

Segmentation et modélisation 3D par un ensemble de superellipsoïdes

Laurent Chevalier, Fabrice Jaillet, Atilla Baskurt
Laboratoire LIRIS (CNRS UMR5205)
Université de Lyon - Bât Nautibus
8-10 Bld Niels BOHR
F-69622 Villeurbanne Cedex
{lchevalier,fjaillet,abaskurt}@liris.cnrs.fr

6 septembre 2005

Résumé

Nous proposons dans cet article un nouveau modèle pour représenter un ensemble de points 3D non-organisés. Basé sur les superquadriques, ce modèle permet de décrire l'ensemble de points avec une union de superellipsoïdes. Deux méthodes différentes de segmentation et de modélisation sont développées pour obtenir le modèle complet : l'une de type Croissance de Région et l'autre Découpage/Fusion. Cette dernière fournit un modèle peu sensible en comparaison à celui obtenu par Croissance de Région. Le modèle est simple et compact : seulement 11 paramètres sont nécessaires pour chaque superellipsoïde. Cela semble prometteur pour de la compression d'objets 3D, voire pour de l'indexation 3D. Comme les relations topologiques entre les superellipsoïdes sont connues, le modèle peut être associé à un graphe. Et on peut alors utiliser la théorie des graphes pour comparer et mesurer les similarités entre objets 3D.

Abstract

In this article we propose an innovative model to represent a non-organised 3D points set. Based on superquadrics, this model permits to describe the points set with a superellipsoid union. Two different methodologies are developed for segmenting and modelling the whole representation : a Region Growing one, and a Split and Merge one. This latter provides a low sensitive model compared to the first method. The representation is simple and compact, as only 11 parameters are required per superellipsoid. This seems promising to be used for applications such as 3D compression, or even 3D indexing and retrieval. While the relationship between superellipsoids is known, the model could be associated to a graph. Yet, graph theory can be used to compare and measure similarities between 3D objects.

Motsclés

Informatique graphique, segmentation et modélisation 3D, ensemble de superellipsoïdes.

Keywords

Computer graphics, 3D segmentation and modelling, superellipsoids set.

1 Modélisation avec des superellipsoïdes

Cette étude concerne la segmentation et la modélisation d'un ensemble de points 3D non-organisés. Les contraintes imposées sur le modèle dépendent de l'application considérée :

- visualisation rapide des objets 3D représentés par l'ensemble de points 3D ;

- indexation et recherche d’objets similaires dans des bases de données dédiées, fournissant un descripteur ;
- compression de l’ensemble des données pour la transmission et le stockage.

Nous avons besoin d’un descripteur simple, qui permette la représentation d’un ensemble 3D avec un modèle très compact, et la reconstruction d’une version grossière avec un contrôle de la distorsion. Dans ce but, nous avons choisit de décrire les objets 3D avec un ensemble de primitives de type superellipsoïde.

Les superquadriques sont une extension de la famille des quadriques et permettent de représenter un éventail assez large de formes élémentaires tout en étant particulièrement compactes en terme de nombre de paramètres. Comme les quadriques, les superquadriques sont divisées en quatre classes : superparaboloïde, superhyperboloïde, supertoroïde et superellipsoïde. Cette dernière est la classe la plus largement utilisée en modélisation informatique car elle permet le plus naturellement, le plus souvent par assemblage, de modéliser des objets tridimensionnels.

Les superellipsoïdes ont déjà été utilisées pour modéliser des objets 3D [Bar81, Pen87, LJS97]. Dans la majorité de ces travaux, les données disponibles pour représenter les objets ou les scènes 3D étaient des images de profondeur. Celles-ci ont une organisation régulière et sont considérée comme organisées, dans le sens où des points voisins dans l’image de profondeur sont aussi voisins dans l’espace. Nous cherchons à généraliser ces approches pour des données quelconques, sans connaissance *a priori* : le nuage de points est considéré comme étant irrégulier et non-organisé.

Nous commençons par décrire les outils mathématiques dont nous avons besoin : la formulation d’une superellipsoïde et la reconstruction d’un sous-ensemble de points par une unique superellipsoïde. Ensuite, nous proposons de comparer différentes façons d’obtenir le descripteur complet. La première est une extension de la méthode de croissance de région [LSM94, LJS97]. La seconde est une approche originale que nous avons développée. Dans les deux cas, nous discutons des performances qualitatives et quantitatives.

1.1 De la superellipse à la superellipsoïde

Les quadriques résultent du produit sphérique de deux coniques. Ainsi, les superellipsoïdes peuvent être définies de manière similaire comme le produit sphérique de deux superellipses [Bar81].

$$\begin{aligned}
 S(\eta, \mu) &= \begin{bmatrix} \cos^{\epsilon_1}(\eta) \\ a_3 \sin^{\epsilon_1}(\eta) \end{bmatrix} \otimes \begin{bmatrix} a_1 \cos^{\epsilon_2}(\mu) \\ a_2 \sin^{\epsilon_2}(\mu) \end{bmatrix} \\
 &= \begin{bmatrix} a_1 \cos^{\epsilon_1}(\eta) \cos^{\epsilon_2}(\mu) \\ a_2 \cos^{\epsilon_1}(\eta) \sin^{\epsilon_2}(\mu) \\ a_3 \sin^{\epsilon_1}(\eta) \end{bmatrix}, \quad \begin{array}{l} -\frac{\pi}{2} \leq \eta \leq \frac{\pi}{2} \\ -\pi \leq \mu \leq \pi \end{array}
 \end{aligned} \tag{1}$$

On remarquera que la participation des deux courbes au produit n’est pas la même, le produit sphérique n’étant pas commutatif. Intuitivement, la courbe $m(\eta)$ définit la forme des courbes de latitude de la surface tandis que $n(\mu)$ définit celle des courbes de longitude. Les paramètres a_1 , a_2 et a_3 définissent la taille (facteur d’échelle) de la superellipsoïde suivant les axes x , y et z respectivement, tandis que les paramètres ϵ_1 et ϵ_2 caractérisent, respectivement, la courbure latitudinale et longitudinale de la forme. Les paramètres ϵ_1 et ϵ_2 , n’ont donc pas une influence équivalente sur la forme de la superellipsoïde. Par exemple, un ■ cylindre ■ s’obtient avec ϵ_1 proche de 0 et $\epsilon_2 = 1$, mais pas avec l’inverse.

Grâce à ces 5 paramètres, les superellipsoïdes vont nous permettre de modéliser un grand nombre de formes élémentaires de toutes tailles, allant de la simple ellipsoïde au parallélépipède rectangle en passant par le cylindre ou l’octaèdre, et bien sûr toutes les formes intermédiaires (Fig. 1).

On peut également exprimer les superellipsoïdes sous forme implicite. Le passage depuis la forme paramétrique se fait en éliminant d’abord μ , puis η . La surface de la superellipsoïde est l’ensemble des points solution de :

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

La formulation des modèles implicites permet de partitionner très simplement l’espace tridimensionnel en trois lieux. Les superellipsoïdes ne dérogent pas à la règle. On rappelle donc que pour tout point P de coordonnées (x, y, z) , on aura :

- $f(x, y, z) = 1$, si P est sur la surface de la superellipsoïde ;

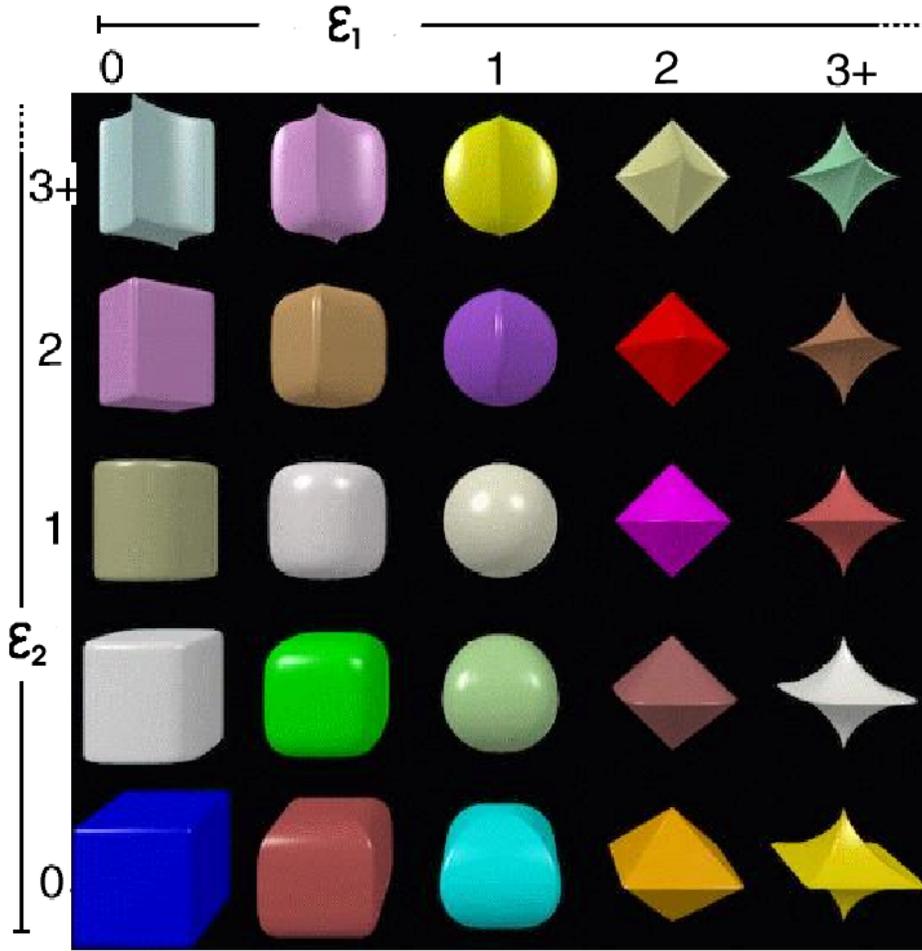


FIG. 1 – Exemples de superellipsoïdes

- $f(x, y, z) > 1$, si P est à l'extérieur de la surface de la superellipsoïde ;
- $f(x, y, z) < 1$, si P est à l'intérieur de la surface de la superellipsoïde.

La particularité des superellipsoïdes à pouvoir être définies à la fois sous forme paramétrique et implicite est intéressante, et surtout, il s'agit d'une des raisons majeures de notre choix. En effet, la forme implicite permet d'obtenir à moindre coût (en terme de calculs) des informations sur la position d'un point par rapport à la surface, ce qui nous sera utile pour l'approximation. La forme paramétrique, quant à elle, permet un échantillonnage immédiat de la surface autorisant ainsi un affichage rapide, ce qui est loin d'être le cas avec une forme implicite pure, malgré quelques travaux récents dans ce sens.

Comme pour les superellipses, plusieurs paramétrages différents permettent d'obtenir sensiblement la même surface. En premier lieu, parce que les superellipsoïdes contiennent des axes de symétrie les rendant invariant par rotation. Par exemple, les paramètres a_1 et a_2 peuvent être échangés sans modifier la forme de la surface (à une rotation près). Mais aussi parce que l'on pourra obtenir un cube avec deux paramétrages, en utilisant pour la courbe définissant la latitude : soit $a_1 = a_2 = 1$ et ϵ_1 proche de 0, soit $a_1 = a_2 = \sqrt{2}$ et $\epsilon_1 = 2$. Cela ne pose pas vraiment de problème lorsque les superellipsoïdes sont employées dans un but d'approximation d'objets ; mais cela doit être pris en compte si on veut comparer des superellipsoïdes entre elles en se basant sur leurs paramètres, pour des applications de reconnaissance de formes, par exemple.

1.2 Paramétrisation régulière d'une superellipsoïde

La forme paramétrique de la superellipsoïde, comme on l'a définie par le produit sphérique de deux superellipses, a la particularité de ne pas produire un échantillonnage régulier, mais au contraire de concentrer les points dans les zones de fortes courbures. Cette paramétrisation est donc très intéressante pour les applications de type affichage puisque plus précise là où il y en a le plus besoin. Néanmoins, d'autres applications peuvent nécessiter une répartition uniforme des points à la surface de la superellipsoïde (Fig. 2). Bardinet propose une méthode de paramétrisation permettant de garder, quelle que soit la valeur des coefficients de courbure ϵ_1 et ϵ_2 , une répartition régulière des courbes latitudinales et longitudinales [BCA95].

Remarquant que l'échantillon de points pour une sphère est régulièrement réparti, l'idée est de projeter ces points sur la superellipsoïde. On obtient alors une paramétrisation permettant un échantillonnage bien plus régulier :

$$S'(\eta, \mu) = \begin{bmatrix} a_1 \rho(\eta, \mu) \cos(\eta) \cos(\mu) \\ a_2 \rho(\eta, \mu) \cos(\eta) \sin(\mu) \\ a_3 \rho(\eta, \mu) \sin(\eta) \end{bmatrix}, \quad \begin{array}{l} -\frac{\pi}{2} \leq \eta \leq \frac{\pi}{2} \\ -\pi \leq \mu \leq \pi \end{array} \quad (3)$$

avec :

$$\rho(\eta, \mu) = \left(\left(|\cos(\mu) \cos(\eta)|^{\frac{2}{\epsilon_2}} + |\sin(\mu) \cos(\eta)|^{\frac{2}{\epsilon_2}} \right)^{\frac{\epsilon_2}{\epsilon_1}} + |\sin(\eta)|^{\frac{2}{\epsilon_1}} \right)^{-\frac{\epsilon_1}{\epsilon_2}}$$

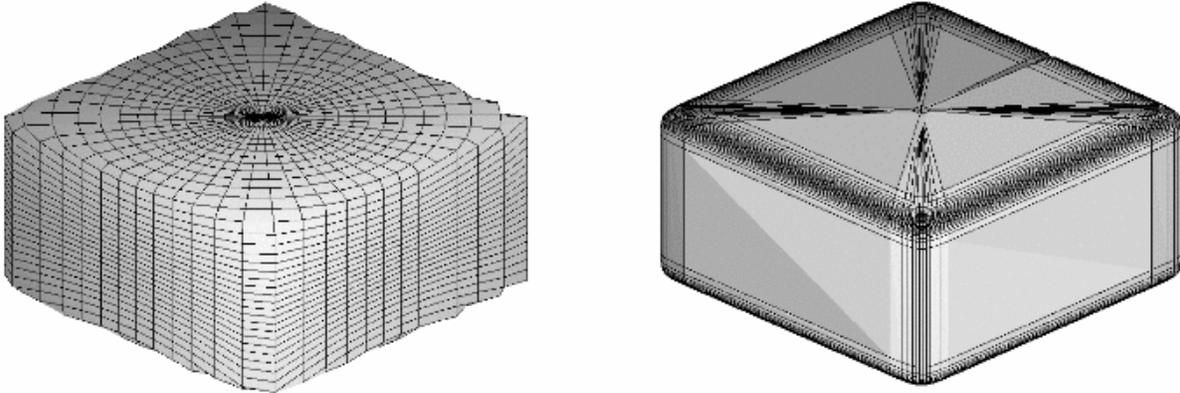


FIG. 2 – Paramétrisation régulière et paramétrisation standard.

Précisons que, comme le montre la figure 2, l'échantillonnage de la superquadrique n'est pas complètement régulier si on considère l'intersection des lignes latitudinales et longitudinales. Il y a une concentration d'intersections, et donc de points, au niveau des pôles. Mais l'espacement entre ces lignes, lui, est régulier.

1.3 Vecteur normal d'une superellipsoïde

La connaissance du vecteur normal en tout point à une surface est très utile, par exemple pour calculer la distance d'un point à la surface. Nous allons voir que l'expression du vecteur normal à une superellipsoïde possède une forme remarquable et peut-être calculé très simplement [Bar81, JLS00] : par le produit vectoriel des tangentes en ce point :

$$N(\eta, \mu) = \frac{\delta S_\eta(\eta, \mu)}{\delta \eta} \wedge \frac{S_\mu(\eta, \mu)}{\delta \eta} \quad (4)$$

Ainsi, la direction du vecteur normal est de forme remarquable, puisque après simplification, on obtient :

$$\begin{cases} N'_x(\eta, \mu) = \frac{1}{a_1} \cos^{2-\epsilon_1}(\eta) \cos^{2-\epsilon_2}(\mu) \\ N'_y(\eta, \mu) = \frac{1}{a_2} \cos^{2-\epsilon_1}(\eta) \sin^{2-\epsilon_2}(\mu) \\ N'_z(\eta, \mu) = \frac{1}{a_3} \sin^{2-\epsilon_1}(\eta) \end{cases}, \quad \begin{array}{l} \eta \in [-\frac{\pi}{2}, \frac{\pi}{2}] \\ \mu \in [-\pi, \pi] \end{array}$$

et pour ϵ_1 et ϵ_2 inférieurs à 2, l'équation des normales définit une autre superellipsoïde (sa duale) de paramètres de courbure $\epsilon'_1 = 2 - \epsilon_1$ et $\epsilon'_2 = 2 - \epsilon_2$ et de dimension $a'_1 = \frac{1}{a_1}$, $a'_2 = \frac{1}{a_2}$ et $a'_3 = \frac{1}{a_3}$.

1.4 Distance d'un point à la surface d'une superellipsoïde

La capacité de pouvoir relativement facilement exprimer la distance d'un point tridimensionnel à sa surface est une condition nécessaire pour l'application de modélisation de données à laquelle nous destinons la superellipsoïde. En effet, c'est grâce à cette distance que nous pourrions juger de la qualité d'une approximation.

Il n'est pas envisageable de calculer la distance euclidienne d'un point à la surface d'une superellipsoïde. Son évaluation nécessite un procédé de régression itératif (Fig. 3). Nous étudierons ici différentes approches d'approximation de cette distance. Pour chacune d'entre elles, ont été tracées les courbes d'équidistance de coupes de différentes superellipsoïdes dans le but de leur donner un côté plus visuel. Pour obtenir ces champs de distance, nous avons calculé la distance de chaque pixel à la surface de la superellipsoïde. Bien entendu, la même échelle a été utilisée pour toutes ces courbes.

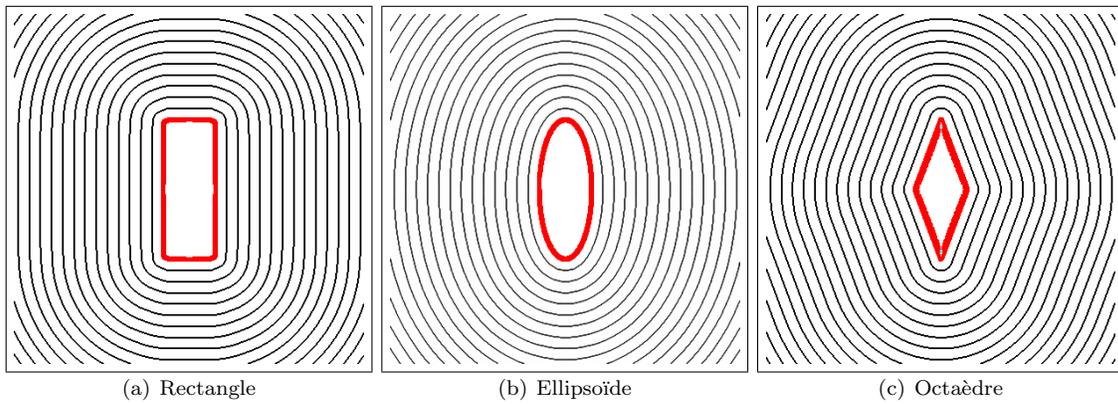


FIG. 3 – Distance euclidienne à une superellipsoïde (tracée en gras et en rouge) ; 15min de calcul.

Dans toute cette partie, nous considérerons que nous essayons de déterminer la distance d'un point P par rapport à une superellipsoïde S (se reporter à la Fig. 4 pour les notations).

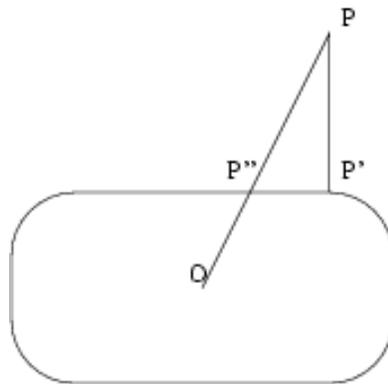


FIG. 4 – Notations utilisées : O le centre de la superellipsoïde S , P le point dont on désire connaître la distance à la surface, P' sa projection orthogonale sur S et P'' l'intersection entre la droite OP et la surface S

Fonction potentiel

Beaucoup d'approximations de la distance se basent sur la fonction potentiel de la superellipsoïde (Éq. 2). Il est vrai que cela est à la fois simple et peu coûteux. Par contre, si l'on désire utiliser cette fonction potentiel pour estimer la distance d'un point à la surface, cela n'est pas satisfaisant. Le premier problème est que les valeurs de ce potentiel varient énormément suivant les valeurs des coefficients de courbure. Le potentiel d'un point à une distance euclidienne donnée d'une ellipsoïde, est beaucoup plus élevé que le potentiel de ce même point à un parallélépipède de même taille et plus faible que le potentiel à un octaèdre. Et comme l'a déjà fait remarquer Solina [SB90], dans le cas d'un ϵ_1 petit, le potentiel suivant l'axe des z va croître très rapidement. Deux autres problèmes sont bien visibles. D'une part, le potentiel ne croît pas linéairement, et d'autre part, la croissance du potentiel n'est pas la même dans toutes les directions. Ce phénomène sera d'autant plus fort que le rapport entre la taille de l'axe principal d'inertie et cet autre axe sera grand.

Fonction standard d'approximation de distance

Solina propose une méthode pour approximer les ensembles de points par une superellipsoïde [SB90]. Pour juger de la distance d'un point à la surface, il utilise le critère d'erreur suivant F_1 , qui deviendra l'estimation de la distance la plus généralement utilisée dans le cas d'approximations :

$$F_1(x, y, z) = f(x, y, z)^{\frac{\epsilon_1}{2}} - 1 \quad (5)$$

Ce critère d'erreur supprime le problème lié au ϵ_1 petit. De plus, sa valeur semble la même pour une distance donnée quelles que soient les valeurs des coefficients de courbure (Fig. 5). Il a une valeur qui croît de façon linéaire quand on s'éloigne du centre dans une direction donnée. Le seul problème persistant est qu'il ne croît pas à la même vitesse dans toutes les directions. Deux points à même distance euclidienne de la surface n'auront donc pas forcément la même valeur.

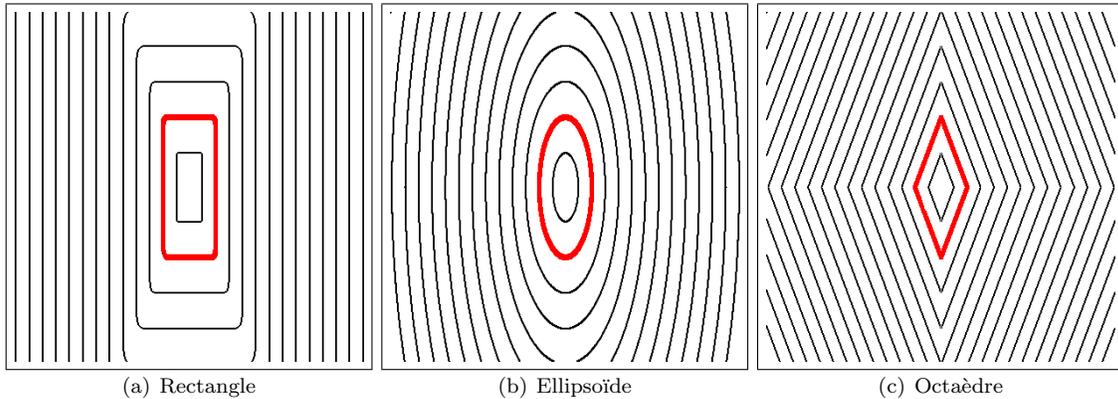


FIG. 5 – Critère d'erreur Solina F_1 ; 1,36s de calcul

Autres critères de distance

Boult et Gross ont proposé un critère (E_3 dans [BG87]) qui a la particularité de ne pas utiliser la forme implicite. Il est défini ainsi :

$$F(x, y, z) = \frac{\|\overrightarrow{PP''}\|}{\sqrt{(P_x - P''_x)^2 + (P_y - P''_y)^2 + (P_z - P''_z)^2}} \quad (6)$$

L'originalité de la méthode est de calculer les coordonnées de P'' grâce à la formule paramétrique. Pour cela il s'agit en fait de déterminer les angles : η et μ . Ce critère est effectivement très proche de la distance euclidienne radiale (voir ci-dessous). On peut cependant noter certains problèmes pour les petites valeurs de ϵ_2 . De plus, le critère n'est pas défini pour les valeurs sur les axes.

Yokoya propose lui aussi une distance n'utilisant pas la forme implicite [YKY92]. Sa méthode est assez proche de la précédente, mais il utilise en plus la normale en chaque point des données pour améliorer la qualité son approximation.

Une approche permettant d'obtenir une approximation plus précise de la distance est proposée Taubin [Tau91]. Si le point considéré est proche de la surface, on peut en effet utiliser l'approximation au premier ordre en série de Taylor [LSM94, BCA95] :

$$F(P) = F(P') + F'(P') \cdot (\overrightarrow{P'P}) + O(\|\overrightarrow{P'P}\|^2) \quad (7)$$

Son comportement est proche de la distance euclidienne dans le cas d'une ellipsoïde et l'effet ■ dilatation ■ n'existe plus. Notons qu'à cause du calcul du gradient, le coût de cette fonction est environ quatre fois plus important que pour F_1 (Éq. 5).

Distance ■ euclidienne radiale ■

La distance ■ euclidienne radiale ■ permet de corriger la plupart de ces problèmes. Il s'agit de la distance du point P au point d'intersection entre la surface et la droite passant par P et le centre de la superellipsoïde, c'est-à-dire $\|\overrightarrow{PP''}\|$ [BCA95]. Cette distance est égale à la distance euclidienne si P' et P'' sont confondus, c'est-à-dire si la droite (OP) est confondue avec l'un des axes de la superellipsoïde. En nommant F_2 cette fonction de ■ distance ■ :

$$F_2(x, y, z) = \|\overrightarrow{PP''}\| = \|\overrightarrow{OP}\| * |1 - f^{\frac{c_1}{2}}(P)| \quad (8)$$

Cette approximation n'est pas beaucoup plus coûteuse que le critère d'erreur de Solina, puisqu'il suffit, en plus, de calculer la distance euclidienne de O à P , deux points dont nous connaissons les coordonnées. Elle a pour avantage d'effacer l'effet ■ dilatation ■ du potentiel F_1 (Fig. 6). Sa croissance est similaire quelle que soit la direction de (OP) .

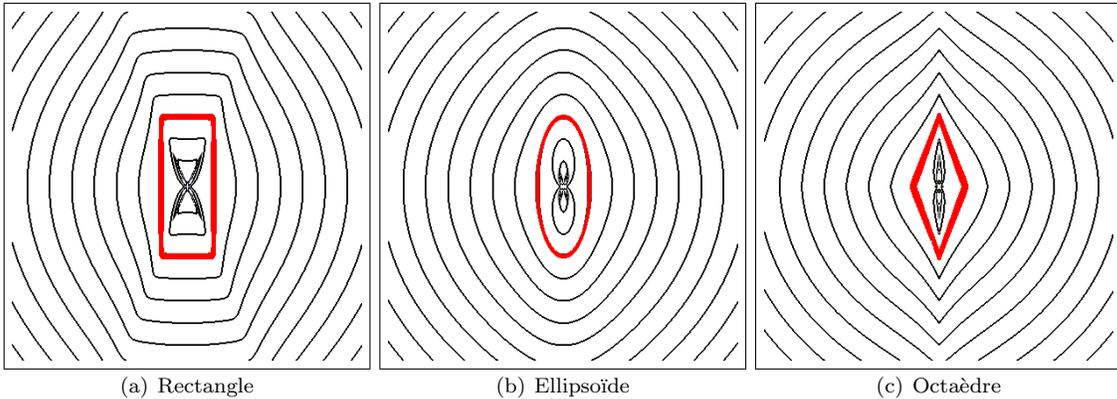


FIG. 6 – Distance euclidienne radiale F_2 ; 1,51s de calcul

Bien que celle proposée par Solina (F_1) soit la plus utilisée dans la littérature, la distance euclidienne radiale F_2 est de loin la plus séduisante. Elle paraît, si on s'en tient aux courbes d'équidistance, de qualité semblable à la distance euclidienne approchée pour un coût en temps de calcul bien inférieur.

2 Approximation d'un ensemble de points 3D par une superellipsoïde

La première étape pour obtenir un descripteur à base de superellipsoïdes, est de savoir approximer un ensemble de points 3D par une seule superellipsoïde. C'est-à-dire, être capable de déterminer les 11 paramètres permettant de définir la superellipsoïde modélisant le mieux ces données.

La méthode que nous utilisons est une adaptation de celle proposée par Solina et Bajcsy [SB90], de type approximation au sens des moindres carrés. Nous considérons que la superellipsoïde qui approche le plus vraisemblablement les données est celle dont la somme des distances au carré de chaque point à sa surface est la plus faible. Pour un ensemble de N points tridimensionnels, il s'agit de trouver le vecteur des 11 paramètres $\mathbf{a} = [a_1, a_2, a_3, \epsilon_1, \epsilon_2, t_x, t_y, t_z, \phi, \theta, \psi]$ qui minimise :

$$\sum_{i=1}^N d(\mathbf{x}_i, \mathbf{a})^2 \quad (9)$$

avec d la distance d'un point à la superellipsoïde (classiquement F_1 , mais nous lui préférons F_2), où t_x , t_y et t_z représentent les translations, et ϕ , θ , et ψ sont les angles de rotation, suivant les axes x , y et z respectivement.

Solina *et al.* travaillaient sur des données de type images de profondeur. Les objets qu'ils approximaient étaient par conséquent incomplets (une seule face visible). Beaucoup de superellipsoïdes peuvent alors correspondre aux données. Pour résoudre ce problème, ils vont introduire le facteur $(a_1 * a_2 * a_3)^{\frac{1}{3}}$ dans la formule à minimiser privilégiant ainsi les petites superquadriques. Finalement, le problème à résoudre revient donc à trouver \mathbf{a} minimisant :

$$\Delta(\mathbf{a}) = (a_1 * a_2 * a_3)^{\frac{1}{3}} \sum_{i=1}^N d(\mathbf{x}_i, \mathbf{a})^2 \quad (10)$$

La résolution ne peut se faire de manière directe. Il est alors nécessaire d'utiliser un algorithme de régression. La méthode se décompose alors en deux étapes. La première est de trouver un ensemble de paramètres initiaux, suivie d'une seconde étape de minimisation itérative.

2.1 Estimation des paramètres initiaux

Cette étape de première estimation des paramètres est très importante pour la suite de l'approximation. Il s'agit donc d'estimer les paramètres les plus proches possible de la solution afin de donner le meilleur point de départ à l'algorithme de régression, qui outre accélérer le processus, permettra d'éviter de rester piégé par les minima locaux. Il n'est pas possible d'estimer les coefficients de courbure de manière simple. C'est pourquoi on fixe arbitrairement ceux d'une ellipsoïde : ϵ_1 et $\epsilon_2 = 1$.

2.1.1 Ellipsoïde d'inertie

L'approche de Solina *et al.* est d'utiliser le centre de gravité des données et la matrice des moments centraux afin de déterminer la position et l'orientation de l'ellipsoïde. Comme nous pouvons le pressentir, cette approche statistique d'estimation fait l'hypothèse de données régulièrement réparties sur la surface.

Le centre de la superellipsoïde (permettant d'évaluer les paramètres de translation) est initialement le centre de gravité de l'ensemble de points :

$$t_x = \bar{x}; \quad t_y = \bar{y}; \quad t_z = \bar{z}$$

Pour estimer l'orientation de la superellipsoïde (les paramètres de rotation) et sa taille (a_1 , a_2 , a_3), on utilise la matrice des moments centraux d'inertie d'ordre 2 :

$$M = \frac{1}{N} \sum_{i=1}^N \begin{pmatrix} (y_i - \bar{y})^2 + (z_i - \bar{z})^2 & -(y_i - \bar{y})(x_i - \bar{x}) & -(z_i - \bar{z})(x_i - \bar{x}) \\ -(x_i - \bar{x})(y_i - \bar{y}) & (x_i - \bar{x})^2 + (z_i - \bar{z})^2 & -(z_i - \bar{z})(y_i - \bar{y}) \\ -(x_i - \bar{x})(z_i - \bar{z}) & -(y_i - \bar{y})(z_i - \bar{z}) & (x_i - \bar{x})^2 + (y_i - \bar{y})^2 \end{pmatrix}$$

On cherche alors la matrice de rotation R qui diagonalise M :

$$D = R^{-1}MR$$

R est la matrice dont les colonnes sont les vecteurs propres de M , et D la matrice diagonale contenant les valeurs propres de M : λ_1 , λ_2 et λ_3 . R est aussi la matrice de rotation permettant de passer du

repère absolu au repère intrinsèque de l'ellipsoïde. On va donc déduire de cette matrice l'estimation des paramètres de rotation en l'identifiant à une matrice de rotation standard. On aura donc, par exemple :

$$\begin{aligned}\phi &= -\arcsin(R_{1,3}) \\ \theta &= \arctan\left(-\frac{R_{2,3}}{\cos(\theta)}, -\frac{R_{3,3}}{\cos(\theta)}\right) \\ \psi &= \arctan\left(-\frac{R_{1,2}}{\cos(\theta)}, -\frac{R_{1,1}}{\cos(\theta)}\right)\end{aligned}$$

avec ϕ , θ et ψ les angles de rotation respectivement sur les axes x , y et z .

On a alors les paramètres de translation et de rotation. Il nous faut maintenant estimer les trois derniers paramètres de l'ellipsoïde, c'est-à-dire ceux définissant sa taille (a_1 , a_2 , a_3). Rappelons que nous nous bornons à estimer les paramètres d'une simple ellipsoïde. Dans l'approche originale, Solina utilisait la distance du centre de gravité au point le plus éloigné suivant chaque axe pour déterminer les paramètres de taille. Cette méthode étant extrêmement sensible aux points aberrants, nous lui préférons celle de Bardinet [BCA95] qui en comparant la matrice D et la matrice d'inertie J d'une ellipsoïde :

$$J = \frac{1}{3} \begin{pmatrix} a_2^2 + a_3^2 & 0 & 0 \\ 0 & a_1^2 + a_3^2 & 0 \\ 0 & 0 & a_1^2 + a_2^2 \end{pmatrix}$$

permet de calculer :

$$\begin{aligned}a_1^2 &= \frac{3}{2}(\lambda_2 + \lambda_3 - \lambda_1) \\ a_2^2 &= \frac{3}{2}(\lambda_1 + \lambda_3 - \lambda_2) \\ a_3^2 &= \frac{3}{2}(\lambda_1 + \lambda_2 - \lambda_3)\end{aligned}$$

On a ainsi estimé les neuf paramètres de l'ellipsoïde. Mais les valeurs obtenues ont été attribuées de manière arbitraire à chaque axe. Cela n'a pas d'importance dans le cas d'une ellipsoïde puisque que cette dernière est symétrique suivant les trois axes. Mais pour une superellipsoïde, il y a alors un choix à faire sur l'orientation du modèle et décider à quel axe on attribuera la plus grande taille (i.e. l'axe d'inertie). Le choix se limite en fait à x ou z , puisque x et y ont une influence complètement équivalente. Ne pouvant estimer les paramètres de courbure de la superellipsoïde, nous ne pouvons que faire un choix arbitraire. Nous décidons que z sera l'axe d'inertie ce qui privilégie les cylindres de type \blacksquare tuyau \blacksquare à l'encontre des \blacksquare boîtes de camembert \blacksquare . De toutes façons, ce choix n'est théoriquement pas définitif puisque, lors de l'ajustage des paramètres, la rotation a une liberté de 2π sur tous les axes.

Le résultat est relativement proche de la solution dans la plupart des cas, mais dépendant entièrement de la répartition des données, il peut s'en éloigner fortement dans le cas de données irrégulières. Il faut se rappeler que Solina travaillait sur des données de type image de profondeur qui bien qu'incomplètes sont beaucoup mieux réparties que celles auxquelles nous voulons nous confronter.

2.1.2 Estimation basée sur l'approximation directe d'ellipse

Évoquons l'idée d'une estimation des paramètres basée sur une méthode d'approximation d'ellipsoïdes au sens des moindres carrés, non-itérative, qui serait l'extension de celle existant pour les ellipses proposée par Fitzgibbon [FPF99]. Son approche est la suivante : une conique générale peut être représentée par l'équation implicite appelé \blacksquare distance algébrique \blacksquare :

$$F(\mathbf{a}, \mathbf{x}) = \mathbf{a} \cdot \mathbf{x} = ax^2 + bxy + cy^2 + dx + ey + f = 0$$

avec $\mathbf{a} = [a, b, c, d, e, f]$ et $\mathbf{x} = [x^2, xy, y^2, x, y, 1]$.

Si l'on désire approximer un ensemble de N points 2D, une approche est de minimiser la somme des carrés de la \blacksquare distance algébrique \blacksquare d'un point à la courbe :

$$E(a) = \sum_{i=1}^N F(\mathbf{a}, \mathbf{x}_i)^2$$

sous la contrainte que la conique représentée par \mathbf{a} soit une ellipse, c'est-à-dire que :

$$4ac - b^2 > 0$$

soit :

$$a^T C a = a^T \begin{pmatrix} 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} a > 0$$

Notons que cette contrainte interdit aussi la solution triviale à savoir \mathbf{a} est un vecteur nul.

Si on considère maintenant la matrice de donnée $D = [\mathbf{x}_1, \mathbf{x}_2 \dots \mathbf{x}_N]^T$, le problème devient : minimiser $E(\mathbf{a}) = |D\mathbf{a}|^2$ avec $\mathbf{a}^T C \mathbf{a} > 0$.

Si on devait en rester là, le problème serait complexe, mais puisque toute conique définie par l'ensemble de paramètres \mathbf{a} peut aussi l'être avec les paramètres $\mathbf{a}' = k * \mathbf{a}$ pour tout $k \in \mathbb{R}^*$, on peut simplifier la contrainte par $\mathbf{a}'^T C \mathbf{a}' = 1$. Cette simplification nous permet d'utiliser le théorème de Lagrange. On a :

$$2D^T D a - 2\lambda C a = 0$$

$$D^T D a = \lambda C a$$

$$\frac{1}{\lambda} * a = (D^T D)^{-1} C a$$

Ce système se résout grâce à une simple extraction des vecteurs/valeurs propres.

Tentative de passage en 3D

Il serait très intéressant de pouvoir utiliser une telle méthode pour l'estimation des paramètres de la superellipsoïde. La régularité de la répartition des données jouerait beaucoup moins sur le résultat et le critère (minimisation au sens des moindres carrés) est tout à fait adapté pour la suite de l'approximation. Mais l'adaptation à la 3D n'est pas immédiate. Le problème est la contrainte des paramètres. En effet, pour une quadrique, la formule permettant d'assurer que la solution définit bien une ellipsoïde n'est pas quadratique. Elle ne peut donc pas être exprimée sous la forme $\mathbf{a}^T C \mathbf{a}$. Le problème ne peut alors plus être résolu de manière simple avec une extraction de valeurs et vecteurs propres et perd beaucoup de son intérêt.

2.2 Minimisation de l'énergie

Maintenant que nous savons obtenir une première estimation des paramètres de la superellipsoïde, nous devons sélectionner un algorithme de minimisation. La méthode de régression devra être applicable à une fonction non-linéaire et pourra utiliser ses dérivées puisque nous pouvons les calculer à la fois numériquement et analytiquement.

L'algorithme utilisé est celui de Levenberg-Marquardt (LM). Bien que cela soit un classique, il a ses limites. Ainsi, certains essayèrent de réaliser cette minimisation différemment, et en particulier à l'aide d'algorithmes stochastiques (Downhill Simplex Method, algorithmes génétiques), que nous avons testés par ailleurs, et dont nous donnons une comparaison des performances et des différents résultats obtenus.

Levenberg-Marquardt (LM)

L'algorithme de LM est une méthode classique de régression non-linéaire [PTVF92, BCA95]. Il nécessite les dérivées du premier ordre de la fonction à minimiser. Cet algorithme va osciller entre deux méthodes de minimisation. La première est une simple descente de gradient, qui peut être formulée ainsi :

$$\mathbf{a}' = \mathbf{a} - C \nabla \Delta(\mathbf{a}) \tag{11}$$

avec \mathbf{a} un ensemble de onze paramètres définissant une superquadrique et C une constante définissant le pas de descente.

La seconde méthode est une descente quadratique, qui va supposer que, lorsque l'on est proche de la solution, la fonction à minimiser peut être approximée par une forme quadratique. Le minimum de l'énergie à minimiser est alors le minimum de la forme quadratique qui peut se calculer directement :

$$\mathbf{a}_{min} = \mathbf{a} + \mathbf{D}^{-1} \cdot [-\nabla\Delta(\mathbf{a})] \quad (12)$$

avec D la matrice Hessienne de $\Delta(\mathbf{a})$.

L'idée de l'algorithme de LM est de combiner ces approches en remarquant que quand \mathbf{D} est diagonale la méthode quadratique devient une simple descente de gradient. Si on pose alors \mathbf{D}' telle que :

$$\begin{aligned} \mathbf{D}'_{jj} &= \mathbf{D}_{jj} * (1 + \lambda) \\ \mathbf{D}'_{ij} &= \mathbf{D}_{ij} \quad (j \neq i) \end{aligned}$$

La formule permettant la régression devient :

$$\mathbf{a}_{min} = \mathbf{a} + \frac{1}{\lambda} \mathbf{D}'^{-1} \cdot [-\nabla\Delta(\mathbf{a})]$$

Ainsi, quand λ sera grand, la matrice D' sera presque diagonale, on sera proche d'une descente de gradient (Éq. 11), tandis que quand λ sera proche de 0, la régression se fera à la manière d'une descente quadratique (Éq. 12).

Si \mathbf{a} est un ensemble de paramètres de la fonction à minimiser à une étape donnée de la régression, l'étape suivante se déroule alors ainsi :

1. Calculer \mathbf{a}'
2. Si $\Delta(\mathbf{a}') > \Delta(\mathbf{a})$ alors $\lambda = 10 * \lambda$
3. Si $\Delta(\mathbf{a}') \leq \Delta(\mathbf{a})$ alors $\lambda = 0.1 * \lambda$ et $\mathbf{a} = \mathbf{a}'$.

On répète itérativement ce procédé jusqu'à ce que $\Delta(\mathbf{a})$ ne régresse plus, ce qui revient à s'arrêter quand λ est trop grand.

L'algorithme de LM converge généralement assez rapidement mais se laisse piéger par les minima locaux. Il est toujours possible, comme dans le cas du DSM de se sortir de ces minima locaux en ■ secouant ■ les paramètres après une première convergence puis de relancer la régression.

D'un point de vue plus général, en ne considérant pas seulement les exemples présentés dans cet article, mais l'ensemble des tests effectués durant notre travail [Che04], on peut dire que l'algorithme génétique et l'algorithme de LM donnent tous deux de bons résultats. Le premier a parfois l'avantage de donner de meilleurs résultats sur des temps de régressions plus longs que ceux que nous avons présentés ici et le second d'être beaucoup plus efficace dans les premières secondes de la minimisation. Par contre, un avantage de LM est de ne nécessiter aucun paramétrage, tandis qu'avant de pouvoir utiliser l'algorithme génétique, il faut fixer la taille de la population, le taux de mutations, le taux de croisement et divers autres paramètres qui sont venus compléter les méthodes stochastiques récentes.

3 Segmentation des données tridimensionnelles

Nous avons vu comment approximer un ensemble de points par une superellipsoïde, mais un objet 3D ne peut en général pas être décrit par une seule superellipsoïde. Il nous faut maintenant une méthode permettant de passer des données à une décomposition structurée en sous-ensembles de points représentables par une unique superellipsoïde. Bien sûr cette décomposition doit produire un nombre minimal de parties pour une distorsion donnée.

3.1 Local à global : croissance de région

Cette première approche de segmentation des données tridimensionnelles a été implémentée en parallèle à notre méthode originale que nous exposerons dans la partie suivante. Cette méthode est basée sur un algorithme de type croissance de régions et a été proposée par Leonardis [LJS97, LSM94]. Un travail d'adaptation pour passer des images de profondeur à nos données non-organisées et irrégulières a été effectué. La méthode se décompose en trois phases principales :

1. **Initialisation des germes** : correspond à la création d'un ensemble de germes, c'est-à-dire un ensemble de régions de l'espace contenant un nombre réduit de points tridimensionnels voisins. Chacune de ces régions doit pouvoir être approximée par une seule superellipsoïde avec une qualité donnée ;
2. **Phase de croissance** : durant cette phase, les germes vont tenter de grossir en incorporant des points de leur voisinage ;
3. **Phase de sélection** : cette étape est chargée d'éliminer les germes qui sont devenus trop redondants, qui décrivent une même partie de l'objet.

Après l'initialisation des germes, des alternances de phase de croissance et de sélection se succéderont jusqu'à ce que l'ensemble de régions soit stable.

La figure 7 montre le résultat obtenu sur un objet synthétique peu complexe, mais où la méthode segmentation ne parvient pas à obtenir le nombre attendu de primitives. En effet, dans le descripteur final 7c, le rectangle inférieur est représenté par deux descripteurs. Ainsi, les deux superellipsoïdes qui semblent être redondantes ne le sont en fait pas, car les sous-ensembles de points qu'elles représentent sont relativement disjoints, on ne s'appuie pas sur les superellipsoïdes pour effectuer la sélection mais sur les sous-ensembles de points qu'elles modélisent. En effet, le principal problème que nous avons rencontré avec cet algorithme se situe au niveau de la recherche de points à ajouter aux régions lors de cette phase de croissance (nos données sont irrégulières et non-organisées).

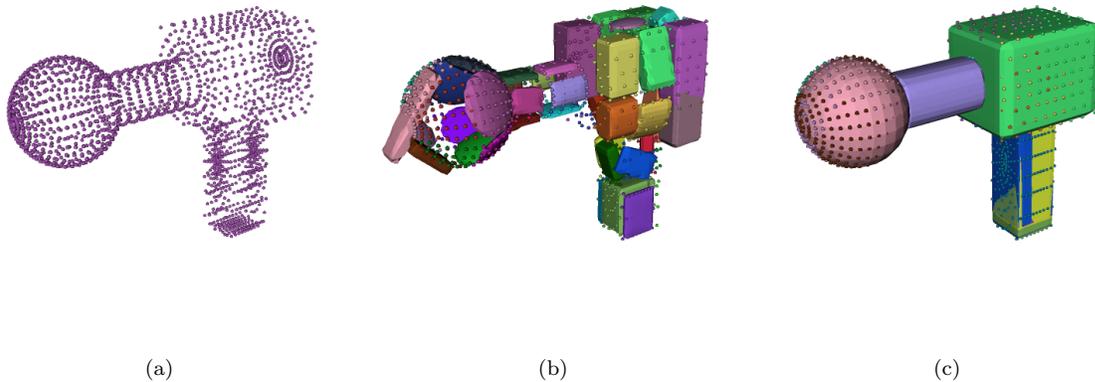


FIG. 7 – Résultat de la segmentation par croissance de régions. (a) Données 3D. (b) Les germes. (c) Le descripteur final.

Il est important de souligner un autre inconvénient intrinsèque à cette méthode. Nous ne pourrions jamais être sûrs que chaque partie de l'objet 3D est bien représentée par une superellipsoïde dans le descripteur final, pour la simple raison que la phase d'initialisation n'assure en aucun cas qu'il y ait au moins un germe dans chaque partie de l'objet. Si une partie n'est pas représentée dès le début, elle ne le sera sans doute pas non plus à la fin de la segmentation, puisque la croissance des germes est faite justement dans l'optique de ne pas absorber d'autres parties de l'objet.

Comme alternative à cet algorithme, et pour être mieux adapté à des données 3D plus générales, nous proposons une méthode originale basée sur une approche *global vers local* de type découpage-fusion.

3.2 Global à local : découpage-fusion (split and merge)

Le principe de la méthode découpage-fusion est classique, principalement en analyse d'image 2D. Comme son nom l'indique, il s'agit de la succession de deux étapes :

1. **Phase de découpage (split)** : les données vont être récursivement scindées en plusieurs parties suivant un critère d'homogénéité, jusqu'à ce que ce critère soit respecté pour toutes les régions ;
2. **Phase de fusion (merge)** : afin de réduire l'ensemble des trop nombreuses régions produites précédemment, certaines vont être regroupées sans pour autant que l'homogénéité des sous-parties ne s'en ressente ou que la distorsion du modèle aux données n'augmente significativement.

3.2.1 Phase de découpage (split)

Comme nous venons de l'évoquer, le but de cette étape est d'obtenir une partition des données, telle que chacune des sous-parties soit homogène suivant un critère. Mais, cette partition ne doit pas forcément être minimale puisque l'étape suivante va justement s'occuper de regrouper ces parties afin de rendre leur nombre minimal. En fait, soit on ne sait pas comment localiser directement les frontières des régions naturelles de l'objet¹, soit on ne peut pas les calculer simplement ou en un temps raisonnable. Nous allons donc, de manière récursive, scinder les données arbitrairement en un nombre donné de sous-ensembles en projetant que cette approche va forcément, à un niveau ou à un autre, isoler dans un de ces sous-ensembles un unique morceau de région naturelle de l'objet.

Ainsi, à la fin de cette étape, on obtient généralement une configuration remarquable :

1. des régions peu nombreuses et de taille importante au centre des parties naturelles de l'objet ;
2. des régions beaucoup plus nombreuses et de taille réduite dans les zones de frontières.

À la différence de l'algorithme de croissance de régions où il y avait un problème d'initialisation des germes, ici la précision est complètement guidée par les données. Si une partie d'objet est complexe beaucoup de régions seront produites localement pour celle-ci, mais ce ne sera pas forcément le cas dans les parties plus simples.

Si on revient à notre problème, nous devons donc déterminer deux choses. La première est une manière de découper les données et la seconde est un critère d'homogénéité permettant de stopper la scission.

Méthode de scission

Procédons par étapes pour la scission, en discutant d'abord le nombre de régions produites. Dans le cas d'un algorithme de découpage-fusion appliqué à de la segmentation d'image 3D, il est habituel que ce découpage se fasse en huit et ainsi obtenir une structure d'*octree*. Mais cela ne serait pas forcément judicieux dans notre cas. En effet, en coupant en huit sous-parties, les ensembles de points tridimensionnels à approximer seront la plupart du temps des surfaces très ■ ouvertes ■ ce qui rendrait d'autant plus difficile l'approximation et produirait aussi de très nombreuses régions. Pour notre objectif, nous choisirons donc de scinder les données en seulement deux parties, dans le but d'ouvrir les surfaces le moins possible. La méthode de découpage peut rester simple, relativement arbitraire. C'est l'association des étapes découpage et fusion qui se charge de déterminer les frontières complexes des parties de l'objet.

Ayant décidé de scinder les données en seulement deux parties, le plus simple est donc d'utiliser un plan de coupe. C'est-à-dire, pour une région donnée R et un plan P , les régions R_1 et R_2 résultantes de la scission de R par P seront formées de l'ensemble des points de R sur P ou à gauche de P pour la première, et strictement à droite de P pour la seconde. N'ayant pas d'indication sur comment placer ce plan ou comment l'orienter, nous utiliserons le plan le plus naturel, c'est-à-dire celui qui passe par le centre de gravité de l'ensemble des points tridimensionnels et est orthogonal à son axe principal (Fig. 8). Notons que ce plan de coupe est d'autant plus pertinent quand les données sont régulièrement réparties, ce qui n'est pas forcément le cas. La méthode du choix du plan de coupe est importante et mériterait peut être plus d'approfondissement. En effet, s'il n'est pas sensé altérer le résultat final de la segmentation, ce plan de coupe a une importance cruciale dans la progression de l'algorithme et du nombre d'étapes nécessaires pour aboutir à la segmentation. Cependant, nos données non-organisées et irrégulières ne nous laissent pas beaucoup d'options. Par exemple, les méthodes qui se basent sur l'analyse des courbures ne peuvent pas être envisagées si on ne veut pas passer, comme dans la méthode de croissance de région par l'étude du voisinage des points 3D.

Critères d'arrêt

Afin de stopper cette scission récursive du nuage de points, nous devons établir un critère d'arrêt.

¹Nous qualifierons de naturelles les régions ou parties de l'objet pouvant être approximées par une seule superellipsoïde

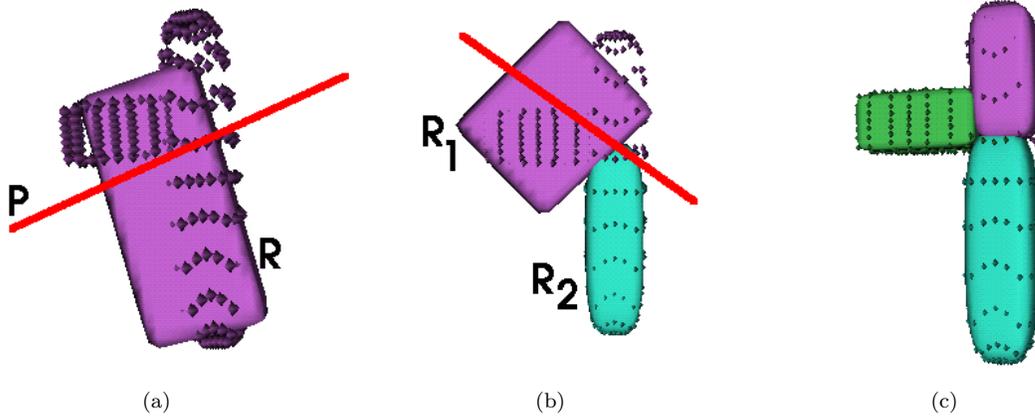


FIG. 8 – Exemple de phase de découpage

D'abord, il y a le critère de taille. Par expérience, on peut affirmer qu'un ensemble de seulement dix points n'est pas vraiment approximable par une superquadrique avec notre technique, cela ne sert à rien de descendre aussi bas (i.e. de découper).

Bien entendu, il y a aussi un critère d'homogénéité de la région. Ce critère va arrêter la phase de découpage pour une région avant que celle-ci n'ait atteint la taille minimum requise. Comme nous ne savons pas ce qui va se passer dans la seconde étape de l'algorithme (i.e. fusion), si telles ou telles régions vont fusionner ou pas, chaque partie que nous obtenons dans cette première étape doit être approximable par une superellipsoïde suivant le niveau de distorsion Δ désiré. Cela doit être pris en compte dans le critère d'homogénéité. En fait, on dira qu'une région, c'est-à-dire un ensemble de points tridimensionnels, est homogène s'il est possible de l'approximer par une superellipsoïde avec une tolérance donnée. Si la tolérance d'approximation est τ_s , et suivant la notation de l'équation 10, une région est homogène si et seulement si : $\Delta(\mathbf{a}) \leq \tau_s$

D'autres formes de critères pourraient être pris en compte, notamment la différence de qualité d'approximation d'une région R et de celle des régions résultantes de la scission de R : R_1 et R_2 . Une région serait dite homogène si les approximations des régions R_1 et R_2 ne sont pas meilleures que celle de R .

Algorithme de découpage

Ainsi, en prenant pour région initiale la totalité des données 3D, si une région donnée est approximable par une superellipsoïde répondant aux critères précédemment définis, la phase de découpage s'arrête, sinon la région est scindée en deux régions par le plan de coupe passant par son centre de gravité et orthogonal à son axe d'inertie, et le procédé est répété récursivement sur les deux régions résultantes (cf. algorithme).

Ensure: Scission(REGION R , REGION R_1 , REGION R_2)

$P = \text{PlanDeCoupe}(R)$

for all $p \in R$ **do**

if (AGauche(x, P)) **then**

$R_1 = R_1 + p$

else

$R_2 = R_2 + p$

end if

end for

Ensure: Split(REGION R)

if ((taille(R) ≥ 20) **et** ($\exists \mathbf{a}$ tel que $\Delta(\mathbf{a}) \leq \tau_s$)) **then**

 Scission(R, R_1, R_2);

 Split(R_1);

 Split(R_2);

end if

3.2.2 Phase de fusion (merge)

L'étape de découpage nous a permis d'obtenir un partitionnement, tel que chacune des parties peut être approximée avec précision par une seule superquadrique. Mais le nombre de régions créé est loin d'être optimal et de correspondre aux parties naturelles de l'objet traité. Cela vient du fait que le plan de coupe ne tient pas compte des frontières naturelles de cet objet. En revanche, toute région ne fait partie que d'une partie naturelle. L'étape de fusion va permettre de regrouper les régions qui appartiennent à la même partie naturelle mais qui ont été séparées.

C'est donc l'étape de découpage qui va déterminer la précision de la segmentation. En effet, durant l'étape de fusion, aucune nouvelle frontière entre les régions ne va apparaître, au contraire certaines vont disparaître. C'est donc le caractère du critère d'homogénéité de l'étape précédente (τ_s) qui déterminera la précision maximum de la segmentation obtenue par cette méthode.

Afin de mener à bien ces regroupements, nous avons besoin de deux choses. La première est un critère permettant de déterminer si deux régions méritent d'être regroupées, et la seconde est une stratégie nous indiquant quels couples de régions peuvent être candidats à la fusion (Fig. 9).

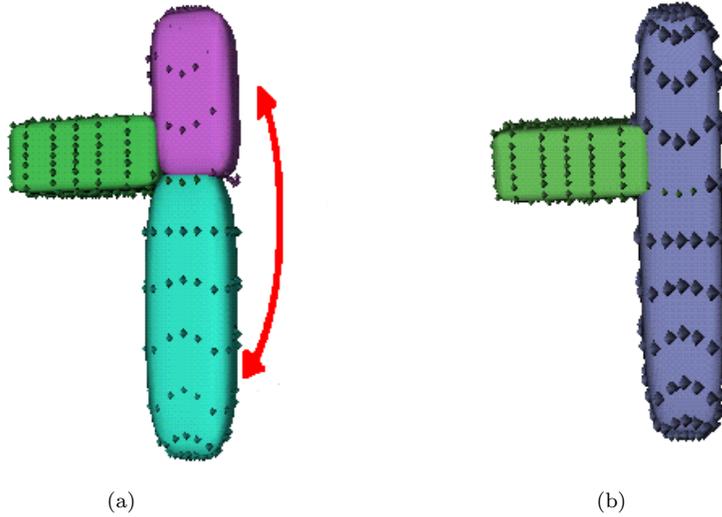


FIG. 9 – Exemple de phase de fusion

Critère de regroupement

Comme pour la phase précédente, certains des ensembles de points qui vont être créés dans cette étape par regroupement seront ceux qui formeront la segmentation finale. Nous devons donc être en capacité de représenter tous ces ensembles par une superellipsoïde pour un taux de distorsion donné. Cela doit faire partie du critère de regroupement. On ne peut regrouper deux régions que si pour l'union des ensembles de points qui les composent, il existe un ensemble \mathbf{a} de paramètres d'une superellipsoïde, tel que : $\Delta(\mathbf{a}) < \tau_m$. Les seuils de distorsions τ_s et τ_m utilisés respectivement dans la phase de découpage et celle de fusion sont idéalement les mêmes ($\tau_s = \tau_m$), puisqu'ils représentent tous deux la distorsion maximale désirée pour toute superellipsoïde du descripteur final.

Comme pour le critère d'homogénéité, une prise en compte des itérations précédentes est aussi envisageable. Ainsi le regroupement de deux régions R_1 et R_2 en une seule région R n'aurait lieu que si la qualité de modélisation de R est aussi bonne que celle R_1 et R_2 . Notons que ce critère ainsi défini pourrait autoriser moins de regroupement que nécessaire. Si par exemple, R_1 et R_2 sont modélisées presque parfaitement, il faudrait que R le soit aussi.

En plus de la qualité d’approximation, d’autres considérations doivent être faites dans le cas du critère de regroupement. Supposons que nous essayons de déterminer si deux tiges de même taille, parallèles entre elles ou étant jointes bout à bout et formant un angle droit, doivent être regroupées. Il est tout à fait possible que l’approximation, au sens où nous l’avons définie, de ces deux objets par une seule superellipsoïde soit de bonne qualité, puisque la surface d’une superellipsoïde de type ■ plaque ■ peut être relativement près des deux ensembles de points (Fig. 10). La seule chose que nous pouvons reprocher à ce genre d’approximation est qu’une grande partie de la superellipsoïde ne modélise aucune donnée. C’est pourquoi, dans le critère de regroupement, nous devons tenir compte de l’augmentation du volume lors du passage d’une modélisation par deux superellipsoïdes à une seule superellipsoïde.

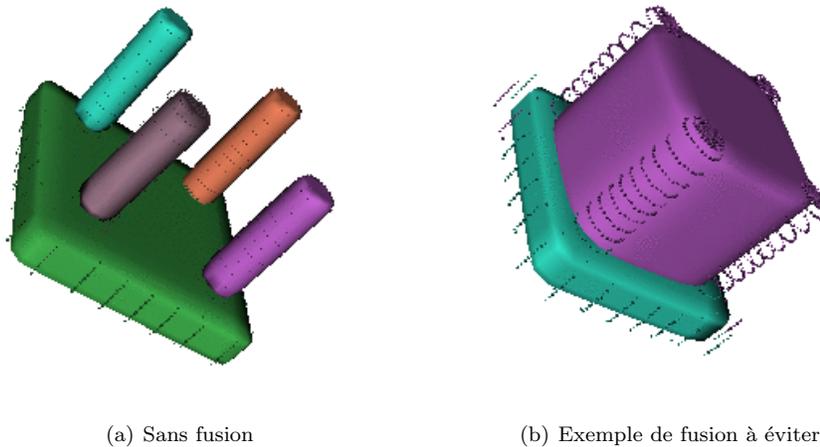


FIG. 10 – En complément du critère de distorsion, si l’augmentation du volume est trop importante, on ne fusionne pas.

Stratégie de recherche des couples candidats

Il n’est évidemment pas envisageable pour cette phase de fusion de tester tous les couples possibles de régions produites par l’étape de découpage. D’une part parce que ce serait trop long, et d’autre part, il n’y a pas de raison de faire ce test pour certains couples trop éloignés par exemple. La stratégie de regroupement est relativement primordiale puisque, pour garantir la compacité du modèle, il est essentiel que tous les couples faisant partie d’une région soient pris en compte, et d’autre part, l’ordre de sélection va privilégier certains regroupements plutôt que d’autres. Par exemple, un couple R_1R_2 est sélectionné. Ce couple satisfait le critère de regroupement, il y a donc fusion en une région R_3 . Il se peut alors que plus aucune région ne puisse être groupée avec R_3 , alors que si R_1 et R_2 n’avaient pas fusionné, certains regroupements auraient pu être faits par la suite avec R_1 ou R_2 .

Nous avons choisi de sélectionner le sous-ensemble de points qui est le mieux approximé par une superellipsoïde. Ensuite, nous testons la fusion de cette région avec l’ensemble de ses régions voisines et regroupons définitivement le couple de sous-ensembles le mieux modélisé. Nous ne considérons pas ici uniquement le voisinage direct des ensembles de points, mais les “voisins des voisins” sont aussi pris en compte, ce qui permet d’empêcher l’arrêt prématuré de la phase de fusion.

3.3 Résultats

Nous présentons ici les résultats obtenus grâce à la méthode de découpage-fusion, en utilisant la fonction de distance euclidienne radiale F_2 .

La figure 11 détaille la méthode de segmentation en douze étapes. Les étapes présentées ne sont pas exhaustives mais nous avons sélectionné les plus significatives suivant cette vue. La phase de découpage

s'étale de la figure 11b à la figure 11h, la suite présentant des étapes de fusion. Le descripteur final est constitué de trois superellipsoïdes comme le modèle synthétique original.

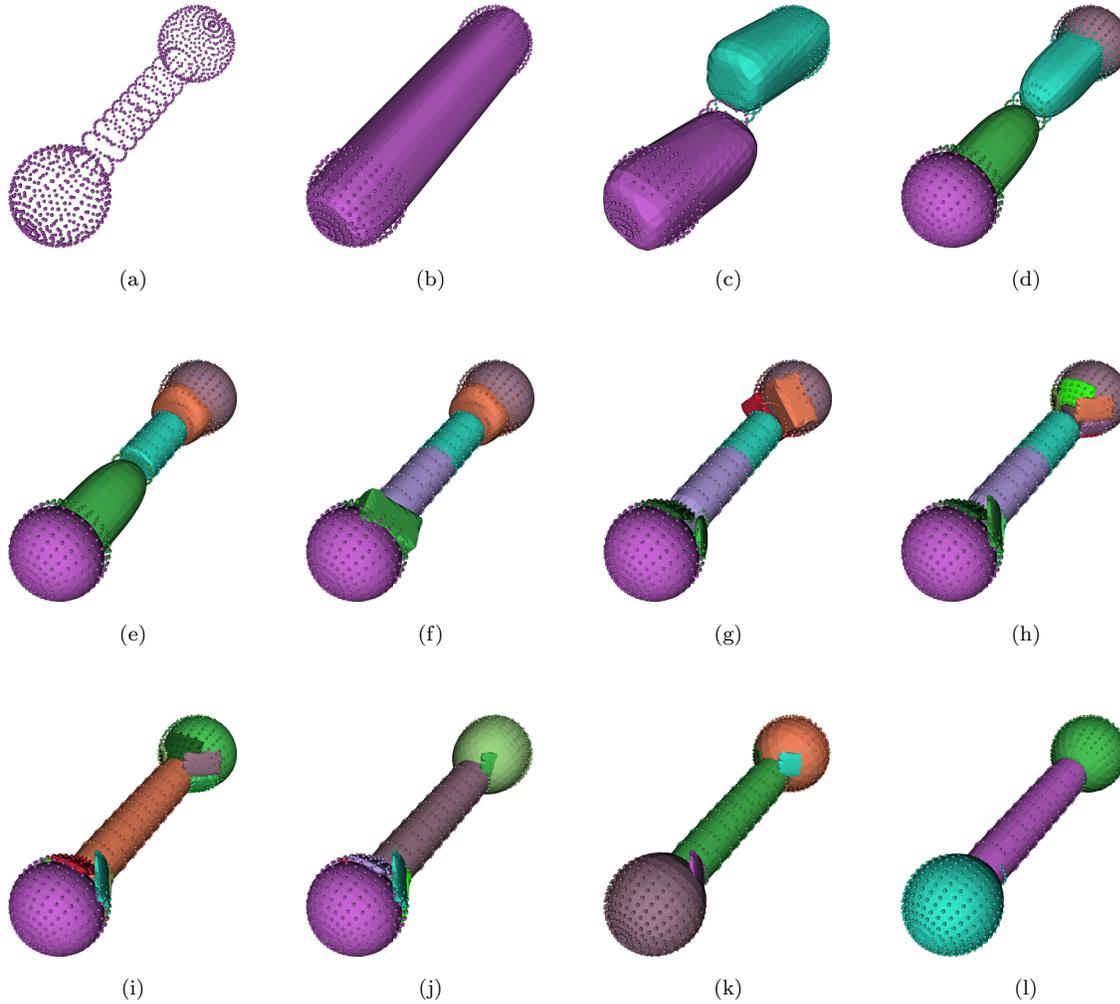


FIG. 11 – Étapes principales de la segmentation par la méthode de découpage-fusion. (a) l'ensemble de points 3D. (b)-(h) phase de découpage. (h)-(l) phase de fusion. (l) descripteur final

La figure 12 montre le résultat de cette nouvelle méthode sur un objet synthétique que nous avons déjà testé avec la méthode de croissance de régions (Fig. 7). Cette fois, la segmentation et le descripteur final est tout à fait conforme à ce qu'on attend.

Cette méthode de segmentation s'avère plus adaptée à notre type de données et nous a donc permis de décrire des objets plus complexes. La figure 13 montre le résultat obtenu sur un objet de type chaise, qui est issu d'un objet maillé que nous avons échantillonné. Comme on peut le voir, le résultat est globalement satisfaisant. Un problème persiste au niveau du dossier. Les barreaux qui le constituent ont fusionné pour donner deux plaques. Le problème de cette fusion aurait normalement du être pris en charge par le contrôle du volume que nous avons intégré. Il s'avère que, dans ce cas, les barreaux sont à la fois fins et rapprochés, ce qui les rend difficilement différenciables d'une simple plaque et le contrôle du volume nécessaire serait beaucoup trop strict pour que la segmentation aboutisse convenablement dans des cas plus généraux.

Malgré tout, les résultats sont probants la fois en terme de segmentation et de représentation pour la visualisation lorsqu'il s'agit de décrire des objets dont les parties sont très proches de superellipsoïdes (c'est le cas de tous objets précédemment cités). Les figures 14 et 15 montrent l'application de notre méthode sur des objets dont les parties principales sont plus complexes qu'une simple superellipsoïde.

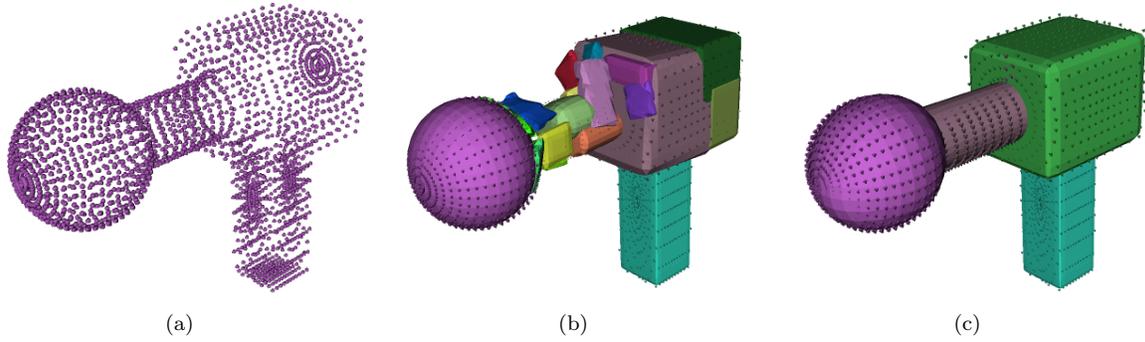


FIG. 12 – (a) Données 3D Pistolet. (b) à la fin du découpage. (c) descripteur final.

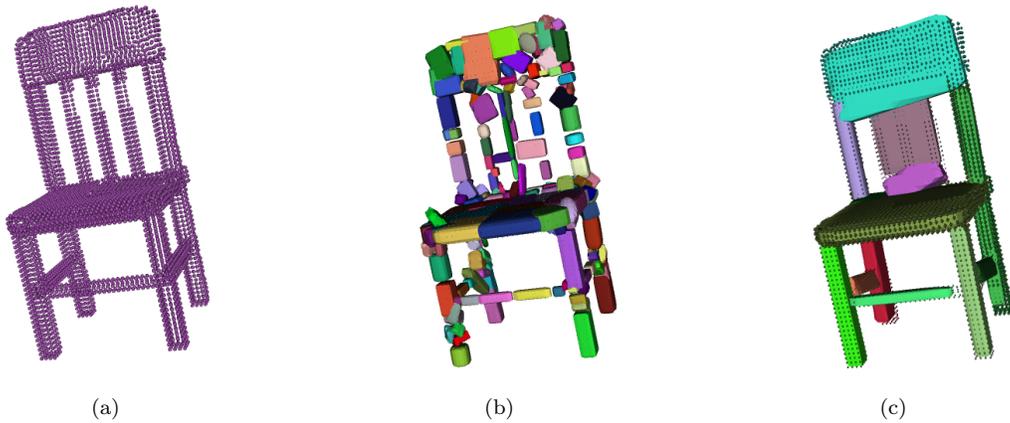


FIG. 13 – (a) Données 3D Chaise2. (b) à la fin du découpage. (c) descripteur final.

Comme on peut le voir, la segmentation est correcte et chaque partie principale de l'objet est représentée par une superellipsoïde et une seule. En revanche, en ce qui concerne la visualisation, on remarque une certaine limite de description des superellipsoïdes. En effet, la représentation est assez grossière. On ne peut pas accuser la méthode d'approximation : les sous-parties ne sont tout simplement pas modélisables avec une superellipsoïde de manière précise. Tout le problème est bien entendu de concilier compacité et information sémantique du descripteur d'une part, et précision de la représentation d'autre part. Dans ce travail, nous avons surtout privilégié la segmentation sur la représentation.

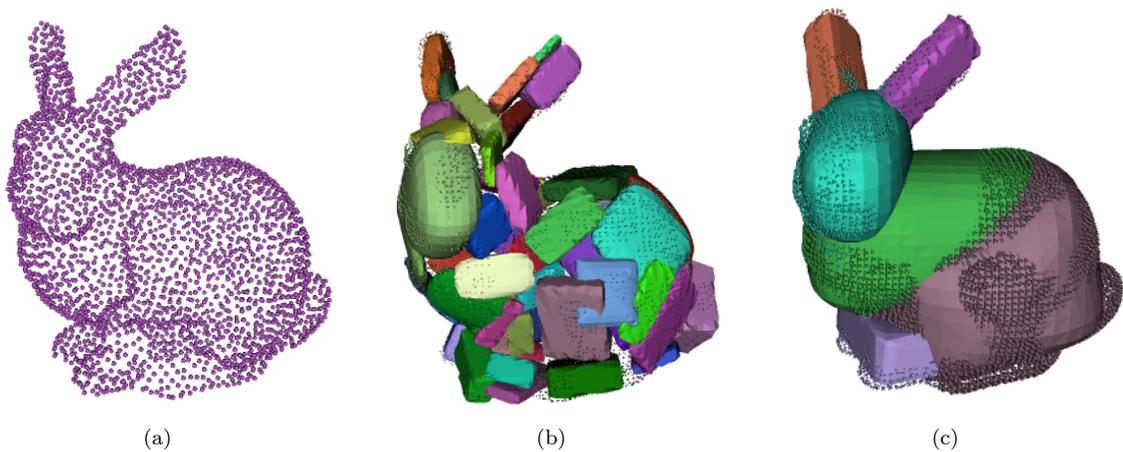


FIG. 14 – (a) Données 3D Bunny . (b) à la fin du découpage. (c) descripteur final.

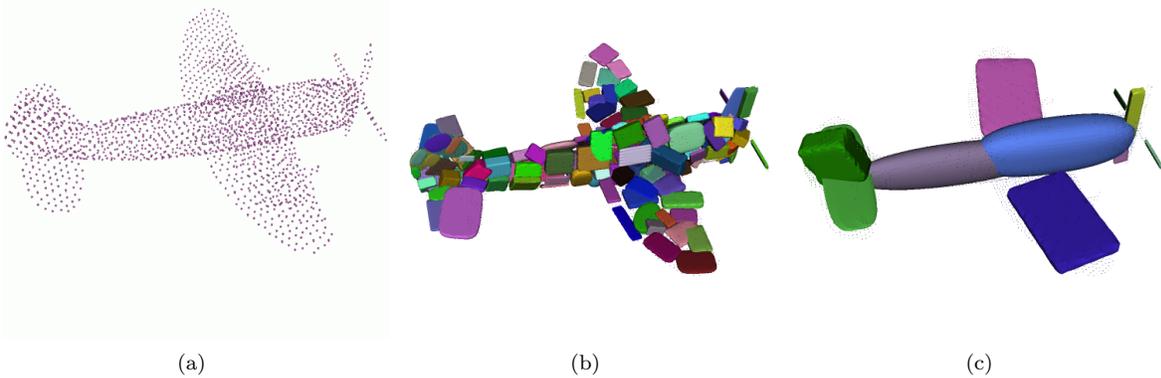


FIG. 15 – (a) Donnée 3D Avion. (b) à la fin du découpage. (c) descripteur final.

3.3.1 Structure du descripteur

À la différence de la méthode par *croissance de régions*, le descripteur final n'est pas uniquement un ensemble non-organisé de primitives. En effet, durant tout l'algorithme nous devons conserver les relations d'adjacences entre les superellipsoïdes afin de pouvoir, durant la seconde phase, procéder à la fusion de régions voisines le cas échéant. Ces relations topologiques sont toujours présentes dans le descripteur final. Ainsi, le résultat de cette méthode de segmentation est en fait un graphe dont les sommets sont des superellipsoïdes et les arcs indiquent une relation d'adjacence entre celles-ci (Fig. 16).

Cette structuration est bien sûr un avantage puisqu'elle charge le descripteur d'une information sémantique beaucoup plus importante qu'un ensemble non-organisé de superellipsoïdes. Il serait sans doute possible de déterminer ce graphe pour un descripteur issu de la méthode *croissance de région*, mais cela serait plus difficile surtout que dans ce cas-là les régions peuvent se chevaucher.

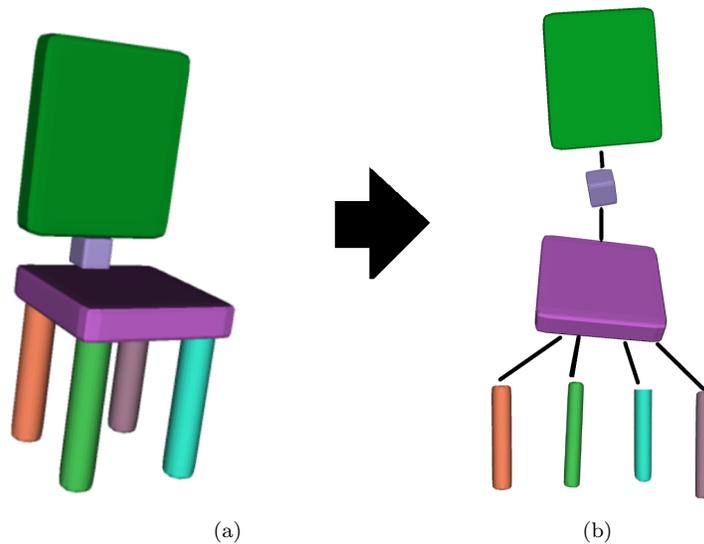


FIG. 16 – Le descripteur est structuré par un graphe d'adjacence.

4 Conclusion

Dans cet article, nous avons présenté deux méthodes de segmentation, ainsi que les outils nécessaires à leur mise en place. Si la première méthode, de type croissance de régions, n'a pas donné entière satisfaction, la seconde, de type découpage-fusion, a fourni de bons résultats au niveau de la segmentation des données tridimensionnelles. Dans cette approche, nous ne faisons référence à aucune connaissance *a priori* sur l'organisation ou la régularité de l'ensemble de points (tel que des relations de voisinage par exemple). Le descripteur que nous obtenons finalement est un ensemble de superellipsoïdes structuré par un graphe d'adjacence.

Au début de cette étude, nous visions un ensemble assez large de possibilités d'exploitation pour notre descripteur, nous pensions à des applications de type visualisation basse résolution, visualisation progressive ou multi-résolutions pour les réseaux bas débits, compression et comparaison d'objets pour une application à l'indexation. Malgré la capacité de description très importante des superellipsoïdes, relativement à leur faible nombre de paramètres, la pratique a montré qu'il était difficile de concilier à la fois un faible nombre de primitives et une distorsion peu élevée ou du moins une représentation suffisamment précise de la surface pour des applications de visualisation en moyenne et haute résolution. Certains auteurs ont proposé très tôt de déformer les superquadriques pour augmenter la variété des formes. Cela peut être des déformations globales [Bar84, PMY94, BCA95, ZK01], ou locales [TM91]. Dans tous les cas, on perd l'équivalence entre la formulation implicite et paramétrique, ce qui n'est pas toujours souhaitable.

Si, en l'état, le descripteur est donc limité pour les applications de type visualisation, il est en revanche porteur d'information sémantique forte. En plus, ce descripteur est très compact. Cela rend donc crédible son utilisation dans une application de type indexation d'objets 3D.

Références

- [Bar81] Alan H. Barr. Superquadrics and angle preserving transformations. *IEEE Computer Graphics and Applications*, 1(1) :11–23, January 1981.
- [Bar84] Alan H. Barr. Global and local deformations of solid primitives. *Computer Graphics*, 18(3), 1984.
- [BCA95] Eric Bardinet, Laurent D. Cohen, and Nicolas Ayache. A parametric deformable model to fit unstructured 3d data. Research report 2617 - INRIA, 1995.
- [BG87] T.E. Boult and A.D. Gross. Recovery of superquadrics from 3-D information. *SPIE Intell. Robots and Comp. Vision*, 1987.
- [Che04] L. Chevalier. *Modélisation et indexation d'objets 3D à l'aide de superellipsoïdes*. PhD thesis, Univ. Lyon1, 2004.
- [FPF99] Andrew W. Fitzgibbon, Maurizio Pilu, and Robert B. Fisher. Direct least square fitting of ellipses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5) :476–480, 1999.
- [JLS00] Ales Jaklic, Ales Leonardis, and Franc Solina. *Segmentation and Recovery of Superquadrics*, volume 20 of *Computational imaging and vision*. Kluwer, Dordrecht, 2000. ISBN 0-7923-6601-8.
- [LJS97] Ales Leonardis, Ales Jaklic, and Franc Solina. Superquadrics for segmenting and modeling range data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(11) :1289–1295, November 1997.
- [LSM94] 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.
- [Pen87] Alex Pentland. Recognition by parts. In *IEEE First Int. Conf. on Computer Vision*, pages 612–620, 1987. London.
- [PMY94] J. Park, D. Metaxas, and A. Young. Deformable models with parameter functions : application to heart-wall modeling. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, pages 437–442, June 1994.

- [PTVF92] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flanery. *Numerical Recipes in C*, chapter 15. Cambridge University Press, 1992.
- [SB90] 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*, 12(2), 1990.
- [Tau91] G. Taubin. Estimation of planar curves, surfaces and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(11) :1115–1138, 1991.
- [TM91] Demetri Terzopoulos and Dimitri Metaxas. Dynamic 3D models with local and global deformations : deformable superquadrics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(7) :703–714, July 1991.
- [YKY92] Naokazu Yokoya, Masahisa Kaneta, and Kazuhiko Yamamoto. Recovery of superquadric primitives from a range image using simulated annealing. In *Proceedings of International joint Conference on Pattern Recognition*, pages 168–172, 1992.
- [ZK01] Lin Zhou and Chandra Kambhamettu. Extending superquadrics with exponent functions : Modeling and reconstruction. *Graphical models*, 63(1) :1–20, January 2001.