Conference on Computer Vision (ECCV'90)*, pages 56–65, Antibes, France, April 1990. Springer-Verlag.
- [15] F. Preparata and M. Shamos. *Computational Geometry, An Introduction*. Springer Verlag, 1986.
- [16] C. Studholme, J. Little, G. Penny, D. Hill, and D. Hawkes. Automated multimodality registration using the full affine transformation : application to MR and CT guided skull base surgery. In K. Hoehne and R. Kikinis, editors, *Visualization and Biomedical Computing (VBC'96)*, pages 601–606. Springer-Verlag, 1996.
- [17] G. Subsol, J.P. Thirion, and N. Ayache. A General Scheme for Automatically Building 3D Morphometric Anatomical. Technical Report 2586, Atlases : application to a Skull AtlasINRIA, France, May 1995.
- [18] R. Szeliski and S. Lavallee. Matching 3-D anatomical surfaces with non-rigid deformations using octree-splines. *Int. J. of Computer Vision (IJCV)*, (18)(2) :171–186, MAY 1996.