Description et reconstruction rapide de surfaces

Laurent Chevalier, Fabrice Jaillet, Atilla Baskurt
Laboratoire d'Informatique Graphique, Image et Modélisation
LIGIM, EA 1899, Université Claude Bernard Lyon 1
http://ligim.univ-lyon1.fr

1 Introduction

Dans de nombreux domaines, il peut être intéressant de disposer de descripteurs performants associés aux surfaces présentes dans une image 3D afin d'obtenir une segmentation et une reconstruction rapide des objets tridimensionnels. Ces descripteurs doivent pouvoir être visualisés en temps réel. Ceci implique une modélisation simplifiée.

Nous décrivons les scènes ou objets 3D à l'aide de superellipsoïdes. Ce modèle de surface est très utilisé dans ce domaine d'application , car il est à la fois compact et généraliste. Il n'est cependant pas suffisant pour décrire des objets avec précision. Deux solutions sont alors possibles : soit on déforme l'unique superellipsoïde qui approxime les données, c'est l'approche de Bardinet [1] qui ajoute une « *Free Form Deformation (FFD)*» à une superellipsoïde pour modéliser des données médicales; soit on décompose l'objet en éléments primitifs, chaque élément devant être approximable correctement par une simple superellipsoïde [2, 3, 4, 5, 6]. Les approches diffèrent alors par le choix de l'algorithme de décomposition.

Dans cet article, nous comparons deux méthodes de segmentation des données. La première, due à Leonardis [2, 3], se base sur un algorithme ascendant de type "croissance de régions", tandis que la seconde, que nous développons, est une méthode descendante de type "division et fusion".

Nous présenterons d'abord succintement les superellipsoïdes et la manière d'approximer un ensemble de points de R^3 avec une superellipsoïde. Nous détaillerons ensuite les deux approches de segmentation. Puis nous finirons par une comparaison des résultats que nous obtenons.

2 Superellipsoïdes

2.1 Généralités

Une superellipsoïde est un modèle de surface introduit en informatique graphique en 1981 par A.H. Barr [7]. Elle peut être définie par la forme implicite :

$$f(x,y,z) = \left(\left(\frac{x}{a_1}\right)^{\frac{2}{\epsilon_2}} + \left(\frac{y}{a_2}\right)^{\frac{2}{\epsilon_2}} \right)^{\frac{\epsilon_2}{\epsilon_1}} + \left(\frac{z}{a_3}\right)^{\frac{2}{\epsilon_1}} \qquad (1)$$

avec :
- a_1, a_2, a_3 définissant le facteur d'échelle respectivement sur l'axe des x, y et z.
- ϵ_1 et ϵ_2 permettant de définir respectivement la courbure latitudinale et longitudinale de la forme.

et :

- $f(x,y,z) = 1$ si le point (x,y,z) est sur la surface.
- $f(x,y,z) < 1$ si le point (x,y,z) est à l'intérieur.
- $f(x,y,z) > 1$ si le point (x,y,z) est à l'extérieur.

Les superellipsoïdes décrivent un large éventail de formes allant de simples ellipsoïdes ($\epsilon_1 = 1$ et $\epsilon = 1$), aux parallélépipèdes ($\epsilon_1 \to 0$ et $\epsilon_2 \to 0$) et aux cylindres ($\epsilon_1 = 1$ et $\epsilon_2 \to 0$)(Fig. 1a). Elles permettent une modélisation compacte (seulement 5 paramètres), généraliste et particulièrement adaptée à notre application.

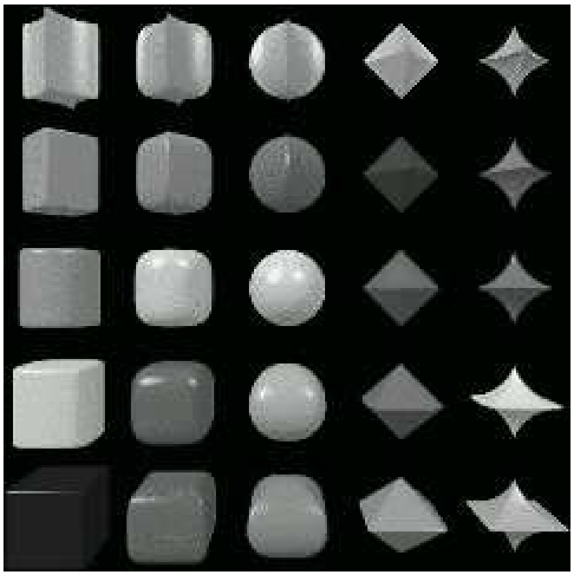


FIG. 1 – *Exemples de superellipsoïdes.*

2.2 Approximation du nuage par une superellipsoïde

Ayant pour données d'entrée un ensemble non organisé de N points de R^3, sans information de relation entre ces points, il s'agit de déterminer les valeurs des paramètres de notre modèle pour approximer ce nuage de points.

La méthode que nous utilisons a été proposée par Solina [8] et est la plus largement répandue pour approximer des données tridimensionnelles avec une superellipsoïde. L'approximation se fait au sens des moindres carrés.

L'approximation de la distance $d(x, y, z)$ d'un point à la surface est définie par:

$$d = f^{\frac{\epsilon_1}{2}} - 1 \tag{2}$$

L'unique considération de la distance des données au modèle n'est pas suffisante surtout lorsque le nuage est ouvert. En effet, plusieurs superellipsoïdes de tailles très diverses peuvent approximer le même nuage de points. Afin de résoudre ce problème, on privilégie les petites superellipsoïdes. Solina a montré, par expérimentation, que le meilleur choix était l'ajout du facteur $\sqrt{a_1 a_2 a_3}$.

En tenant compte de tous les paramètres de la superellipsoïde, de la translation (t_x, t_y, t_z) et de la rotation (r_x, r_y, r_z) éventuelles, l'approximation consiste à minimiser la fonction :

$$\sqrt{a_1 a_2 a_3} \sum_{i=1}^{N} \left(f\left(x_i, y_i, z_i; a_1, a_2, a_3, \epsilon_1, \epsilon_2, t_x, t_y, t_z, r_x, r_y, r_z\right)^{\frac{\epsilon_1}{2}} - 1 \right)^2 \tag{3}$$

qui représente l'erreur globale d'approximation.

L'approximation par une superellipsoïde est donc un problème nécessitant un algorithme de régression non linéaire sur 11 variables. Nous utilisons l'algorithme de Levenberg-Marquardt qui permet une convergence rapide.

3 Construction du descripteur

Notre objectif est d'obtenir un assemblage de primitives modélisant de manière relativement précise un nuage de points de R^3. Il nous faut une méthode de partitionnement des données telle que chaque sous-nuage s'approxime de manière idéale avec une superellipsoïde. Cette méthode doit produire un nombre minimum de primitives pour une qualité d'approximation donnée.

3.1 Méthode ascendante de Leonardis

Nous avons adapté une méthode de "Region Growing" [2, 3] (initialement développée pour des images en niveaux de gris) pour modéliser un nuage de points de l'espace. Le nuage va être découpé selon une grille, et le principe de l'accroissement de région va être appliqué : on fait croître des nuages et on sélectionne ceux qui permettent une description compacte du nuage initial.

3.1.1 Création des germes

Le nuage de points est initialement découpé de façon régulière. Nous éliminons les sous-nuages qui contiennent trop peu de points. Nous estimons une superellipsoïde sur chacun des sous-nuages restants. Les nuages dont l'erreur d'ajustement se situe en deçà de l'erreur fixée par l'utilisateur constituent les germes. Il est nécessaire de placer un grand nombre de germes lorsque l'on désire une erreur assez faible. Cela pose problème, car plus le nombre de germes est important, plus l'algorithme sera long à converger. Il est aussi important de vérifier que nous disposons de germes dans toutes les parties (connexes ou non) de l'objet pour éviter la formation de trous lors de la phase suivante.

3.1.2 Grossissement des régions

Une fois que nous disposons des germes, il va falloir les faire grossir. Pour cela, nous recherchons les points du nuage situés dans le voisinage. Cette phase est délicate car les points ne possèdent pas de structuration comme c'est le cas pour Leonardis avec ses images de profondeur. Nous avons donc choisi de considérer le k-voisinage, constitué des k plus proches voisins en terme de potentiel par rapport à la superellipsoïde. On teste l'aptitude du voisinage à être inséré dans le germe : on estime une superellipsoïde sur l'union des points du nuage et du voisinage. Si le test est positif (l'erreur commise est inférieure à l'erreur acceptée), les points sont insérés. Sinon, le voisinage est découpé aléatoirement en deux sous-nuages complémentaires et l'on recommence avec chacun des sous-voisinages. On garde celui pour lequel l'erreur est moindre, et on réitère l'opération tant qu'il reste un nombre suffisant de points dans le voisinage. Si le processus précédent a échoué, le germe est considéré comme mature et ne grossira plus.

3.1.3 Sélection des superquadriques

Nous utilisons un algorithme glouton pour sélectionner un ensemble non redondant de germes par maximisation de la forme quadratique de f décrite dans la section précédente. Pour cela, les coefficients d'une matrice Q sont calculés en fonction du contexte courant et de l'état de chaque superellipsoïde. Puis l'algorithme glouton permet d'obtenir un maximum local en sélectionnant de façon incrémentale les modèles correspondant à un terme diagonal positif maximal, jusqu'à ce qu'il n'en trouve plus. Les modèles non sélectionnés sont alors éliminés et deviennent inactifs.

Le calcul des termes de la matrice se fait de la façon suivante :

$$\begin{aligned} \xi[i] &= card(N[i]) * e[i] \text{ (erreur d'ajustement associée à } N[i]) \\ Q[i][i] &= K_1 card(N[i]) - K_2 \xi[i] \end{aligned} \tag{4}$$

et pour les termes en dehors de la diagonale :

$$\begin{aligned} \xi[i][j] &= max\left(\sum_{x \in N[i] \cap N[j]} potentiel(x, M[i]), \sum_{x \in N[i] \cap N[j]} potentiel(x, M[j])\right) \\ Q[i][j] &= Q[j][i] = (-K_1 card(N[i] \cap N[j]) + K_2 \xi[i][j])/2 \end{aligned} \tag{5}$$

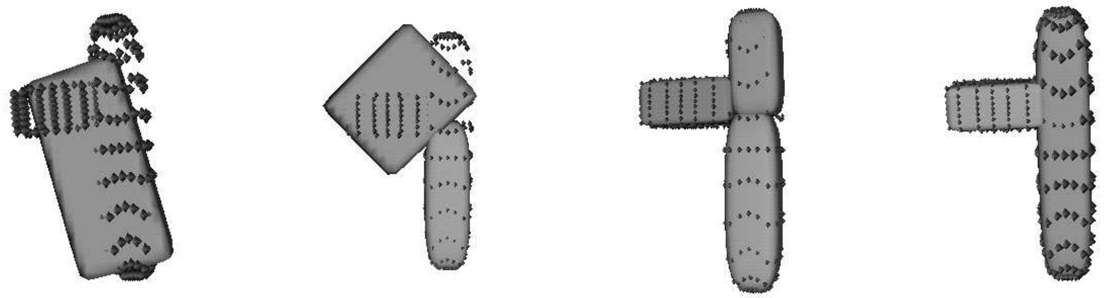
où $M[i]$ est la superellipsoïde associée au nuage $N[i]$. Les termes diagonaux concernent l'adéquation de la superellipsoïde à l'objet, alors que les termes hors diagonale concernent les points communs à deux superellipsoïdes. Lorsqu'un modèle i est choisi, sa contribution aux autres modèles est reportée dans la matrice de la façon suivante :

$$Q[j][j] = Q[j][j] + 2 * Q[j][i] \quad \text{pour} \quad j \neq i$$

À la fin de la phase de sélection, quand tous les germes sont matures, il reste une description compacte de l'objet initial à l'aide de superellipsoïdes.

3.2 Notre méthode descendante

L'algorithme de décomposition utilisé est de type « division et fusion ».



(a) (b) (c) (d)

Fig. 2 – *(a) Données originales et première approximation. (b) Première étape de division. (c) Secconde étape de division. Tous les sous-ensembles sont correctement approximés. La phase de fusion commence. (d) Deux sous-ensembles fusionnent. Le descripteur final est l'union de deux superellipsoïdes.*

3.2.1 Première étape : division du nuage

La phase de découpage est récursive. La première approximation du nuage se fait par une seule superellipsoïde. Si, à la suite de cette approximation, l'erreur D (Eq. 3) normalisée par le nombre de points N est inférieure à un seuil donné, alors le nuage reste intact et la superellipsoïde résultante est considérée comme un modèle de ce dernier. Dans le cas contraire, le nuage est scindé en deux. Ce partage se fait par le plan orthogonal à l'axe d'inertie du nuage et passant par son centre de gravité. Suite à cette scission, on considérera par le même procédé les deux nuages résultants de manière indépendante. Le mécanisme s'arrêtera, lorsque pour chaque région, le critère de division n'est plus respecté.

3.2.2 Deuxième étape : regroupement des régions similaires

La première étape fournit une partition des données, pour laquelle chaque région est associée à une superellipsoïde. Le nombre de superellipsoïdes constituant alors le descripteur est souvent très important.

En effet, le plan de coupe ne tenant pas compte des relations topologiques entre les points, la scission d'une région contenant une partie d'objet représentable par une seule superellipsoïde peut conduire à la division de cette même partie d'objet.

Dans la seconde étape, nous tentons de regrouper ces régions. Pour ce faire, la procédure d'approximation est appliquée à la fusion des nuages de chaque couple (région , voisin de cette même région). Si cette approximation est satisfaisante, les régions seront regroupées, sinon, elles resteront indépendantes. Cette étape s'arrête lorsque la partition des données est stable.

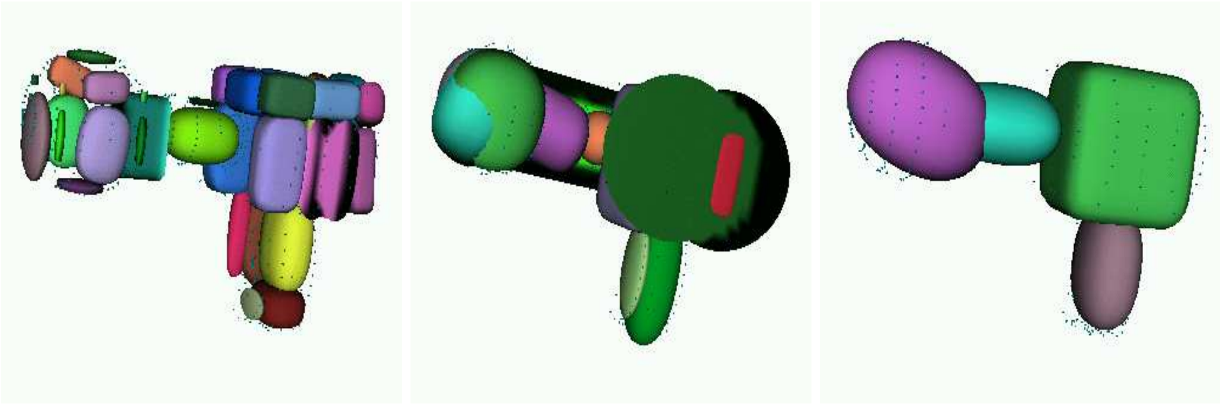
A la fin de cette deuxième et dernière étape, l'ensemble des superellipsoïdes a été restreint au mieux pour la qualité d'approximation voulue, compte tenu des critères utilisés dans la procédure « division/fusion » (Fig. 2).

4 Résultats et comparaison

Les deux méthodes ont été appliquées sur les mêmes données en entrée. La figure 3 présente les résultats de la première méthode présentée (ascendante), alors que la figure 4 illustre notre méthode (descendante).

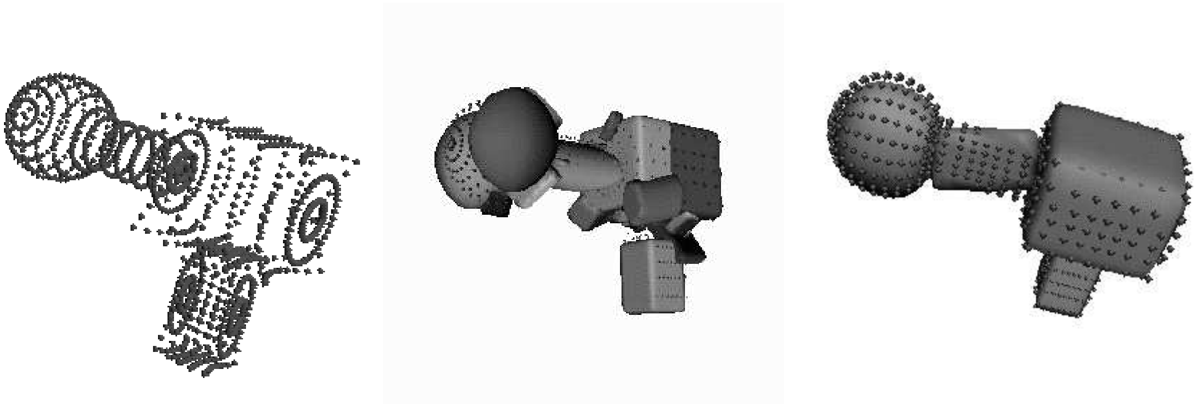
Les résultats présentés ici sont de qualité à peu près équivalente. Cependant, en règle générale, la méthode ascendante est plus longue à converger car elle part d'un grand nombre de germes et cherche à les faire grossir et à en éliminer pour obtenir une description la plus compacte possible. Les premières étapes sont donc très longues puisqu'elles travaillent sur beaucoup de données, mais cela permet d'avoir une description très précise de l'objet. Notre méthode produit un nombre légèrement plus petit de primitives. Un de ses intérêts est aussi sa progressivité, car elle commence par fournir une description grossière qui est affinée tout au long du processus et, ainsi, le descripteur peut être utile dans le cas de transmission sur un réseau.

	Nombre de points du nuage	Erreur commise	Nombre de primitives	Temps de calcul
ascendante	1500	0,0052	4	40 min
desscendante	1500	0,0026	4	10 min



(a) (b) (c)

Fig. 3 – *Méthode ascendante :(a) Les germes initiaux. (b) Étape intermédiaire. (c) Descripteur final (4 superellipsoïdes).*



(a) (b) (c)

Fig. 4 – *Méthode descendante : (a) Données originales. (b) Après la phase de découpage. (c) Descripteur final (4 superellipsoïdes).*

Références

[1] Eric BARDINET, Laurent D. COHEN, and Nicolas AYACHE, "A parametric deformable model to fit unstructured 3d data," Research report 2617 - INRIA, 1995.

[2] Ales LEONARDIS, Franc SOLINA, and Alenka MACERL, "A direct recovery of superquadric models in range images using recover-and select paradigm," *Proceedings of 3rd European Conference on Computer Vision*, 1994.

[3] Ales LEONARDIS, Ales JAKLIC, and Franc SOLINA, "Superquadrics for segmenting and modeling range data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 11, pp. 1289–1295, November 1997.

[4] Alok GUPTA and Ruzena BAJCSY, "Volumetric segmentation of range images of 3d objects using superquadric models," *CVGIP : Image understanding*, vol. 58, no. 3, pp. 302–326, November 1993.

[5] Demetri TERZOPOULOS and Dimitri METAXAS, "Dynamic 3D models with local and global deformations: deformable superquadrics," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 7, pp. 703–714, July 1991.

[6] Alex PENTLAND, "Recognition by parts," *IEEE First Int. Conf. on Computer Vision*, pp. 612–620, 1987, London.

[7] Alan H. BARR, "Superquadrics and angle preserving transformations," *IEEE Computer Graphics and Applications*, vol. 1, no. 1, pp. 11–23, January 1981.

[8] Franc SOLINA and Ruzena BAJCSY, "Recovery of parametric models from range images: the case for superquadrics with global deformations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 2, 1991.