

Extraction d'axes de préhension d'un objet de forme quelconque pour l'apprentissage par imitation de la saisie humaine

Sahar EL-KHOURY, Anis SAHBANI et Véronique PERDEREAU

Laboratoire des Instruments et Systèmes d'Île-de-France,

Université Pierre et Marie Curie, Paris 6, boîte 252

3 rue Galilée, 94200 Ivry-sur-seine, France

s.khoury@lisif.jussieu.fr, sahbani@ccr.jussieu.fr, vperd@ccr.jussieu.fr

Résumé : Cet article traite l'extraction d'axes de préhension d'un objet de forme quelconque pour l'apprentissage par imitation de la saisie d'objets par une main robotisée. Il comporte deux aspects essentiels. Tout d'abord, à partir d'une représentation 3D de l'objet considéré, notre méthode décompose les données tridimensionnelles en primitives géométriques simples dont les relations topologiques sont bien définies. Nous associons à chacune de ces primitives plusieurs axes de préhension. Un objet quelconque est alors représenté par un ensemble de formes géométriques simples ainsi qu'un ensemble d'axes de saisie. Ensuite, pour la réalisation de la prise, une phase d'apprentissage est requise. A partir de l'observation de la saisie humaine, un réseau de neurones sera entraîné afin de déterminer l'axe de préhension naturel associé à chaque objet.

L'originalité de l'approche est, en premier lieu, qu'elle propose une architecture complète pour la saisie d'objets de formes géométriques quelconques. En second lieu, elle représente les objets par des matrices de dimension bien déterminée à la différence des représentations par graphes trouvées dans la littérature. Cette représentation nous permettra, dans des travaux futurs, de réaliser l'apprentissage par imitation. Cet apprentissage servira à la détermination des axes de saisie en termes de fonctionnalité de l'objet, imitant ainsi le comportement humain.

Mots-clés : Intelligence artificielle, apprentissage par démonstration, préhension.

1 INTRODUCTION

La robotique a comme objectif le développement de robots autonomes. Par autonome, on sous-entend qu'un robot est capable de percevoir son environnement, de raisonner pour accomplir une mission et d'agir sur ce même environnement. La mission à laquelle nous nous intéressons, dans cet article, est la saisie d'objets inconnus. C'est l'une des fonctions les plus complexes à réaliser par un système robotisé à cause du grand nombre de degrés de liberté qui entrent en jeu. La prise initiale représente une phase capitale de la manipulation d'un objet et conditionne fortement le succès du travail demandé. Comme il existe plusieurs façons de saisir un objet, notre

but est de déterminer les différents axes de préhension qui lui correspondent selon son modèle géométrique. L'imitation du comportement humain permettra dans la suite d'entraîner un réseau de neurones afin de choisir la prise adéquate à la fonctionnalité de l'objet. Le travail présenté dans cet article traite l'étape préalable à l'apprentissage, le codage des objets. Ce codage servira comme entrée pour l'apprentissage.

Différentes approches ont été proposées dans la littérature pour la génération automatique des saisies en se basant sur le modèle géométrique de l'objet. Une méthode de détermination d'un axe caractérisant naturellement l'orientation de la main chez l'homme lors de la phase de saisie est proposée dans [Michel, 2004]. La méthode est fondée sur la minimisation d'un critère quadratique basé sur des propriétés géométriques du solide. Dans [Lopez-Damian, 2005], les auteurs proposent une démarche similaire mais s'appuyant sur une autre stratégie pour permettre à un robot de planifier son mouvement de saisie d'un objet non-convexe. En effet, l'approche s'appuie sur une décomposition itérative de l'objet à saisir en "composantes convexes approchées" [Lien, 2003]. En introduisant une mesure de la qualité de prise, l'algorithme permet de trouver la première partie non-convexe, saisissable, de l'objet considéré. Bien entendu, ces méthodes cherchent une prise possible de l'objet mais elle n'est pas forcément la meilleure du point de vue fonctionnalité de l'objet.

Pollard [Pollard, 2004], construit géométriquement une famille de prises à partir d'un seul exemple de saisie. En effet, son algorithme projette une prise donnée d'un objet sur d'autres objets de formes géométriques différentes. Elle passe ensuite par une étude de qualité de la prise générée. Borst propose [Borst, 1999] une méthode heuristique pour la planification de la prise. Il génère aléatoirement différentes prises de l'objet tout en connaissant son modèle géométrique 3D. Il passe ensuite par une étude de qualité de chacune des prises générées. Dans le même esprit, Miller [Miller, 2003] essaie de réduire le nombre de configurations possibles de la main robotisée lors de la saisie. Il décompose les objets en primitives simples, à savoir, sphères, cylindres, cuboïdes et cônes. Ensuite, il impose des règles pour l'ap-

proche de chacune des primitives et passe par une étude de qualité de la prise réalisée.

Pour éviter les calculs lourds liés à la qualité de la prise, plusieurs chercheurs utilisent l'apprentissage pour réaliser la saisie automatique. En ce qui concerne l'apprentissage par imitation, Fischer [Fischer, 1998] utilise un gant de données, gant truffé de capteurs permettant d'enregistrer en temps réel les positions de la main ainsi que son orientation. Ainsi, il est capable, après apprentissage, d'obtenir les positions des doigts d'une main robotisée qui correspondent à celle d'une main humaine. Ekvall [Ekvall, 2004] utilise, de même, un gant muni de capteurs puis fait le passage entre une main robotisée et une main humaine. Il ne cherche pas à déterminer les coordonnées des bouts des doigts mais plutôt à reconnaître le type de prise parmi quatre classes prédéfinies.

Aucune des méthodes présentes dans la littérature n'associe objet, forme et tâche pour générer une prise. Cet article propose une méthode permettant l'apprentissage des axes naturels de préhension d'objets complexes à partir de leurs formes géométriques. Notre approche est générale puisqu'elle s'applique aux objets de formes quelconques et génère l'axe naturel de préhension de ces objets. Comme la saisie d'un objet diffère selon ce qu'on veut faire avec, l'apprentissage sert à identifier le ou les axes de préhension correspondant à chacune des prises. Ces axes, dits naturels, caractérisent en effet l'orientation de la main de l'homme lors de la saisie d'un objet.

Ce document va aborder les différentes phases permettant l'apprentissage par imitation de la saisie humaine. La section 2 présente la problématique de l'apprentissage à réaliser. La section 3 définit les primitives géométriques utilisées pour la représentation des objets et la section 4 détaille leur codage.

2 VERS L'APPRENTISSAGE PAR IMITATION DE LA SAISIE HUMAINE

2.1 L'apprentissage par imitation

Notre intérêt se porte sur la détermination d'une prise adaptée à un objet donné, à partir de sa forme géométrique et des contraintes fixées par la tâche.

Or l'orientation de la main et le choix des points de contact sur le solide constituent deux aspects fortement liés. Par conséquent, les axes naturels de l'objet, au sens de la préhension, influent considérablement sur la prise effectivement retenue. Dans cette optique, et en considérant le fait qu'un objet peut être saisi de plusieurs façons, nous proposerons un algorithme permettant l'apprentissage des axes de prises adéquats à l'objet.

Notre apprentissage se fera par imitation. L'imitation désigne la correspondance entre le comportement du robot et celui de l'être humain lorsque cette correspondance résulte de l'observation par le premier du comportement du second. Afin de remplir cette condition, plusieurs personnes seront amenées à saisir différents objets. Un suivi du mouvement de la main [Menezes, 2005] permettra l'extraction de l'axe naturel de préhension. Le but de l'apprentissage sera alors d'identifier, à partir d'une représentation des objets par un ensemble de primitives

géométriques et d'axes de préhension, le ou les axes naturels caractérisant l'orientation de la main de l'homme lors de la saisie de ces objets.

2.2 Représentation des objets par un ensemble de primitives géométriques et d'axes de préhension

Nous voulons réaliser l'apprentissage par réseaux de neurones. Ce type d'apprentissage nécessite d'avoir en entrée un vecteur de dimension bien déterminée. En d'autres termes, nous avons besoin de modéliser nos objets par une matrice de dimension bien définie.

D'une part, nous nous inspirons de Miller [Miller, 2003] pour la décomposition des objets en primitives géométriques simples. Nous choisissons comme primitives les superquadriques grâce à leur compacité en terme de nombre de paramètres. En partant des données tridimensionnelles représentant l'objet, nous le segmentons en plusieurs composantes en exploitant ses propriétés locales telle que la courbure. Chaque composante est approximée par une superquadrique. L'objet est ainsi représenté par un ensemble de superquadriques.

D'autre part, nous déterminons les axes naturels de préhension de l'objet. Dans ce cadre, diverses stratégies de résolution sont envisageables. Nous citons [Yoshikawa, 1998, Kaneko, 1997] qui montrent que l'axe de préhension correspond à une droite la plus parallèle possible aux surfaces du solide. Dans la même optique, Michel et al. [Michel, 2004] trouvent un ensemble de faces opposées sur le solide pour déterminer un axe le plus parallèle à ces couples de faces. Dans [Lopez-Damian, 2005], les auteurs proposent un planificateur basé sur une décomposition itérative de l'objet à saisir en éléments convexes approchés dont ils calculent les axes d'inertie. Les approches décrites ci-dessus ne permettent pas de trouver tous les axes de préhension associés à un objet. Nous nous en inspirons, cependant, pour trouver les axes de préhension associés aux primitives géométriques simples constituant l'objet. En effet, le calcul d'axes d'inertie de l'objet en entier n'a aucun intérêt, d'où la nécessité d'une décomposition de l'objet en primitives simples. Le calcul de la matrice des moments centraux de chacune des primitives donne ensuite une représentation complète de l'objet en terme d'axes de préhension.

Cette approche permet ainsi de représenter l'objet par un ensemble structuré de primitives dont les relations topologiques sont bien définies. Une matrice de dimension $m \times n$, où n et m sont respectivement le nombre de primitives retenues et celles constituant les objets, est ensuite utilisée pour coder ces primitives ainsi que leurs positions relatives. Cette matrice, dont le calcul sera détaillé dans la section 4, servira comme entrée pour l'apprentissage réalisé.

3 LES SUPERQUADRIQUES, FORMES COMPACTES EN NOMBRE DE PARAMÈTRES

Les superquadriques sont une classe de surfaces paramétriques généralisant les quadriques. Elles ont été introduites en informatique graphique par Barr

[Barr, 1981]. Elles sont fréquemment utilisées en reconnaissance de formes et en reconstruction de surfaces à cause de leur simplicité et de leur compacité en terme de nombre de paramètres. Comme les quadriques, les superquadriques sont divisées en quatre classes : les superellipsoïdes, les superhyperboloïdes à une nappe, les superhyperboloïdes à deux nappes et les supertoroïdes. La première classe est la plus utilisée en informatique graphique car elle permet de modéliser, le plus naturellement, des objets tridimensionnels. Les superellipsoïdes peuvent être définies à la fois sous forme implicite et paramétrique.

3.1 La forme implicite

La surface d'une superellipsoïde est l'ensemble des points solution de $f(x, y, z) = 1$, avec :

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

où :

- $a_1, a_2, et a_3$, définissent le facteur d'échelle respectivement sur l'axe des x, y et z .
- ϵ_1 et ϵ_2 , représentent respectivement la courbure latitudinale et longitudinale de la superquadrique.

Cette forme a l'avantage de nous donner des informations sur la position d'un point par rapport à la surface de la superellipsoïde, ce qui nous sera utile pour l'approximation. En effet, elle permet le partitionnement de l'espace tridimensionnel en trois lieux. Ainsi, pour tout point P de coordonnées (x, y, z) :

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

3.2 La forme paramétrique

Bien que la forme implicite rende relativement simple le calcul de la distance d'un point dans l'espace à la superellipsoïde et ainsi l'approximation de la surface, elle est particulièrement lourde du point de vue affichage. La forme paramétrique a la particularité de rendre justement la visualisation très simple. La forme paramétrique des superellipsoïdes est obtenue grâce au produit sphérique de deux ellipses [Barr, 1981] :

$$\begin{cases} x(u, v) = a_1 \cos^{\epsilon_1} u \cos^{\epsilon_2} v & u \in \left[\frac{-\pi}{2}, \frac{\pi}{2} \right] \\ y(u, v) = a_2 \cos^{\epsilon_1} u \sin^{\epsilon_2} v & v \in \left[-\pi, \pi \right] \\ z(u, v) = a_3 \sin u \end{cases}$$

3.3 Les déformations des superquadriques

Grâce aux cinq paramètres $a_1, a_2, a_3, \epsilon_1$ et ϵ_2 , les superellipsoïdes, sous leur forme implicite ou paramétrique, nous permettent de modéliser un grand nombre de formes

élémentaires de toutes tailles. Ces formes peuvent aller de la simple ellipsoïde au parallélépipède rectangle en passant par le cylindre et bien sûr toutes les formes intermédiaires.

Au prix d'une augmentation de la complexité, il est possible d'ajouter des paramètres aux superellipsoïdes de façon à prendre en compte des déformations telles que l'étirement et le fléchissement :

- l'étirement sera décrit par les deux paramètres K_x et K_y tels que :

$$X = f_x(z)x \quad (2)$$

$$Y = f_y(z)y \quad (3)$$

$$Z = z \quad (4)$$

avec :

$$f_x(z) = \frac{K_x}{a_3} Z + 1 \quad (5)$$

$$f_y(z) = \frac{K_y}{a_3} Z + 1 \quad (6)$$

- le fléchissement est décrit par les deux paramètres k et α tels que :

$$X = x + \cos(\alpha)(R - r) \quad (7)$$

$$Y = y + \sin(\alpha)(R - r) \quad (8)$$

$$Z = \sin(\gamma)(k^{-1} - r) \quad (9)$$

avec :

$$\gamma = \arctan \frac{Z}{k^{-1} - R} \quad (10)$$

$$r = k^{-1} - \sqrt{Z^2 + (k^{-1} - R)^2} \quad (11)$$

$$R = \cos \left(\alpha - \arctan \frac{Y}{X} \right) \sqrt{X^2 + Y^2} \quad (12)$$

Précisons que X, Y et Z représentent les coordonnées de la superquadrique déformée alors que x, y et z représentent les coordonnées d'un modèle d'une superquadrique non-déformée dans un repère ayant pour origine le centre de gravité de celle-ci.

En appliquant ces déformations sur les trois superellipsoïdes simples, cylindre, cuboïde et ellipsoïde, nous obtenons sept primitives (cf. figure 1).

3.4 Les superquadriques utilisées

Nous rappelons que nous choisissons d'approximer les objets par des superquadriques vu leur compacité en terme de nombre de paramètres. Deux solutions sont alors possibles : soit nous déformons une superquadrique qui approxime l'ensemble des données tridimensionnelles [Bardinet, 1995], soit nous décomposons l'objet en éléments primitifs, chacun étant approximable par une superquadrique [Chevalier, 2003, Solina, 1990]. La première solution est l'approche qu'utilise Bardinet pour modéliser des données médicales. Il commence par

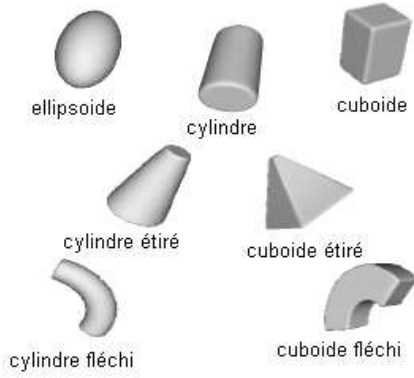


FIG. 1 – Les sept formes des superquadriques utilisées.

approximer ses données par une superquadrique non-déformée pour ensuite ajouter une "Free Form Deformation" (FFD) à la superquadrique trouvée. Cette approche ne nous convient pas. En effet, le modèle final de l'objet est décrit par onze paramètres définissant la superquadrique non-déformée et un nombre variable de points de contrôle nécessaires pour la FFD. Nous avons opté pour la deuxième méthode, la segmentation des données. Ainsi, un objet, de forme géométrique quelconque, sera représenté en utilisant les sept primitives [Wu, 1993] définies dans le paragraphe précédent.

4 CODAGE DES OBJETS

La méthode que nous utilisons pour le codage des objets est composée de trois phases : la segmentation du nuage de points 3D représentant l'objet, l'approximation de chaque partie obtenue par une superquadrique et la représentation de l'objet par un vecteur dans l'espace des superquadriques.

Dans ce qui suit, nous détaillons chacune de ces trois phases. Mais avant d'attaquer la procédure de codage, il faut se placer dans un repère lié à l'objet.

4.1 Redressement des objets

La représentation des objets doit être nécessairement invariante en translation et en rotation. Pour ceci, nous appliquons une transformation homogène T^{-1} aux points 3D représentant l'objet dans le repère monde et donnés par le vecteur X pour passer dans le repère R_o lié à l'objet [Rosenfeld, 1982, Horn, 1986]. T est constituée d'un vecteur de translation t et d'une matrice de rotation R . t n'est autre que le centre de gravité du nuage de points dans le référentiel monde. R peut être définie comme étant la matrice diagonalisant M , matrice de covariance d'ordre 2 :

$$M = E[(X - t)(X - t)^T] \quad (13)$$

Ainsi, R est constituée des vecteurs propres de M . Soit e_1 le vecteur propre correspondant à la valeur propre la plus petite, λ_1 . e_1 est aussi connu sous le nom d'axe principal. Une fois R et t calculés, nous pouvons appliquer au nuage de points 3D la transformation homogène

T^{-1} pour passer du repère monde R_m au repère objet R_o .

$$\begin{pmatrix} x_o \\ y_o \\ z_o \\ 1 \end{pmatrix} = T^{-1} \begin{pmatrix} x_m \\ y_m \\ z_m \\ 1 \end{pmatrix}$$

4.2 Segmentation de l'objet

La méthode utilisée pour la segmentation des données tridimensionnelles se base sur un algorithme en 4 étapes [Zhang, 2003]. Nous commençons par l'estimation de la courbure, puis la détection des frontières pour ensuite déterminer les différentes régions de l'objet et terminer par un post-traitement.

4.2.1 Estimation de la courbure et détection des frontières

Nos objets sont décrits par un nuage de points 3D. Nous estimons la courbure en chacun des points en utilisant la méthode décrite dans [Lin, 1982]. Cette méthode nécessite une triangulation des données tridimensionnelles. En effet, la courbure d'un point P (cf. figure 2) est donnée par la formule suivante :

$$k(p) = \frac{3(2\pi - \sum_i^N \theta_i)}{\sum_i^N A_i} \quad (14)$$

avec :

- N , nombre de triangles de sommet P .
- A_i , représente l'aire du triangle i de sommet P .
- θ_i , l'angle au sommet P du triangle i .

Après avoir calculé la courbure en tous les points représentant l'objet, nous les étiquetons. Les points de courbures négatives seront libellés comme frontières.

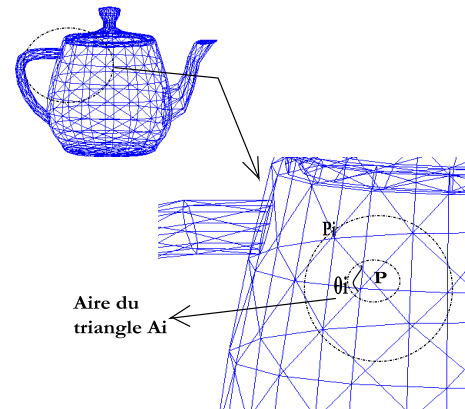


FIG. 2 – Calcul de la courbure d'un point P de la théière en utilisant les triangles voisins.

4.2.2 Segmentation

Dans cette étape, tous les points de l'objet sont déjà munis soit d'un label frontière soit d'un label non-frontière. La segmentation en régions de l'objet se fait de la façon suivante : en partant d'un point P libellé non-frontière, nous

affectons à P un numéro de région G. Ensuite, nous propageons cette étiquette aux voisins P_i de P, libellés non-frontières. Le même processus d'étiquetage s'applique aux voisins de P. La région G s'étend ainsi de proche en proche jusqu'à ce qu'elle soit entourée par des points libellés frontières. La même démarche itérative est utilisée pour détecter les différentes régions constituant l'objet et ceci en partant toujours d'un point étiqueté non-frontière qui n'a pas été affecté d'un numéro de région.

4.2.3 Post-traitement

A ce niveau de l'algorithme, tous les points libellés non-frontière sont affectés d'un numéro de région. Il nous reste à traiter les points frontières. Soit x , un point libellé frontière et soient x_i , les voisins de x . Le calcul de la distance euclidienne entre x et chacun des x_i permet de sélectionner, parmi ceux-ci, le point x_k le plus proche de x ayant un numéro de région. Le numéro de région de x_k sera attribué à x . Ainsi, chaque point sera muni d'un numéro de région. L'ensemble des régions trouvées constituent une partition de l'objet. La figure (3) représente le résultat obtenu pour la décomposition d'une théière. Chaque région de la théière est représentée par une couleur différente sur la figure.

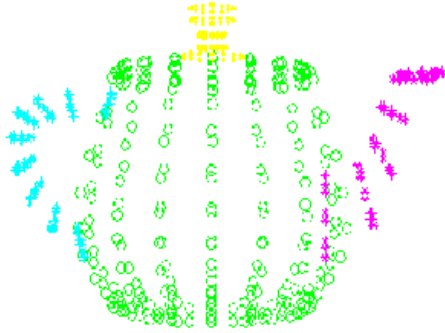


FIG. 3 – Décomposition d'une théière.

4.3 Approximation d'un nuage des points par une superquadrique

Nous avons comme entrée un nuage de points 3D obtenu suite à la décomposition de l'objet en différentes parties. Il s'agit d'approximer ce nuage de points par l'une des sept superellipsoïdes prédéfinies. En d'autres termes, il s'agit de déterminer les valeurs des paramètres du modèle de la superellipsoïde qui approxime le mieux notre nuage de points. L'estimation des paramètres du modèle nécessite la minimisation de la distance entre le nuage de points 3D et les modèles des superellipsoïdes [Solina, 1990]. Cette distance est une fonction non-linéaire. Nous utilisons l'algorithme déterministe de Levenberg-Marquardt [Madsen, 2004] qui est une méthode d'optimisation au sens des moindres carrés. La fonction exprimant la distance d d'un point à la surface d'une superquadrique est donnée par :

$$F(x, y, z) = f^{\epsilon_1} \quad (15)$$

$$d = F - 1 \quad (16)$$

Nous remarquons que pour le calcul de d , nous utilisons F au lieu de f . Ceci n'influe pas sur les paramètres de la superquadrique mais est indispensable pour la minimisation. En effet, nous voulons que les points situés à la même distance de la superquadrique aient la même valeur de d . Nous ne pouvons pas nous limiter à la simple utilisation de f (équation 1), sinon pour $\epsilon_1 \ll 1$, une petite déviation d'un point de la superquadrique selon l'axe des z serait largement amplifiée. D'autre part, l'unique considération de la distance des données au modèle n'est pas suffisante. Plusieurs superquadriques peuvent approximer le même nuage de points. Solina a montré, par expérimentation, que le volume de la superquadrique doit aussi être pris en compte. Cette prise en compte se fait en multipliant l'erreur d'approximation par le facteur $\sqrt{a_1 a_2 a_3}$ qui privilégie les petites superquadriques.

En tenant compte de tous les paramètres de la superquadrique, de sa position, de son orientation, et des deux types de déformations, regroupés dans $P_{sq} = \{a_1, a_2, a_3, \epsilon_1, \epsilon_2, t_x, t_y, t_z, r_x, r_y, r_z, k_x, k_y, k, \alpha\}$, l'approximation consiste à minimiser la fonction d'erreur suivante :

$$D = \sqrt{a_1 a_2 a_3} \sum_i^N (F(x_i, y_i, z_i; P_{sq}) - 1)^2 \quad (17)$$

Le problème de l'approximation par une superquadrique se ramène à un problème de minimisation sur 15 paramètres. L'algorithme de Levenberg-Marquardt permet une convergence rapide. Nous notons, toutefois, que cet algorithme a tendance à converger vers un minimum local si les paramètres du modèle ne sont pas bien initialisés. L'estimation des valeurs initiales se fait aisément. Les paramètres de translation (t_x, t_y, t_z) qui indiquent la position de la superquadrique sont donnés par le centre de gravité du nuage de points. L'orientation de la superquadrique (r_x, r_y, r_z) est calculée grâce à la matrice M de covariance d'ordre 2 (équation 13). La taille de la superquadrique (a_1, a_2, a_3) est initialisée avec la taille de la plus petite boîte englobant tous les points du nuage. ϵ_1 et ϵ_2 sont fixés à 1 pour partir d'une ellipsoïde. k_x, k_y et α sont initialisés à 0 et pour des raisons de singularité k est initialisée à une faible valeur et ceci pour partir d'un modèle non-déformé.

4.4 Représentation d'un objet 3D dans l'espace des superquadriques

Nous avons déjà précisé que nos objets seront modélisés par des superquadriques et un ensemble d'axes de préhension. Pour obtenir une représentation complète des objets, nous devons décrire les relations d'adjacence et éventuellement les positions relatives de ces différentes superquadriques. Les méthodes décrites dans la littérature représentent les objets par des graphes relationnels, où les noeuds représentent les surfaces de l'objet, et pour lesquels les arcs expriment les relations d'adjacence entre les surfaces. Des représentations par graphe d'adjacence

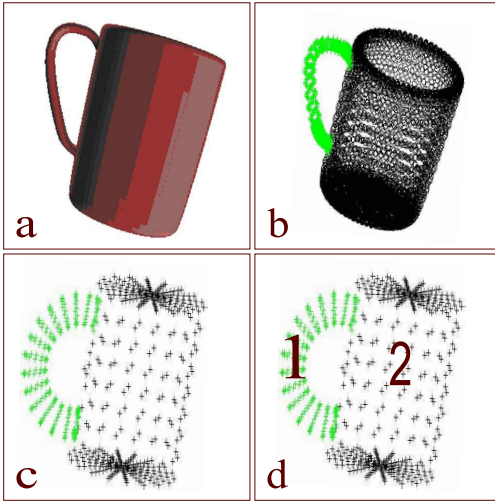


FIG. 6 – Etiquetage d’une tasse approximée par 2 superquadriques.

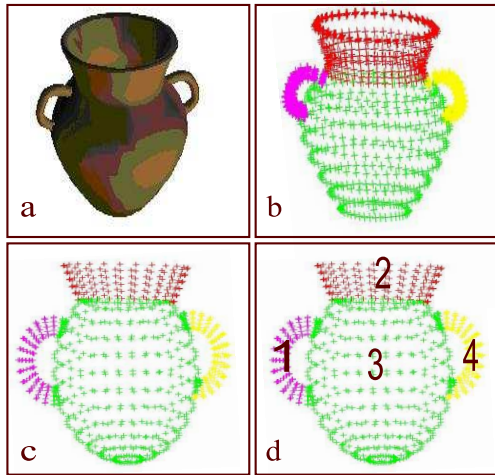


FIG. 7 – Etiquetage d’un vase approximé par 4 superquadriques.

$$\begin{array}{cccccc}
 cd & el & cb & cde & cdf & cbe & cbf \\
 \left(\begin{array}{cccccc}
 0 & 0 & 0 & 0 & T_{v1} & 0 & 0 \\
 0 & 0 & 0 & T_{v2} & 0 & 0 & 0 \\
 0 & T_{v3} & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & T_{v4} & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0
 \end{array} \right)
 \end{array}$$

Nous précisons que T_{t1} et T_{t2} sont les matrices homogènes représentant la position et l’orientation du cylindre et de l’anse dans le repère Rt_o associé à la tasse. Tandis que T_{v1}, T_{v2}, T_{v3} et T_{v4} sont les matrices homogènes représentant la position et l’orientation du cylindre, de l’ellipsoïde et des deux anses dans le repère Rv_o associé au vase.

5 CONCLUSION

Nous avons présenté dans cet article une architecture complète du codage des objets pour l’apprentissage par

imitation de la préhension humaine. Notre approche est générale puisqu’elle s’applique aux objets de toutes formes géométriques. Elle ne se limite pas à trouver une prise quelconque de l’objet. Grâce à une phase d’apprentissage, basée sur l’observation du comportement humain, notre architecture pourra identifier, pour un objet donné, les axes naturels de prise. L’apprentissage sera assuré par un réseau de neurones. Chaque objet est tout d’abord segmenté, puis approximé par un ensemble de superquadriques. Une matrice (l’entrée du réseau de neurones), codant chaque objet manipulé, est construite à partir de ces superquadriques ainsi que leurs dispositions relatives.

Nos travaux futurs se consacreront à l’optimisation des différentes techniques présentées dans cet article. Un accent particulier sera mis sur la phase d’apprentissage afin de choisir la structure du réseau de neurones la mieux adaptée à notre problème. Une comparaison entre les différentes méthodes d’apprentissage, à savoir, réseaux de neurones, règles de Bayes, arbres de décision, pourrait être envisagée. Pour terminer, la question reste de savoir si l’apprentissage par imitation de l’axe naturel de préhension se ramène effectivement à l’apprentissage de la tâche associée à un objet donné.

BIBLIOGRAPHIE

- [Bardinet, 1995] Bardinet E., Cohen L.D and Ayache N., A parametric deformable model to fit unstructured 3D data, rapport de recherche 2617-INRIA.
- [Barr, 1981] Barr A.H, Superquadrics and angle-preserving transformations, IEEE Comput. Graphics Applicat., vol.1, pp. 11-23.
- [Borst, 1999] Borst Ch., Fisher M. and Hirzinger G., A fast and robust grasp planner for arbitrary 3D objects, Proc. IEEE International Conference on Robotics and Automation, Detroit, Michigan.
- [Chevalier, 2003] Chevalier L., Jaillet F. and Baskurt A., Segmentation and superquadric modeling of 3D objects, journal of winter school of computer graphics, WSCG’03, Pilsen, Rep.Tchèque.
- [Dickinson, 1998] Dickinson S.J, Pentland A. and Stevenson S., Viewpoint-invariant indexing for content-based image retrieval, Proc. IEEE Int. Workshop on Content-Based Access of Image and Video Databse, pp. 20-30.
- [Dickinson, 1997] Dickinson S.J, Metaxas D. and Pentland A., The role of model-based segmentation in the recovery of volumetric parts from range data, IEEE Trans.On PAMI, Vol. 19, No. 3, pp. 259-267.
- [Ekvall, 2004] Ekvall S. and Kragic D., Interactive grasp learning based on humain demonstration, IEEE/RSJ International Conference on Robotics and Automation, New Orleans, USA.
- [Fischer, 1998] Fischer M., Van der Smagt P. and Hirzinger G., Learning techniques in a dataglove based telemanipulation system for the DLR Hand, Proc. IEEE International Conference on Robotics and Automation.

- [Horn, 1986] Horn B.K.P, Robot Vision, Cambridge, MA :M.I.T. Press.
- [Kaneko, 1997] Kaneko M., Hino Y., Tsuji T., On Three Phases for Achieving Enveloping Grasps, Proceedings of the IEEE International Conference on Robotics and Automation, Albuquerque, USA.
- [Lien, 2003] Lien J.M and Amato N., Approximate convex decomposition, technical report, TR03-001, PARASOL LAB, Texas A&M University.
- [Lin, 1982] Lin C. and Perry M., Shape description using surface triangulation, conf. Computer Vision : Representation and Control, pp. 38-43.
- [Lopez-Damian, 2005] Lopez-Damian E., Sidobre D. and Alami R., Grasp planning for non-convex objects, 36th International Symposium on Robotics, ISR, Tokyo, Japan.
- [Madsen, 2004] Madsen K., Nielsen H. and Tingleff O., Methods for non-linear least squares problems, Technical University of Denmark.
- [Menezes, 2005] Menezes P., Lerasle F., Dias J. and Chatila R., Single camera-based tracking of 3d gestures, International Conference on Robotics and Applications, Cambridge, USA.
- [Michel, 2004] Michel C., Rimond C., Perdereau V. and Drouin M., A robotic grasping planner based on the natural grasping axis, Proceedings of the International Conference on Intelligent Manipulation and Grasping, Genoa, Italy.
- [Miller, 2003] Miller A.T, Knoop S., H.I.Chritensen and P.K.Allen, Automatic grasp planning using shape primitives, Proc. IEEE International Conference on Robotics and Automation, Taipei, Taiwan.
- [Pollard, 2004] Pollard N.S, Closure and quality equivalence for efficient synthesis of grasps from examples, International Journal of Robotics Research, 23(6), 595-614.
- [Rosenfeld, 1982] Rosenfeld A. and Kak A., Digital picture processing, Orlando, FL :Academic.
- [Solina, 1990] Solina F. and Bajcsy R., Recovery of parametric models from range images : the case of superquadrics with global deformations, IEEE Transactions on Pattern Analysis and Machine Intelligence, 12(2), pp. 131-147.
- [Wu, 1993] Wu K. and Levine M.D, 3D object representation using parametric geons, technical report CIM-93-13, Center for Intelligent Machines, McGill University, Montreal, Quebec, Canada.
- [Yoshikawa, 1998] Yoshikawa T., Virtual Truss Model for Characterization of Internal Forces for Multiple Finger Grasps, Proceedings of the IEEE International Conference on Robotics and Automation, Leuven Belgium.
- [Zhang, 2003] Zhang Y., Koschan A. and Abidi M., Superquadrics based 3D objects representation of automotive parts utilizing part decomposition, Proc. of SPIE 6th International Conference on Quality Control by Artificial Vision, TN, Vol. 5132, pp. 241-251, Gatinburg.