



**HAL**  
open science

# Modélisation d'Objets 3D en vue de leur reconnaissance et leur manipulation par un robot personnel

Nizar Sallem, Michel Devy

## ► To cite this version:

Nizar Sallem, Michel Devy. Modélisation d'Objets 3D en vue de leur reconnaissance et leur manipulation par un robot personnel. ORASIS'09 - Congrès des jeunes chercheurs en vision par ordinateur, Jun 2009, Trégastel, France, France. inria-00404645

**HAL Id: inria-00404645**

**<https://inria.hal.science/inria-00404645>**

Submitted on 16 Jul 2009

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Modélisation d'Objets 3D en vue de leur reconnaissance et leur manipulation par un robot personnel

## Modelling of 3D objects for their recognition and manipulation by a companion robot

Nizar SALLEM<sup>1,2</sup>

Michel DEVY<sup>1,2</sup>

<sup>1</sup> CNRS ; LAAS ; 7, Avenue du Colonel Roche, F-31077 Toulouse, France

<sup>2</sup> Université de Toulouse ; UPS, INSA, INP, ISAE ; LAAS ; F-31077 Toulouse, France  
mail : nksallem@laas.fr, michel@laas.fr

### Résumé

*Ce papier décrit les fonctions visuelles à intégrer sur un robot personnel qui doit manipuler des objets 3D : il est focalisé sur l'apprentissage des représentations géométriques et photométriques de ces objets. Notre méthode est fondée sur l'extraction et la mise en correspondance de points d'intérêt depuis des images acquises par une seule caméra déplacée autour de l'objet en phase d'apprentissage. Tout d'abord, l'objet est segmenté dans les images en supposant le fond statique ou uniforme ; puis, à partir de pixels extraits et appariés dans des images acquises depuis plusieurs points de vue, un ensemble de points 3D est reconstruit, d'abord par une simple triangulation, puis par un ajustement de faisceaux ; à chaque point 3D est associé l'ensemble des signatures des points d'intérêt qui sont ses projections dans les images d'apprentissage ; un maillage est construit à partir de ces points 3D. Ces représentations seront ensuite exploitées pour reconnaître, localiser et manipuler ces objets.*

### Mots Clef

Ajustement de faisceaux, reconstruction 3D, nuage de points, localisation, reconnaissance, modèle géométrique, modèle d'apparence, points d'intérêt.

### Abstract

*This article describes visual functions to be integrated on a companion robot which has to manipulate 3D objects : we focus on the learning of geometrical and appearance-based representations. Our method is based on the extraction and the matching of interest points from images acquired by a camera during motions around the object to be learnt. At first, the object is segmented in these images, assuming a static and uniform background ; then from interest points matched between images acquired from different view points, a set of 3D points is reconstructed, initially by a simple triangulation, and then by a bundle adjustment ; each 3D point is memorized with the signatures (typically SIFT) of interest points from which it has been computed ; a triangular mesh is built from this set of 3D points. These*

*object representations will be used in order to recognize, locate and manipulate the learnt 3D objects.*

### Keywords

Bundle adjustmet, 3D surface reconstruction, points cloud, localization, recognition, geometric model, appearance model, interest features.

## 1 Introduction

La robotique de manipulation est sortie des usines d'assemblage pour s'immiscer dans des environnements plus humanisés. Contrairement aux robots industriels, qui exécutent des manipulations à poste fixe, dans des environnements complètement contrôlés pour faciliter la réussite des tâches robotisées, les robots compagnons assistent les humains dans des environnements non instrumentés, typiquement dans leurs domiciles ; nos travaux seront appliqués à terme sur un robot d'assistance aux personnes handicapées, dans leurs lieux de vie, environnements difficilement adaptables. Pour y évoluer et y exécuter des tâches, notre robot doit être doté de capacités d'apprentissage pour acquérir les représentations nécessaires, et d'adaptation pour s'accommoder des aléas qui surviendront lors de l'exécution des tâches.

Dans un tel contexte, la machine doit être équipée d'un certain nombre de capteurs qui lui permettent de reconnaître le monde qui l'entoure, de s'y déplacer en sécurité, d'éviter les obstacles, de manipuler des objets... Nous présentons dans cet article la solution que nous avons développée afin qu'un robot puisse apprendre les représentations des objets qui l'entourent afin les manipuler, puis de les reconnaître et les localiser lorsqu'ils seront de nouveau perçus.

Ce papier présente uniquement cette phase d'apprentissage du modèle des objets. La section 2 présente un état de l'art rapide sur la modélisation d'objets 3D en Robotique, et justifie nos principaux choix, fruits d'expériences passées dans notre équipe. La section 3 décrit notre plateforme expérimentale. En section 4 nous présentons l'essentiel de notre contribution actuelle, portant sur la modélisation d'objets 3D depuis un ensemble de points 3D éparés.

La section 5 évoquera seulement comment cette représentation sera ensuite exploitée pour reconnaître et localiser les objets appris. Enfin en section 6, nous tirerons quelques conclusions de ces travaux et évoquerons des perspectives déjà en cours.

## 2 Modélisation d'Objets 3D en Robotique

Si la reconstruction d'objet en 3D a débuté dans les années 70 avec les travaux de Shirai et Suwa, ce fut Baker qui en 1977 statua que "une vision effective requière une modélisation 3D flexible et indépendante du domaine". Depuis et grâce au développement des capteurs, ces techniques ont bien évolué. En 1992, Stenstrom et Connolly dans [17], introduisaient la reconstruction géométrique d'un objet *Facette-arête-sommet* directement à partir des images issues d'un laser. Ponce *et al.* en [16] reconstruisent un objet en le faisant tourner devant une caméra en ce servant de la calibration de Tsai. La surface de l'objet est approximés par des B-splines. Kragic *et al.* dans [7] présente un système de vision pour les tâches de manipulation robotique dans des environnements domestiques naturels. A cet effet l'auteur utilise deux paires de caméras stereo : une caméra périphérique et une fovéale. Selon le contexte, l'une ou l'autre de ces caméras sera utilisé pour modéliser l'objet.

Comme [16] et [7], nous pensons qu'un objet peut être bien approximé moyennant un nombre suffisant d'images, connaissant le déplacement de la caméra et tirant partie des multi-vues. Dans des travaux précédents de notre équipe, M.Cottret [5] a proposé une stratégie autonome d'apprentissage des objets présents dans l'environnement, en exploitant comme D.Kragic, des mécanismes pré-attentifs (vision panoramique) et attentifs (vision focalisée). Au fil de ses déplacements, le robot détecte des zones saillantes, retient celles qui peuvent correspondre à des objets d'intérêt, et modélisent l'apparence de ces objets afin de les reconnaître lors de prochaines perceptions. Nos travaux s'inscrivent également dans cette problématique ; notre robot détecte un objet isolé sur une table, accoste la table et apprend son modèle.

Initialement nous avons exploité un capteur stéréo pour acquérir des données 3D denses, depuis plusieurs points de vue sélectionnés alors que le capteur se déplace autour de l'objet [18]. Du fait du bruit des données stéréo, le recalage des images 3D acquises autour de l'objet, s'est révélé délicat ; le modèle obtenu finalement, est trop grossier, de sorte que la planification des actions de manipulation devient trop difficile. Ici nous proposons donc une approche purement monoculaire afin de reconstruire un ensemble de points épars sur la surface de l'objet : les représentations géométrique et photométrique de l'objet seront construites à partir de ces points.

## 3 Plateforme expérimentale

Les travaux réalisés dans ce contexte ont été portés sur le robot manipulateur du LAAS. Cet agent nous a permis de valider les algorithmes développés et les stratégies choisies dans le cadre de ces travaux. Nous présentons dans cette section l'environnement matériel et logiciel de ce robot.

### 3.1 Matériel

La plateforme expérimentale est une Neobotix Platform MP-L655 équipée d'un bras Mitsubishi Arm PA10-6C pour les manipulations appelé Jido (voir figure 1). Jido possède deux lasers SICK, un capteur d'effort JR3 et une pince à 3 doigts et 2 mors parallèles en guise de main. La partie vision est composée d'une platine pan-tilt-zoom montée sur un mat plus une paire de caméras fixée sur la main. Les caméras sont des PointGrey Flea couleur.



FIGURE 1 – La plateforme expérimentale Jido

### 3.2 Logiciel

Le système logiciel tournant sur l'agent est découpé en trois niveaux.

- supervision ;
- planification ;
- exécution.

La vision est un sous ensemble de l'exécution. A la base, un module d'acquisition permet la capture d'images et de vidéos depuis les différents bancs de caméras. L'ensemble des algorithmes développés dans le groupe, est contenu dans des libraries. La communication entre eux se fait par messages.

## 4 Modélisation 3D : des images au maillage

La modélisation 3D constitue l'apprentissage de l'objet. En entrée, les images de la scène ; à la sortie un modèle 3D de la surface de l'objet appris. Ce modèle servira aux autres modules du robot pour réaliser des tâches telle que la pré-

hension. La figure 2 décrit toute la chaîne. En noir les différentes fonctions des algorithmes développés et en bleu les entrées/sorties de ces blocs.

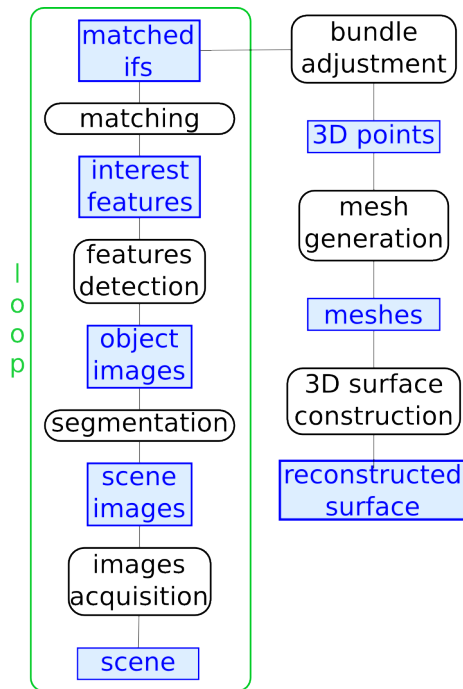


FIGURE 2 – Processus de vision

Les sections suivantes font le détail de ses algorithmes.

#### 4.1 Segmentation

La segmentation est l'opération de traitement d'image qui permet de découper l'image en régions d'intérêt sémantique. L'homme est capable naturellement de différencier les objets d'une scène grâce à un mécanisme alliant à la fois vision, compréhension et assimilation. L'élaboration informatique d'un tel algorithme, à un niveau aussi haut, demeure un des défis de la communauté vision. Dans la littérature, les différentes techniques de segmentation peuvent être groupées en quatre classes.

1. Segmentation basée sur les régions : croissance de région, décomposition/fusion ...
2. Segmentation basée sur les contours : segmentation par contours actifs ...
3. Segmentation par classification ou seuillage : segmentation par seuillage adaptatif ...
4. Segmentation basée sur la composition des trois premières familles.

Deux approches ont été proposées. La première, relativement simple, est la segmentation par seuillage adaptatif. Nous désignons une table comme *table d'apprentissage* et nous prenons un certain nombre d'images avant de mettre dessus l'objet à apprendre et de reprendre une nouvelle série de photos à partir des mêmes positions. Ces positions sont récupérées à partir de la discrétisation d'une demi sphère

de Gauss dont le centre coïncide avec le barycentre de l'objet (voir figure 3). L'image de l'objet est calculée à partir

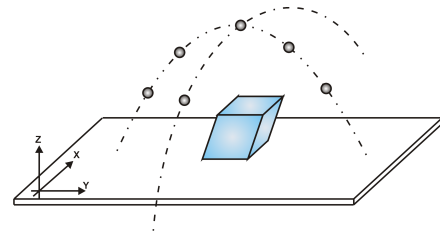


FIGURE 3 – Semi sphère discrétisée

de la différence entre une image de la scène avec l'objet et une image de la scène sans l'objet : moyennant les opérateurs morphologiques nous obtenons un masque binaire que nous appliquons à l'image (b) pour enlever le fond. Quoique naive, cette méthode s'avère être assez robuste et sans pertes importantes d'information.

Soit  $X$  l'image de la scène avec objet,  $Y$  celle de la scène vide,  $Z$  celle de l'objet et  $L$  le masque à calculer.

$$L' = X - Y$$

$$L = L' \oplus B \text{ où } B, \text{ l'élément structurant, est un disque}$$

$$Z = X \cap L$$

de rayon  $r$  tel que  $r = \frac{d}{2}$  où  $d$  est la distance maximale entre les éléments connexes de  $L$ .

La figure 4 montre des images de la scène sans l'objet (a), de la scène avec l'objet (b) et le résultat de la segmentation (c). Cette opération est répétée sur l'ensemble des posi-

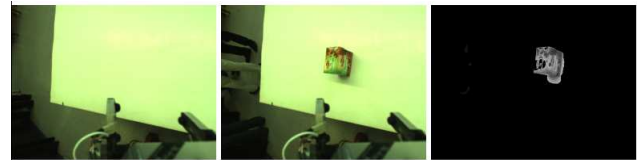


FIGURE 4 – Segmentation a. scène vide b. scène avec objet c. résultat

tions de la caméra. La sortie est un ensemble de points de vues de l'objet.

La seconde approche est basée sur les contours actifs. L'hypothèse est que le fond est uniforme. Un contour initial (figure 5 gauche) est défini autour de la position connue de l'objet ; un algorithme *snake* classique basé sur GVF est exécuté pour converger sur la silhouette de l'objet (figure 5 droite). Les points d'intérêt ne sont extraits que dans cette silhouette. Cette méthode est moins robuste parce que très sensible au bruit dans la détection de contours.

#### 4.2 Détection des points d'intérêt

La littérature est abondante en détection de points d'intérêt à partir d'images. Ces travaux ont été ponctués par Moravec(1981) [14], Harris et Stephens (1988) [6], Lowe(1999) [13] et Herbert, Tuytelaars et Van Gool (2006) [1]. Tous traitent le problème de trouver des points d'intérêt d'une façon objective et automatisée suffisamment robustes pour

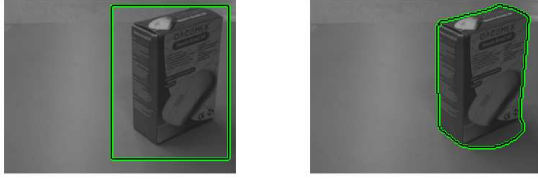


FIGURE 5 – Segmentation d'un objet par un snake : positions initiales et finales.

l'appariement et le suivi. Le plus utilisé est le détecteur de coins et de contours de Harris basé sur la matrice de moments de second ordre. Les SIFT introduits par le troisième sont invariants au changement d'échelle et se calculent à partir de la différence de gaussiennes à différentes échelles. Les SURF proposés par les derniers sont un bon compromis entre robustesse au changement d'échelle et temps de calcul. Ils ont comme support les réponses des ondelettes 2D de Haar. La stratégie adoptée a été d'inclure ces trois détecteurs dans nos algorithmes. En effet, le détecteur de Harris se prête mieux au calcul en temps réel mais le détecteur de SIFT reste le plus performant en mode hors ligne. Sur la figure 6, nous montrons le résultat du détecteur de Harris avec les paramètres suivant : facteur d'échelle = 1.0,  $\sigma_{derivation} = 1.0$  et  $\sigma_{convolution} = 2.0$ .

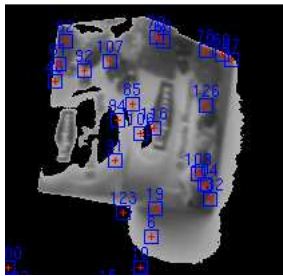


FIGURE 6 – Points d'intérêt détecté par Harris

### 4.3 Appariement des points d'intérêt

L'appariement est effectué par les *appariements de groupe de points d'intérêt* [8] une combinaison de contraintes de similarité géométrique et signalétique. Les ensembles de points d'intérêts  $\{x\}$  et  $\{x'\}$  détectés respectivement sur les images  $I$  et  $I'$  sont organisés en groupes locaux formés par un point pivot  $g_0$  et ses  $n$  plus proches voisins  $\{g_1, \dots, g_n\}$ .  $n$  doit être assez petit pour ne couvrir qu'une petite partie de l'image. Un seuillage maximum de la distance entre le pivot et son voisinage garantit cette contrainte.

Etant donné deux ensemble de groupes  $\mathbf{G} = \{G_1, \dots, G_n\}$  et  $\mathbf{G}' = \{G'_1, \dots, G'_n\}$  tels que  $G_i = \{g_1, \dots, g_n\}$ ,  $g_0 = x_i$ , l'hypothèse d'appariement  $H(G_i, G'_j)$  est définie par :

- une rotation  $\theta$
- un ensemble  $M(G_i, G'_j)$  de rotation  $\theta$ , supérieur à un certain seuil  $T_{Sp}$

- un score de similarité  $S_G$  défini par la somme des similarités des points d'intérêt appariés correspondants.

Une fois la meilleure hypothèse d'appariement sélectionnée pour chaque groupe candidat  $G'_j$  pour  $G_i$ , il faut déterminer celui qui correspond à un *vrai* appariement. Ceci est fait en comparant leurs similarités  $S_G$  après avoir appliqué le même seuil  $\Delta_{S_G}$  pour s'assurer que le meilleur appariement n'est pas ambigu. Le groupe d'appariement est validé en évaluant son score de corrélation normalisée à moyenne nulle (Zero Mean Normalized Cross Correlation) entre deux fenêtres centrées sur les pivots  $(g_0, g'_0)$  des deux groupes.

### 4.4 Reconstruction des points d'intérêt

Il s'agit de retrouver les coordonnées 3D des points d'intérêts appariés dans un repère indépendant du robot. A cet effet, nous employons la géométrie multi-vues. Nous procédons en deux étapes :

1. Résolution linéaire des  $2n$  systèmes ( $n$  étant le nombre de points de vue issus de 4.1)
2. Affinage du résultat par optimisation non linéaire

**Géométrie multi-vues.** La formulation mathématique est donné en 1. Considérons un point B de la scène de coordonnées  $(X, Y, Z)^T$  se projetant en un point  $x$  de l'image  $I_i$  en  $(u_i, v_i)^T$  et la matrice fondamentale de la caméra  $M = IE$ , nous utilisons la résolution par moindres carrés pour  $u_i$  et  $v_i$ . Le déplacement de la caméra est estimé par odométrie.

$$\begin{pmatrix} su \\ sv \\ s \end{pmatrix} = M \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = IE \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \quad (1)$$

$I = \begin{pmatrix} k_u f & 0 & u_0 & 0 \\ 0 & k_v f & v_0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$  est la *matrice intrinsèque*

obtenue après calibration avec **Camera Calibration Toolbox** sous Matlab.

$E = \begin{pmatrix} R & T \\ 0 & 1 \end{pmatrix}$  est la *matrice extrinsèque*. Elle représente

la matrice de passage entre le repère caméra et le repère du monde.  $E = R_{world}^{robot} \times R_{robot}^{hand} \times R_{hand}^{eye}$

Durant cette étape nous calculons pour chaque B l'erreur de reprojection en chaque vue  $I_i$  où il a été détecté. L'erreur moyenne de reprojection  $\mu$  est un outil de contrôle de faux appariements. Nous choisissons de rejeter les points dont l'erreur de reprojection dépasse  $1.5\mu$ .

**Ajustement de faisceaux.** L'ensemble des points  $B = \{B_1, \dots, B_n\}$  et l'ensemble des caméras  $A = \{A_1, \dots, A_m\}$  constitue les estimées initiales  $P$  de l'algorithme d'optimisation. Lourakis et Lhuillier ont indépendamment développé respectivement en [12] et [9] des algorithmes tenant compte de la structure creuse d'un système d'ajustement de faisceaux. L'algorithme développé est une implémentation des travaux du premier

se basant sur Levenberg-Marquardt (LM). Le développement de Taylor pour un  $\|\delta_p\|$  suffisamment petit donne  $f(p + \delta_p) \approx f(p) + J\delta_p$  avec  $J$  la matrice jacobienne  $\frac{df(p)}{dp}$ . A chaque étape du LM, il faut calculer  $\delta_p$  qui minimise la quantité

$$\|x - f(p + \delta_p)\| \approx \|x - f(p) - J\delta_p\| = \|\epsilon - J\delta_p\|$$

Pour un système de 4 points visibles dans 3 vues, la jacobienne possède la structure suivante (inspiré de [12]) :

$$\frac{dX}{dP} = \begin{bmatrix} A_{11} & 0 & 0 & B_{11} & 0 & 0 & 0 \\ 0 & A_{12} & 0 & B_{11} & 0 & 0 & 0 \\ 0 & 0 & A_{13} & B_{13} & 0 & 0 & 0 \\ A_{21} & 0 & 0 & 0 & B_{21} & 0 & 0 \\ 0 & A_{22} & 0 & 0 & B_{22} & 0 & 0 \\ 0 & 0 & A_{23} & 0 & B_{23} & 0 & 0 \\ A_{31} & 0 & 0 & 0 & 0 & B_{31} & 0 \\ 0 & A_{32} & 0 & 0 & 0 & B_{32} & 0 \\ 0 & 0 & A_{33} & 0 & 0 & B_{33} & 0 \\ A_{41} & 0 & 0 & 0 & 0 & 0 & B_{41} \\ 0 & A_{42} & 0 & 0 & 0 & 0 & B_{42} \\ 0 & 0 & A_{43} & 0 & 0 & 0 & B_{43} \end{bmatrix}$$

L'exploitation de cette structure creuse a l'avantage de réduire le temps de calcul en comparaison à un LM classique.

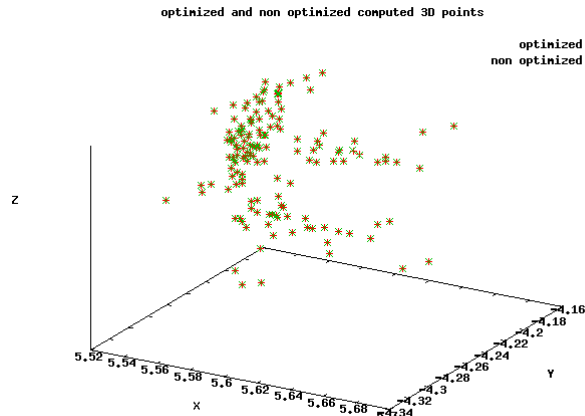


FIGURE 7 – Points 3D estimés par moindres carrés, puis par ajustement de faisceaux.

**Représentation.** Une structuration particulière des données issues de la détection permet une adaptation rapide à ces deux étapes. Chaque point d'intérêt  $PI_k$  est vu dans ensemble de vues  $\{V_1, \dots, V_n\}$ . Une vue  $V_i = (I_i, P_i)$  où  $I_i$  est une image et  $P_i$  la position associée. Dans chaque image  $I_i$  il est représenté par un pixel  $(u_i, v_i)$ . A l'issue de 4.4 nous disposons d'un couple  $(PI_k, (X, Y, Z)^T_k)$ . Il est à noter que chaque vue  $V_i$  est associé à une caméra  $C_j$  d'intrinsèques  $I_j$  et d'extrinsèques  $P_i$ . Cette structure de données est schématisée dans la figure 8.

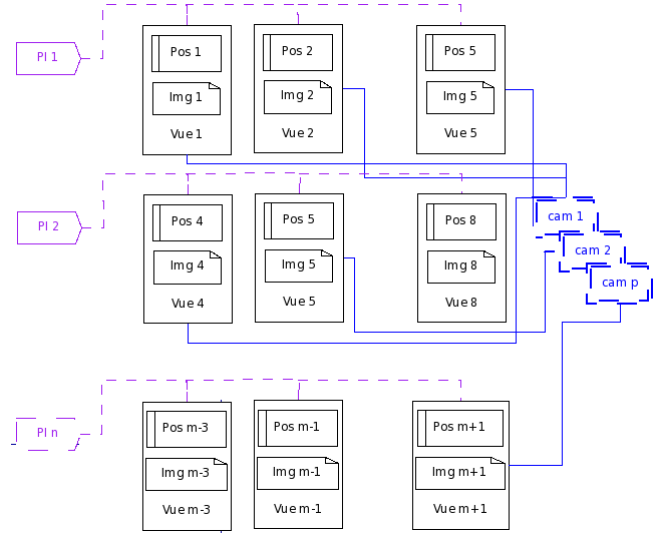


FIGURE 8 – Représentation des données pour la reconstruction

## 4.5 Reconstruction de surface

Les techniques de reconstructions de surfaces peuvent être groupées, classiquement, en deux familles :

- méthodes de reconstruction implicites : trouver une fonction  $f : \mathbb{R}^3 \rightarrow \mathbb{R}$  tel que  $\{x_1, \dots, x_n\}$  s'approche de l'ensemble nulle  $Z(f)$ . La différence entre elles concerne la forme de  $f$  et la mesure de la proximité.
- méthodes de reconstruction paramétriques : représenter la surface reconstruite comme une intégration topologique de  $f(\Lambda)$  d'un domaine paramétrique de dimension 2  $\Lambda$  dans  $\mathbb{R}^3$ . L'inconvénient de ces techniques c'est qu'elles requièrent des informations supplémentaires tel que la normale à la surface ou la topologie de la surface à reconstruire.

Dans ce contexte l'une et l'autre de ces techniques serait inappropriée étant donné que :

- l'objet est inconnu a priori ;
- l'objet peut être de forme quelconque donc complexe ;
- la seule information disponible est un ensemble de points 3D sur - sinon proches - de la surface à reconstruire ;
- il n'y a aucun ordre apparent quant à l'organisation des dits points.

Cependant, dans la littérature il existe un certain nombre d'algorithmes adaptés à la reconstruction de surface à partir d'un nuage de points 3D non organisé. Le plus connu est *marching cubes* (MC) de Lorensen et Cline [11]. Dans [15], les auteurs font la synthèse de tous les algorithmes qui s'en sont inspirés. L'une des rares alternatives au MC, est l'algorithme *ball pivoting* (BPA), introduit en [2] ou le *greedy Delaunay* (GD) présenté en [4].

Nous avons implémenté le BPA ainsi que le GD parce qu'ils dépassent en terme de performances et de qualité le MC dans le cadre d'un nuage de points 3D éparse. Le résultat

tat de la reconstruction par BPA est montré dans la figure 9

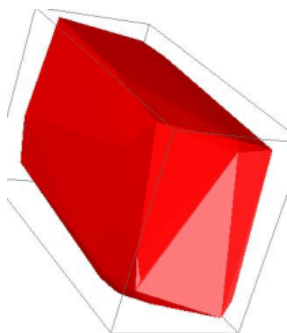


FIGURE 9 – Surface reconstruite par BPA

## 5 Reconnaissance/Localisation 3D

Une fois terminé la phase d'apprentissage, il est nécessaire de pouvoir 1. identifier l'objet 2. le localiser s'il changeait de position dans l'espace.

### 5.1 Reconnaissance

La reconnaissance s'effectue par le calcul d'appariement (section 4.3). Un objet est dit reconnu s'il existe suffisamment d'appariement entre son image courante et un des modèles d'objet enregistrés en base de données. Dans la figure 10, l'algorithme d'apprentissage ayant tourné sur une séquence d'images, nous lui présentons une image inconnue et nous lui demandons de conclure quant à la reconnaissance. Il calcule alors le rapport entre les points d'intérêts détectés et le plus grand nombre appariés. Le résultat est *reconnu à 81%*.

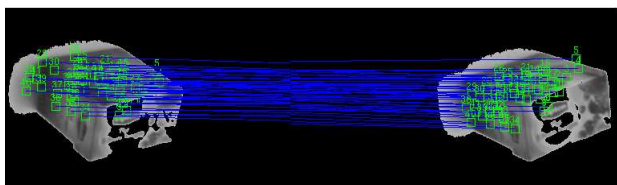


FIGURE 10 – Reconnaissance par group matching

Sur la figure, les points appariés sont reliés entre eux. Nous estimons un objet reconnu si le taux dépasse 80%. Une amélioration future serait de valider ce résultat par re-projection.

### 5.2 Localisation

Comme expliqué dans le paragraphe 4, la sortie de l'apprentissage est un modèle 3D de la surface de l'objet. Nous sauvegardons également la matrice d'inertie du nuage de points [3]. Nous utilisons la calibration de Tsai [19] pour retrouver les extrinsèques de la caméra en remplaçant les coins de la mire et de ses images, respectivement, par les points d'intérêt 2D et leurs correspondants 3D.

La matrice de passage du repère d'origine vers la position courante de la caméra nous donne le déplacement à effectuer. Le produit de cette dernière par la matrice d'inertie nous permet de retrouver rapidement le nouveau repère associé à l'objet.

## 6 Conclusions

Nous avons décrit dans cet article, une approche proposée pour apprendre des représentations géométrique et photométrique des objets que le robot devra ensuite manipuler. Nous avons aussi décrit sommairement comment ces représentations seront exploitées pour la reconnaissance et la localisation de ces objets.

Le modèle géométrique est un maillage triangulaire, qui est envoyé à un module de planification des positions de prise de l'objet [10]. Une fois sélectionnée cette position, le bras est programmé pour aller prendre l'objet : la phase suivante consistera à compléter la modèle de l'objet, donc de présenter la face occultée (face d'appui sur la table) à la caméra.

## Références

- [1] H. Bay, T. Tuytelaars, and L. Van Gool. Surf : Speeded up robust features. pages 404–417. 2006.
- [2] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin. The ball-pivoting algorithm for surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 5(4) :349–359, /1999.
- [3] S. Betge-Brezetz. *Modélisation incrémentale et localisation par amers pou la navigation d'un robot mobile autonome*. PhD thesis, Université Paul Sabatier de Toulouse, 1996.
- [4] D. Cohen-Steiner and F. Da. A greedy delaunay based surface reconstruction algorithm. Technical Report ECG-TR-12402-01, INRIA, 2002.
- [5] M. Cottret and M. Devy. Active learning of local structures from attentive and multi-resolution vision. In *9th International Conference on Intelligent Autonomous Systems (IA S-9)*, Tokyo (Japon), 2006. Intelligent Autonomous Systems 9, IOS Press, 2006, ISBN 1-58603-595-9, pp.534-541.
- [6] C. Harris and M. Stephens. A combined corner and edge detection. In *Proceedings of The Fourth Alvey Vision Conference*, pages 147–151, 1988.
- [7] D. Kragic, M. Björkman, H. I. Christensen, and J-O. Eklundh. Vision for robotic object manipulation in domestic settings. *Robotics and Autonomous Systems*, 52(1) :85 – 100, 2005. Advances in Robot Vision.
- [8] T. Lemaire, C. Berger, I-K. Jung, and S. Lacroix. Vision-based SLAM : stereo and monocular approaches. *International Journal on Computer Vision*, 74(3) :343–364, 2007.

- [9] M. Lhuillier and L. Quan. A quasi-dense approach to surface reconstruction from uncalibrated images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27 :418–433, 2005.
- [10] E. Lopez-Damian. *Grasp planning for object manipulation by an autonomous robot*. PhD thesis, LAAS, 7, Av Colonel Roche Toulouse France, juillet 2006.
- [11] W. E. Lorensen and H. E. Cline. Marching cubes : A high resolution 3d surface construction algorithm. In *SIGGRAPH '87 : Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, volume 21, pages 163–169, New York, NY, USA, July 1987. ACM Press.
- [12] M. I. A. Lourakis and A. A. Argyros. The design and implementation of a generic sparse bundle adjustment software package based on the levenberg-marquardt algorithm. Technical report, Institute of Computer Science, 2004.
- [13] D. G. Lowe. Object recognition from local scale-invariant features. volume 2, pages 1150–1157 vol.2, 1999.
- [14] Hans P. Moravec. 3d graphics and the wave theory. In *SIGGRAPH '81 : Proceedings of the 8th annual conference on Computer graphics and interactive techniques*, pages 289–296, New York, NY, USA, 1981. ACM.
- [15] T. S. Newman and H. Yi. A survey of the marching cubes algorithm. *Computers & Graphics*, 30(5) :854–879, October 2006.
- [16] J. Ponce, Y. Genc, and S. Sullivan. Constructing geometric object models from images. *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*, 1 :871–878 vol.1, 2000.
- [17] J. Ross Stenstrom and C. Ian Connolly. Constructing object models from multiple images. *International Journal of Computer Vision*, 9 :185 – 212, December 1992.
- [18] F. Trujillo and M. Devy. Registration by using a pseudo color attribute. In *International Conference on Pattern Recognition (ICPR2008)*, Tampa, Floride (USA), 2008.
- [19] R. Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *Robotics and Automation, IEEE Journal of*, 3(4) :323–344, Aug 1987.