



**HAL**  
open science

# Modélisation d'objets 3D par construction incrémentale d'un maillage triangulaire, dans un contexte robotique

Juan Andres Restrepo Specht Restrepo Specht

## ► To cite this version:

Juan Andres Restrepo Specht Restrepo Specht. Modélisation d'objets 3D par construction incrémentale d'un maillage triangulaire, dans un contexte robotique. Automatique / Robotique. Université Paul Sabatier - Toulouse III, 2005. Français. NNT: . tel-00009525

**HAL Id: tel-00009525**

**<https://theses.hal.science/tel-00009525>**

Submitted on 17 Jun 2005

**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 par construction incrémentale d'un maillage triangulaire, dans un contexte robotique.

## THÈSE

présentée et soutenue publiquement le 31.01.2005

pour l'obtention du

**Doctorat de l'Université Toulouse III**  
(spécialité systèmes informatiques)

par

Juan Andres Restrepo Specht

### Composition du jury

<i>Rapporteurs :</i>	Françoise Prêteux Marie-Odile Berger
<i>Examineurs :</i>	Raja Chatila Patrice Dalle Jean-José Orteu
<i>Invité :</i>	Carlos Parra Rodriguez
<i>Directeur de thèse :</i>	Michel Devy

Mis en page avec la classe thloria.

*A mi madre, mi hermana y Margarita*



# Table des matières

<b>Table des figures</b>	<b>ix</b>
--------------------------	-----------

<b>Introduction générale</b>	<b>1</b>
------------------------------	----------

1	Introduction . . . . .	1
2	Contributions. . . . .	3
3	Organisation du mémoire. . . . .	4

## 1

### Approche à la modélisation d'objets 3D

1.1	Description. . . . .	5
1.2	Nos choix sur les fonctions de modélisation . . . . .	6
1.2.1	Acquisition de données 3D . . . . .	6
1.2.2	Fusion des données 3D . . . . .	9
1.2.3	Contrôle de la tâche de modélisation . . . . .	10
1.3	Modélisation 3D dans d'autres domaines. . . . .	11
1.4	Cahier des charges de notre système de modélisation. . . . .	11

## 2

### Recalage des images 3D

2.1	Introduction. . . . .	13
2.2	État de l'art. . . . .	14
2.3	Description de la méthode. . . . .	16
2.4	Sélection des points de contrôle. . . . .	17
2.4.1	Extraction des contours 3D. . . . .	18
2.4.2	Construction d'un maillage triangulaire irrégulier. . . . .	21
2.5	Recherche d'appariements entre points de contrôle. . . . .	26
2.5.1	Structure des données. . . . .	26
2.5.2	Processus. . . . .	28
2.5.3	Filtrage. . . . .	30
2.5.4	Algorithme complet. . . . .	31
2.6	Estimation de la transformation 3D. . . . .	31
2.6.1	Estimation par SVD. . . . .	33
2.6.2	Estimation par quaternions doubles. . . . .	35

2.6.3	Estimation par quaternions. . . . .	37
2.7	Stratégies d'itérations avec modification du comportement de l'algorithme. . . . .	38
2.8	Variantes dans l'application de la méthode d'ICP. . . . .	40
2.9	Vérification quantitative. . . . .	41
2.9.1	Méthodologie de comparaison. . . . .	43
2.9.2	Robustesse par rapport aux conditions initiales. . . . .	44
2.9.3	Robustesse par rapport à la résolution des images. . . . .	47
2.9.4	Robustesse par rapport aux taux de recouvrement des vues. . . . .	50
2.9.5	Nombre d'itérations en fonction de l'erreur de recalage. . . . .	55
2.9.6	Evaluation globale. . . . .	57
2.10	Conclusion. . . . .	58

**3**

**Fusion d'images 3D**

3.1	Introduction. . . . .	61
3.2	Etat de l'art. . . . .	63
3.2.1	Construction incrémentale de maillage . . . . .	63
3.2.2	Algorithme des "Marching Cubes". . . . .	64
3.2.3	Surfaces déformables. . . . .	65
3.2.4	Méthodes mixtes : maillage et primitives surfaciques . . . . .	66
3.3	Description de l'algorithme BPA. . . . .	68
3.3.1	Initialisation : choix du triangle-souche . . . . .	70
3.3.2	Recherche des points candidats. . . . .	70
3.3.3	Sélection du point candidat. . . . .	71
3.3.4	Création d'un nouveau triangle. . . . .	73
3.3.5	Exemple d'exécution de l'algorithme. . . . .	74
3.3.6	Evaluation de l'algorithme BPA. . . . .	74
3.3.7	Modalités d'exécution de BPA : construction incrémentale, affinement de maillage. . . . .	74
3.4	Comparaison globale entre les méthodes de maillage. . . . .	77
3.5	Comparaison quantitative des méthodes <i>Ball Pivoting</i> et <i>Surfaces Déformables</i> . . . . .	79
3.5.1	Test de qualité de maillage. . . . .	79
3.5.2	Résultats globaux. . . . .	80
3.5.3	Résultats pour les différentes images de test. . . . .	83
3.6	Conclusion. . . . .	88

**4**

**Planification perceptuelle : sélection de la vue optimale**

4.1	Introduction. . . . .	91
4.2	État de l'art. . . . .	91
4.3	Description de la méthode. . . . .	93
4.3.1	Une vue d'ensemble de notre système. . . . .	93
4.3.2	La fonction d'utilité pour un point de vue. . . . .	94

4.4	Description de l'algorithme. . . . .	99
4.5	Adaptation au cas de la modélisation des objets. . . . .	101
4.6	Conclusion sur la méthode NBV . . . . .	104

## 5

### Résultats expérimentaux

5.1	Introduction. . . . .	107
5.2	Application de la méthode d'ICP pour la navigation d'un robot mobile. . . . .	107
5.3	Résultats de BPA. . . . .	111
5.3.1	Exemples. . . . .	112
5.3.2	Performances. . . . .	114
5.4	Résultats de l'intégration. . . . .	115
5.4.1	Stratégie. . . . .	115
5.4.2	Etalonnage. . . . .	115
5.4.3	Pompe. . . . .	115
5.4.4	Tête-de-dragon. . . . .	118
5.5	Conclusion . . . . .	119

<b>Conclusion générale</b>	<b>121</b>
----------------------------	------------

## A

### Représentations des rotations par les quaternions

A.1	Quaternions. . . . .	125
A.2	Quaternions doubles. . . . .	130

## B

### Pseudo-code pour la fonction ICP

<b>Bibliographie</b>	<b>139</b>
----------------------	------------





# Table des figures

1	<i>Exemples d'applications de la modélisation 3D.</i>	2
2	<i>L'église ronde d'Osterlars sur l'île de Bornholm, vue extérieure et intérieure.</i>	2
3	<i>Organisation des chapitres.</i>	4
1.1	<i>Boucle de modélisation.</i>	5
1.2	<i>Différents capteurs laser 3D et 2D commerciaux.</i>	7
1.3	<i>Dispositif BODYSCAN.</i>	8
1.4	<i>Tête stéréoscopique.</i>	8
1.5	<i>Variante de capture d'images 3D.</i>	9
1.6	<i>Structures des systèmes de modélisation.</i>	10
2.1	<i>Diagramme du processus d'ICP.</i>	15
2.2	<i>Données 3D denses d'origine.</i>	18
2.3	<i>Étapes de l'extraction des contours 3D.</i>	19
2.4	<i>(a) Une colonne de l'image et l'approximation initiale par une parabole. ; (b) Discontinuités de normale retenues.</i>	20
2.5	<i>Vues de détail sur l'extraction des contours 3D.</i>	21
2.6	<i>Sous-image centrée sur l'autel, d'une image de profondeur acquise dans une église.</i>	22
2.7	<i>Ensemble des points de contour extraits sur la sous-image centrée sur l'autel</i>	22
2.8	<i>Résultat de la segmentation en contours : (a) arbre de recouvrement non filtré; (b) arbre de recouvrement filtré</i>	22
2.9	<i>Description du maillage initial.</i>	23
2.10	<i>Processus d'insertion d'un contour : (a) segment à insérer, (b) insertion des extrémités, (c) destruction des triangles, (d) remaillage.</i>	24
2.11	<i>Insertion d'un sommet dans un triangle : (a) point à insérer, (b) création des nouveaux triangles <math>N1, N2, N3</math>, mise à jour des relations de voisinage inter-triangles en (c), inter-sommets en (d).</i>	24
2.12	<i>Filtrage successif d'un maillage : (a) maillage après le filtrage des facettes à l'horizon, (b) maillage après le filtrage de la densité et filtrage topologique.</i>	25
2.13	<i>Exemple de la construction d'un maillage.</i>	25
2.14	<i>Organisation dans l'espace d'un k-d tree.</i>	26
2.15	<i>Organisation de l'espace par clusters.</i>	26
2.16	<i>Recalage de deux vues avec une organisation en clusters.</i>	28
2.17	<i>Cluster <math>B[i][j][k]</math> avec un voisinage de recherche d'un cluster</i>	29
2.18	<i>Structure de données décrivant une organisation de l'espace par clusters.</i>	29
2.19	<i>Filtrage des appariements : points extrêmes d'un des modèles</i>	30
2.20	<i>Filtrage d'unicité avec un empilement doté de 3 minima.</i>	30
2.21	<i>L'algorithme d'appariement basé sur des espaces de clusters et des piles de distances minimales.</i>	32
2.22	<i>Deux ensembles de points appariés, avant et après recalage.</i>	33
2.23	<i>le changement de stratégies pendant un processus d'ICP.</i>	39
2.24	<i>L'algorithme d'ICP.</i>	40

2.25	<i>Variantes d'applications d'ICP.</i>	42
2.26	<i>Exemple de recalages cycliques triangulaires sur l'ensemble d'images de BUNNY.</i>	42
2.27	BUNNY.	43
2.28	DRAGON.	43
2.29	WYE.	43
2.30	HAPPYBUDDHA.	43
2.31	CURVBLOCK et DOUGHBOY.	43
2.32	BLOCK2 et VALVE.	43
2.33	<i>Exemple 1, DRAGON : A gauche, toutes les positions initiales, présentés comme des croix. A droite, les positions qui ont produit une bonne convergence pour les tests fondés sur les contours.</i>	45
2.34	<i>Exemple 1, DRAGON : toutes les erreurs d'ICP, à gauche, pour les tests basés sur les contours et à droite, pour les tests basés sur les sommets de maillage.</i>	45
2.35	<i>Exemple 2, CURVBLOCK : les positions qui ont produit une bonne convergence pour les tests à partir de la méthode du maillage.</i>	46
2.36	<i>Exemple 2, CURVBLOCK : toutes les erreurs d'ICP, à gauche, pour les tests obtenus à l'aide de la technique des contours et à droite, pour les tests à partir de la méthode des maillages.</i>	46
2.37	<i>Exemple 3, BUNNY : les positions qui ont produit une bonne convergence pour les tests basés sur maillage.</i>	48
2.38	<i>Exemple 3, BUNNY : à gauche, toutes les erreurs d'ICP et à droite, l'erreur comparée à la position connue pour les tests à partir de la méthode du maillage.</i>	48
2.39	<i>Évaluation de la méthode d'ICP par rapport à la résolution de l'image : le recalage utilise le maillage à haute (gauche) et à basse résolution (droite).</i>	49
2.40	<i>Évaluation de la méthode d'ICP avec des résolutions différentes. (a) recalage par maillage, (b) recalage par contours, (c) l'erreur d'ICP pour les combinaisons de résolutions.</i>	50
2.41	<i>Évaluation de la méthode d'ICP par rapport au taux de recouvrement entre nuages de points.</i>	51
2.42	<i>Évaluation de la méthode d'ICP par rapport au taux de recouvrement entre les nuages de points pour l'objet DOUGHBOY.</i>	52
2.43	<i>Évaluation de la méthode d'ICP par rapport du taux de recouvrement entre les nuages de points pour l'objet VALVE.</i>	53
2.44	<i>Diminution de l'erreur par rapport à la vérité terrain au cours des itérations pour les objets DRAGON et CURVBLOCK.</i>	55
2.45	<i>Évaluation des méthodes d'ICP sur plusieurs images de BUNNY : recalage avec sommets de maillage (a), recalage avec points de contour (b). Comportement de l'erreur d'ICP (c).</i>	56
2.46	<i>Évaluation de la méthode d'ICP sur l'objet cylindrique WYE, en utilisant seulement les contours de d'orientation à gauche, par contre, à droite tous les points de contour ont été utilisés.</i>	58
2.47	<i>Évaluation de la méthode d'ICP : (gauche) deux vues de WYE recalées avec l'information du maillage. A droite, nous montrons deux vues de BLOCK2 recalées avec les contours.</i>	58
3.1	<i>Exemple de recalage et construction d'une représentation finale, à partir de 4 vues de synthèse.</i>	62
3.2	<i>Combinaisons de base des surfaces pour un voxel de l'algorithme des "Marching Cubes".</i>	64
3.3	<i>Diagramme de l'algorithme de la boule pivotante. La ligne épaisse montre le chemin de croissance d'une surface.</i>	67
3.4	<i>La boule pivotante de rayon <math>\rho</math> et de centre <math>L</math>, repose sur un triangle <math>ABC</math>. <math>M</math> (sur le plan <math>ABC</math>) est le centre du cercle circonscrit à <math>ABC</math>.</i>	69
3.5	<i>La boule pivote autour de l'axe <math>BC</math>, qui est désigné comme l'arête pivot.</i>	69
3.6	<i>Impact de l'utilisation de clusters, en analysant le nombre de points analysés en fonction du nombre de triangles créés.</i>	71
3.7	<i>Diagramme de l'algorithme de recherche de points pour la création d'un triangle décrit a la section 3.3.2.</i>	72
3.8	<i>Progression du maillage suivant une évolution en spirale.</i>	73

---

3.9	<i>La construction de la surface triangulaire étape par étape décrite dans la section 3.3.5.</i>	75
3.10	<i>Résultat de BPA avec segmentation : (gauche en haut) nuage synthétique constitué de deux vues (droite en haut) un seul plan . En bas, des détails sont montrés.</i>	76
3.11	<i>Résultat de BPA avec segmentation : (gauche), nuage synthétique constitué de six vues, (droite) deux plans orthogonaux</i>	76
3.12	<i>Vues détaillées du résultat de la figure 3.11 : seuls deux plans ont été détectés malgré les erreurs de recalage.</i>	77
3.13	<i>Test du critère de la qualité <math>Q_{equ}</math> sur la forme du triangle, appliqué à divers triangles.</i>	80
3.14	<i>Test du critère de qualité <math>Q_{nor}</math> sur la courbure d'un maillage.</i>	81
3.15	<i>Critères <math>Q_{equ}</math> et <math>Q_{nor}</math> testés sur une portion d'image de synthèse.</i>	82
3.16	<i>Exemple d'objet "cat" sur (a) avec BPA (b) surface déformable.</i>	83
3.17	<i>Exemple d'objet "club" sur (a) avec BPA (b) surface déformable.</i>	84
3.18	<i>Exemple d'objet "mech" sur (a) avec BPA (b) surface déformable.</i>	85
3.19	<i>Exemple d'objet "scurv" sur (a) avec BPA (b) surface déformable.</i>	86
3.20	<i>Exemple d'objet "halfhead" sur (a) avec BPA (b) surface déformable. Dans (c), le détail montre la densité variable des points qui a été traitée par BPA par plusieurs exécutions en utilisant un rayon initial de la boule plus petit.</i>	87
3.21	<i>Exemple d'objet "teapot" : (a) avec BPA (b), (c) et (d) surface déformable.</i>	88
4.1	<i>Types de voxels.</i>	94
4.2	<i>Variation de la fonction d'utilité pour chaque type de voxel pour un <math>\alpha</math> donné et les restrictions à accomplir sur les points indiqués.</i>	95
4.3	<i>Matrice d'acquisition d'une résolution <math>L \times H</math> à gauche. Parcours des rayons à droite.</i>	95
4.4	<i>Fonctions des types de voxels marqués occupé (gauche), inconnu et occplane (centre), frontière (droite).</i>	98
4.5	<i>Environnement de test et position du capteur.</i>	99
4.6	<i>Stratégie d'optimisation.</i>	99
4.7	<i>Progression de la fonction d'optimisation pour <math>\phi</math> et <math>\theta</math> à chaque prise de vue.</i>	100
4.8	<i>Données du capteur pour les six premières vues.</i>	101
4.9	<i>Le diagramme de l'algorithme de Next Best View.</i>	102
4.10	<i>Plan de l'environnement à explorer.</i>	103
4.11	<i>Les étapes du processus de Next Best View dans un environnement navigable.</i>	103
4.12	<i>Positions du robot mobile pour l'acquisition de l'objet.</i>	104
4.13	<i>Exemple des positions d'un bras manipulateur GT6a autour de l'objet à modéliser.</i>	105
5.1	<i>Recalage des 3 premiers profils, égal pour les deux approches.</i>	109
5.2	<i>Recalage avec approche 1 : recalage cumulatif.</i>	110
5.3	<i>Recalage avec approche 2 en utilisant uniquement une fenêtre de 3 profils.</i>	111
5.4	<i>Résultat de BPA sur des images synthétiques d'un objet polyédrique.</i>	113
5.5	<i>Résultat de BPA sur des images acquises par stéréovision.</i>	114
5.6	<i>Construction d'un maillage à partir de deux segments d'images acquises par profilométrie.</i>	114
5.7	<i>Images prises par la caméra stéréoscopique sur la pompe (caméra droite).</i>	116
5.8	<i>Différents points de vue autour de l'objet "Pompe".</i>	116
5.9	<i>Les nuages de points des vues fusionnées.</i>	118
5.10	<i>Images prises par la caméra stéréoscopique (caméra droite).</i>	119
5.11	<i>Les nuages de points de contrôle des vues fusionnées.</i>	120
12	<i>Capteur 3D optique à temps de vol.</i>	123
A.1	<i>Le quaternion représente une rotation d'un angle <math>\theta</math> autour d'un vecteur unitaire <math>\vec{u}</math>.</i>	126
B.1	<i>L'algorithme d'appariement basée sur des espaces de clusters</i>	137
B.2	<i>L'algorithme d'ICP</i>	138



# Introduction générale

## 1 Introduction

Le thème de notre travail de thèse est la modélisation tridimensionnelle d'objets. A l'origine, nous avons participé au réseau européen CAMERA, pour *CAd Modelling of Built Environments from Range Analysis*, dédié à la modélisation de bâtiments à partir d'images de profondeur acquises principalement par télémétrie-laser ; ce projet s'est terminé en Décembre 2001. Ensuite, notre intérêt s'est porté sur la modélisation d'objets que devrait manipuler un système robotique, équipé de caméra(s) : changement d'échelle, mais aussi, de capteur et de modalités de déplacement de ce capteur.

L'objectif principal du projet CAMERA était d'entreprendre une recherche multidisciplinaire pré-compétitive afin d'améliorer les solutions existantes dans le domaine de l'acquisition automatisée des modèles de type CAO, d'environnements architecturaux déjà construits. Les secteurs principaux d'application de ce type de travaux sont :

- les industries chimique, nucléaire... pour la modélisation de bâtiments industriels : raffinerie, atelier d'assemblage, centrale nucléaire... Les modèles perceptuels peuvent servir à la planification et simulation d'opérations de maintenance, comme pour le modèle de centrale exploité dans le cadre du projet européen MOLOG, présenté en figure 1 à gauche.
- le secteur des BTP, pour la vérification de conformité d'un bâtiment par comparaison entre un modèle perceptuel et un modèle purement CAO,
- le secteur du tourisme, pour la modélisation de monuments historiques : cathédrales, ruines antiques... modèles exploités sur des sites Web pour permettre des visites virtuelles de monuments, comme le modèle 3D avec projection de texture présenté en figure 1 à droite.

Le but de CAMERA était de traiter des images de profondeur et d'intensité acquises sur un bâtiment, afin d'en construire un modèle en réalité virtuelle. Les technologies principales employées sont les télémètres laser 3D pour l'acquisition des images de profondeur, la reconstruction des surfaces avec projection (ou collage) de textures, la reconstruction photogrammétrique et non-calibrée à partir d'images d'intensité, la représentation de formes par des primitives géométriques de type CAO (facettes planes, cylindriques, sphériques...).

Le bâtiment de test choisi dans ce projet était l'église ronde du douzième siècle d'Osterlars [Haagensen 00] sur l'île de Bornholm, Danemark, présentée sur la figure 2 ; nous avons traité aussi de nombreuses images acquises sur des bâtiments industriels mises à disposition par la compagnie UK Robotics, partenaire du projet.

Dans le cadre du projet CAMERA, nous avons réalisé les travaux présentés dans cette thèse, sur la phase de recalage et sur l'extraction de points d'intérêt 3D, en utilisant les données acquises par un capteur laser de haute résolution. Un fois le projet fini, nous avons envisagé d'autres types de capteurs comme les têtes stéréoscopiques ou les capteurs à lumière structurée, exploitant par exemple la profilométrie. Nous avons privilégié la stéréovision, afin d'appliquer ces techniques à la robotique, plus exactement aux applications de manipulation d'objets ou de navigation : les modèles perceptuels sont alors exploités pour la recherche de positions de prise sur un objet, ou pour la planification de trajectoires pour une plateforme mobile.



FIG. 1 – Exemples d'applications de la modélisation 3D.



FIG. 2 – L'église ronde d'Osterlars sur l'île de Bornholm, vue extérieure et intérieure.

Dans ce nouveau contexte, nos travaux ont suivi la chaîne de traitements sur la modélisation, dont la première phase est le recalage (placement correcte d'images acquises depuis différents points de vue). A partir d'images recalées, comment traiter de la fusion des données sensorielles ? comment représenter l'entité à modéliser : environnement, bâtiment, terrain, objet ... ? Nous avons d'abord analysé les différentes alternatives connues et notamment, les travaux réalisés précédemment dans notre groupe de recherche, sur la modélisation à partir de données 3D :

- thèse de F.Nashashibi sur la construction de modèles polyédriques pour des environnements intérieurs ;
- thèse de P.Fillatreau sur la construction de modèles numériques de terrain par des surfaces B-splines ;
- thèse de X.Fouchet sur la modélisation corporelle par un maillage triangulaire construit par une version de l'algorithme des *Marching Cubes* adapté à des données acquises par profilométrie ;
- et enfin, thèse de J.Bozier, sur la construction incrémentale de maillage triangulaire.

A partir de l'algorithme connu sous le nom *Ball Pivoting*, nous avons développé une méthode de construction de maillage triangulaire, adaptée à notre contexte.

Finalement la modélisation d'un environnement ou d'un objet 3D, se fait à partir de plusieurs images acquises depuis des points de vue différents. Pour fermer la boucle, un système idéal doit exploiter

en ligne, une méthode qui, à chaque image, donne à partir de l'état courant du modèle, la position optimale du capteur. Ce problème est référencé dans la littérature comme le problème du *Next Best View* ; nous avons donc adapté à nos besoins, des travaux entamés dans le projet CAMERA, avec pour but, l'automatisation du processus de modélisation d'un objet 3D depuis un capteur stéréoscopique monté sur un bras manipulateur mobile.

## 2 Contributions.

Nos contributions portent sur des adaptations apportées sur les trois méthodes que nous avons exploitées, et sur l'évaluation de leurs performances.

La méthode ICP, pour *Iterative Closest Point*, traite du recalage entre deux ensembles de points 3D à partir d'appariements entre points de ces deux ensembles ; nous nous sommes intéressés principalement à la robustesse et aux performances de la méthode.

La complexité de ICP est directement fonction du nombre de points 3D des ensembles à recaler : comme pré-traitement du recalage d'une image 3D (nuage de points 3D) avec un modèle courant, nous proposons une méthode d'extraction des points de contour 3D, qui seuls seront exploités par ICP. Nous avons montré que cette technique permettait de réduire le temps de calcul par rapport au procédé alternatif classique fondé sur la décimation d'un maillage local construit à partir de cette image.

Pour rendre ICP plus robuste, nous proposons plusieurs techniques pour filtrer les appariements. En particulier, un filtre d'unicité est appliqué avec une stratégie adaptée à l'état courant du traitement : estimation grossière ou fine du recalage. Un niveau de contrôle permet une commutation automatique entre plusieurs modalités de la méthode ICP, en fonction des erreurs d'approximation.

Après acquisition et recalage d'une nouvelle image sur l'entité à modéliser, deux stratégies sont possibles : soit les points 3D de cette image sont plongés dans un nuage de points 3D, dans lequel ont été fusionnées toutes les images précédemment acquises, soit une représentation plus abstraite (maillage, ensemble de primitives surfaciques...) est construite de manière incrémentale.

La technique de construction de maillage appelée BPA, pour *Ball Pivoting Algorithm*, permet de sélectionner l'une ou l'autre de ces stratégies : construction globale ou incrémentale d'un modèle 3D. Par ailleurs, notre version de l'algorithme BPA permet (1) une segmentation automatique en primitives surfaciques (application pour l'extraction de facettes planes), (2) l'intégration de données de résolution variable, en adaptant les paramètres (rayon de la boule) à la résolution des images ou zones d'image à fusionner...et (3) la prise en compte du bruit sur les données à fusionner. Ces deux dernières adaptations sont particulièrement nécessaires pour traiter des données acquises par stéréovision : résolution et précision sont dans ce cas, irrégulières entre images ou entre zones d'une même image.

Notre contribution sur la technique développée précédemment dans notre groupe, pour traiter du problème NBV, pour *Next Best View*, est plus interne à l'algorithme. Le NBV optimal est sélectionné par maximisation d'une fonction d'utilité non linéaire ; notre technique présentait plusieurs défauts, en particulier, convergence vers des optima locaux, temps de calcul trop important pour une intégration dans un système de modélisation en ligne...

Dans un environnement de simulation, nous avons tout d'abord caractérisé les situations critiques pour lesquelles la technique d'origine ne converge pas. Puis nous avons proposé une technique non optimale, adaptée à notre configuration matérielle : un capteur monté à proximité de l'effecteur d'un bras manipulateur.

Toutes ces méthodes ont été testées sur de nombreuses images 3D : images de synthèse et images acquises par différents types de capteurs (télémétrie, stéréovision, profilométrie)...

Notre ambition était de fournir un module fonctionnel de modélisation automatique d'objets 3D depuis un capteur embarqué sur un robot ; ce module pourrait être intégré dans le système embarqué sur un robot, afin de réaliser la tâche complexe *Saisir un objet inconnu a priori*. Les performances des méthodes



développées n'ont pas permis d'intégrer à bord des robots la chaîne complète de traitement. Mais hors ligne, nous avons pu construire des modèles d'objets 3D à partir de données acquises par nos capteurs.

### 3 Organisation du mémoire.

Le mémoire est organisé en 5 chapitres. Le chapitre 1 est une courte introduction sur le thème de la modélisation et des capteurs. Le chapitre 2 traite du recalage; le chapitre 3 présente nos travaux sur la construction automatique de maillage triangulaire; le chapitre 4 décrit le problème NBV, expose les problèmes détectés sur une technique développée précédemment dans notre équipe et propose une solution non optimale pour la sélection des points de vue depuis lesquels acquérir les images. Enfin, le chapitre 5 présente les résultats obtenus sur différentes images et l'état courant du système intégré de modélisation.

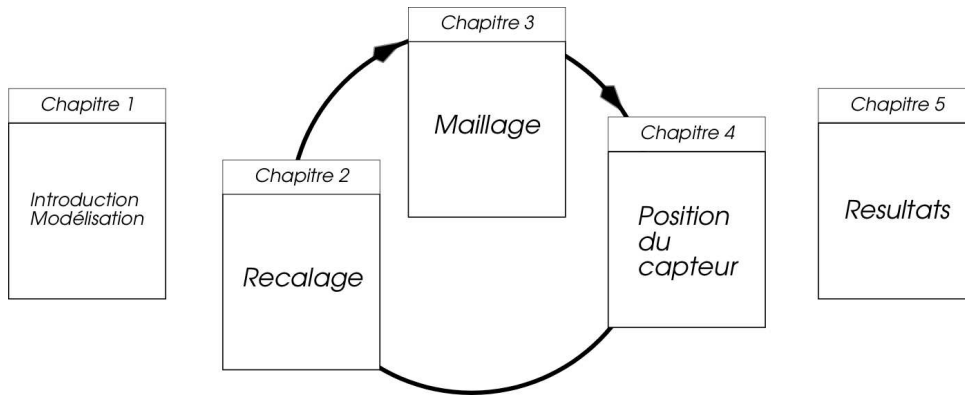


FIG. 3 – Organisation des chapitres.

# Chapitre 1

## Approche à la modélisation d'objets 3D

### 1.1 Description.

La modélisation 3D d'un environnement ou d'un objet à partir de données sensorielles, consiste à acquérir des données depuis les capteurs positionnés en divers points de vue, puis à générer une représentation adaptée à la tâche qui exploitera le modèle obtenu : cela consiste souvent à faire une approximation des points de mesures par des fonctions mathématiques.

Ces représentations peuvent être globales, par exemple la surface, les cylindres inclus, les axes d'inertie de l'objet, ou locales comme les normales et la courbure des surfaces et triangles. Ces attributs seront utilisés pour différents buts, dépendant de l'application.

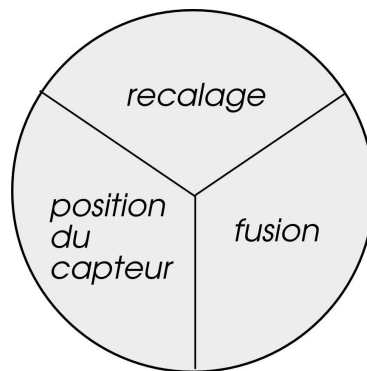


FIG. 1.1 – Boucle de modélisation.

Les applications envisagées pour nos méthodes sont principalement reliées à la robotique mobile ou à la manipulation et la navigation, mais les applications de la modélisation tridimensionnelle en général sont innombrables comme par exemple la simulation du comportement des éléments mécaniques de diverses pièces, la reconstruction par CAO ou le *reverse engineering*, la réalité virtuelle, les applications médicales, les industries de vêtements, l'architecture, l'archéologie, etc.

Il n'est donc pas étonnant que les problèmes traités dans cette thèse aient été abordés dans beaucoup de projets de recherche, comme on peut apercevoir dans la bibliographie, et dans des produits commerciaux dédiés au secteur de la modélisation et du traitement d'images 3D.

Dans le cadre du projet CAMERA, nous avons coopéré avec la compagnie *UK Robotics*, spécialisée

en particulier sur la construction d'un modèle 3D pour des bâtiments. Les applications d'un tel modèle sont par exemple le contrôle de qualité par des architectes, la création d'information 3D des monuments historiques ou sur des bâtiments publics comme des musées afin de proposer des visites virtuelles sur un site Web...

La modélisation implique de nombreux traitements. Les trois fonctions sur lesquelles nous avons travaillé, sont illustrées sur la figure 1.1. Dans ce chapitre, nous précisons tout d'abord, le contexte de nos travaux : nous précisons ci-dessous quels ont été nos principaux choix, puis en section 1.3, nous situons nos travaux vis-à-vis d'autres domaines en fort développement, la problématique du SLAM pour *Simultaneous Localization and Mapping* en Robotique, et du SFM pour *Structure From Motion* en Vision. Enfin nous dressons un bref cahier des charges du système de modélisation que nous souhaitons développer.

## 1.2 Nos choix sur les fonctions de modélisation

Notre sujet concerne la modélisation 3D, ou l'apprentissage de modèle géométrique d'objets 3D. Ces modèles seront exploités sur une machine intelligente, à des fins de raisonnement et planification de tâches. D'autres doctorants traitent dans notre équipe, de la saisie d'objets depuis un bras manipulateur mobile. Une telle tâche requiert des fonctionnalités perceptuelles :

- soit pour modéliser en ligne un objet inconnu a priori, afin de déterminer des positions de prise avant de le saisir ;
- soit pour reconnaître un objet dont le modèle est déjà appris, afin de le localiser en 3D pour générer la trajectoire de la plateforme et du bras qui permettront de saisir cet objet.

Modéliser requiert d'acquérir des données, de générer une représentation dans laquelle ces données seront fusionnées et de contrôler l'ensemble de ces traitements.

### 1.2.1 Acquisition de données 3D

Il existe une grande variété de capteurs utilisés pour la numérisation tridimensionnelle ; ils utilisent des techniques classifiées entre passives et actives. Les capteurs passifs n'émettent pas d'énergie sur la scène : pour nos applications, cela se limite à la Vision. Par contre, les capteurs actifs émettent une énergie vers la scène et détectent les effets de cette énergie afin de faire des mesures 3D ; de nombreux outils de modélisation exploitent ces techniques actives

- la télémétrie-laser, surtout pour la modélisation d'environnements ou de grands objets.
- la vision avec lumière structurée, surtout pour la modélisation d'objets de petite taille.
- la profilométrie qui est une technique spécifique de vision avec lumière structurée, très utilisée pour la modélisation d'objets.

De nombreuses entreprises ont des sites Web sur lesquelles elles proposent des capteurs, ainsi que les logiciels de traitement ou les services associés. Nous ne présenterons pas en détail ces techniques dans ce mémoire, mais nous donnerons quelques exemples.

#### La télémétrie-laser

Le principe physique de la télémétrie-laser est la mesure du temps de vol ou du déphasage, d'une onde lumineuse. Ce dispositif doit être associé à un mécanisme de balayage afin d'acquérir un ensemble de points 3D, organisé sous la forme :

- d'un profil ou d'une coupe 2D pour un balayage dans un seul plan (en azimut généralement),
- d'une image de points 3D pour un balayage deux axes (en site et azimut)

Différents télémètres sont présentés en figure 1.2 ; nous avons utilisé des données de type coupe 2D acquises par un capteur SICK monté sur un robot mobile (voir section 5.1) ainsi que des images 3D acquises par un capteur panoramique Zoeller&Froelich (sous-image présentée dans le chapitre 2, en figure 2.6).



FIG. 1.2 – Différents capteurs laser 3D et 2D commerciaux.

### Vision monoculaire avec lumière structurée

Un capteur de lumière structurée se compose généralement d'une caméra et d'une source laser qui génère un motif lumineux sur la scène comme par exemple une ligne, une matrice de points, une série de cercles, avec ou sans synchronisation lors de l'acquisition. L'information 3D est calculée à partir de la déformation du motif projeté sur la scène à modéliser, par conséquent les points tridimensionnels ne sont pas obtenus directement en sortie du capteur, mais après une série de calculs a posteriori : détection des motifs lumineux dans l'image, calcul d'informations 3D à partir de données de calibrage...

Le coût est supérieur ou égal à un capteur stéréoscopique; par contre les calculs pour l'obtention de l'information 3D sont moins lourds et les résultats sont plus précis. Les expérimentations montrent que les scènes à capturer n'ont pas besoin de texture et les conditions d'éclairage ne sont pas critiques. Les inconvénients sont la sensibilité à la spécularité de la surface et à sa nature plus ou moins absorbante. Ce type de capteur nécessite une phase de calibration comparable en complexité, à celle de la stéréovision.

### La profilométrie.

C'est une technique très utilisée pour la modélisation d'objets, quelque soit leur taille : depuis la dent (pour prendre des empreintes) jusqu'au corps d'une personne, pour des applications médicales, ou pour la réalisation de films de synthèse (multimédia).

En ce cas, l'illuminateur projette un plan de lumière et la caméra détecte l'intersection de ce plan et des surfaces présentes dans la scène, donc une courbe ou un profil. Le déplacement de l'ensemble illuminateur/caméra permet d'acquérir plusieurs profils. Ce déplacement peut être fait de diverses manières :

- selon un axe vertical. Le dispositif BODYSCAN présenté en figure 1.3, a été réalisé dans notre équipe en collaboration avec un laboratoire médical. Quatre profilomètres précisément alignés pour que les plans de lumière soient confondus, balayent un volume d'environ  $1m^3$  dans lequel l'objet à modéliser a été positionné. Pour l'application médicale, l'objet est le corps d'un patient ; l'acquisition prend 4s. et fournit environ 20000 points 3D sur l'objet, avec une précision millimétrique.
- selon un mouvement quelconque, notamment pour des dispositifs déplacés à la main par un opérateur.

Ces techniques reposent également sur un calibrage très précis.

## La stéréovision.

La stéréovision est très exploitée dans notre équipe, afin de modéliser un environnement à partir d'un robot mobile terrestre ou aérien en milieu extérieur naturel. Cette technique est également exploitée pour des tâches de modélisation d'objets : par exemple, de nombreuses entreprises de métrologie (GOM, Correlated Solutions...) remplacent progressivement les techniques à base de palpeurs par des techniques sans contact, à base de stéréovision. Citons les travaux de D. Garcia [Garcia 01], réalisés à l'Ecole des Mines d'Albi-Carmaux, sur les exploitations de la stéréovision pour l'étude du comportement de matériaux et des structures en mécanique expérimentale.

La méthode exploitée pour ce type d'applications est la stéréocorrélation, ou stéréo au niveau du pixel. Cela nécessite un calibrage très précis du capteur stéréoscopique (calibrage des deux caméras et de leur positionnement relative : base, convergence...).

Il est bien connu que la précision d'un banc stéréoscopique décroît comme le carré de la profondeur. De ce fait, nous utiliserons ce capteur pour acquérir des images à très faible distance de l'objet à modéliser. Le banc stéréoscopique que nous avons utilisé, est monté près de l'effecteur du bras manipulateur monté sur la plateforme mobile Hilare 2bis (voir figure 1.4).



FIG. 1.3 – Dispositif BODYSCAN.

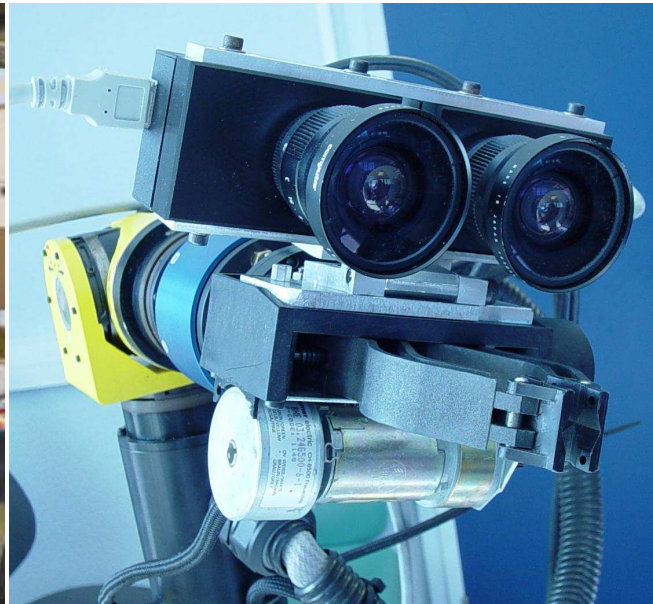


FIG. 1.4 – Tête stéréoscopique.

## Nos choix pour l'acquisition

Nous n'avons pas produit de contributions sur ce thème. A des fins de tests et d'évaluation, nous avons récupéré des images 3D déjà acquises par des télémètres laser dans des bases d'images accessibles par Internet ; nous avons aussi eu à disposition, des images 3D acquises avec le capteur Z&F par *UK Robotics* partenaire du projet CAMERA. Nous avons exploité nos propres capteurs de stéréovision disponibles dans notre équipe.

Signalons qu'acquérir ses propres images 3D à partir des caméras, n'est pas une tâche négligeable.

Afin de modéliser un objet ou un environnement, il est nécessaire d'acquérir des images depuis plusieurs points de vue. Pour ce faire, trois configurations entre scène et capteur, présentées en figure 1.5, peuvent être choisies. Ce choix est une caractéristique importante dans le système de modélisation :

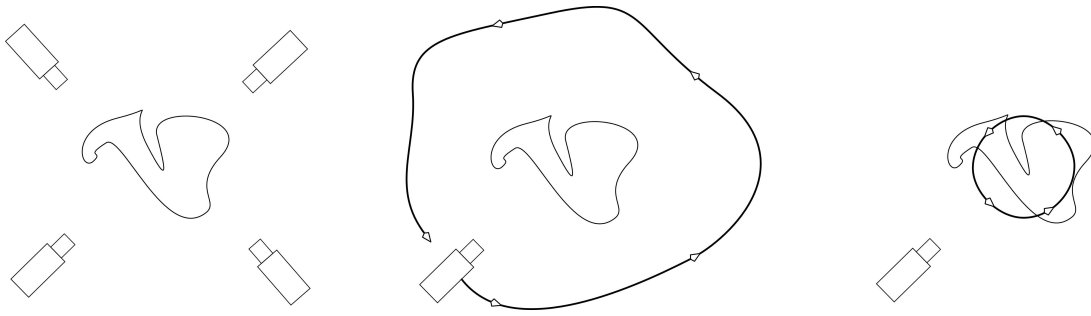


FIG. 1.5 – Variantes de capture d'images 3D.

- Plusieurs capteurs pour une scène.

Cette configuration est utilisée par exemple dans le dispositif BODYSCAN ; elle est figée, et doit être réservée à des applications dédiées... Les positions relatives inter-capteurs peuvent être estimées lors d'une phase de calibrage, ce qui permet d'exprimer directement toutes les images dans un même repère de référence.

- Un capteur mobile et une scène fixe.

Cette méthode s'applique aux capteurs embarqués sur les robots mobiles ou sur les bras manipulateurs, qui sont déplacés automatiquement par le système robotique ; l'estimée des déplacements peut être connue via les capteurs proprioceptifs des robots. Cette méthode est aussi couramment utilisée pour un déplacement manuel du capteur par un opérateur ; en ce cas, l'opérateur doit mesurer approximativement les positions successives du capteur pour simplifier le traitement de recalage des images 3D.

- Un capteur fixe et une scène mobile.

C'est une stratégie utilisée dans l'acquisition des objets de taille réduite, par exemple avec une table en rotation ou un tapis roulant.

Nous nous situons dans le cas du capteur qui est déplacé autour de la scène à modéliser. Nous n'exploitons qu'un seul capteur ; dans le projet CAMERA, pour la modélisation de monuments, des images 3D et des images couleur étaient acquises de manière indépendantes. La fusion des informations colorimétriques sur le modèle 3D requiert en ce cas un traitement spécifique afin de recalibrer entre elles deux images de nature et de résolution différentes : (voir les travaux de P.Dias [Dias 03], qui ont permis de générer le modèle texturé montré en figure 1 à droite.).

### 1.2.2 Fusion des données 3D

Ce traitement nécessite deux fonctions : le recalage des images 3D et leur fusion dans une représentation 3D, ici un maillage triangulaire : c'est le coeur de notre contribution.

Nous ne décrirons pas toutes les représentations 3D possibles : nous nous intéressons seulement à deux types de modèles,

- les ensembles de primitives surfaciques.

C'est une représentation adaptée pour des environnements ou des objets structurés. Ces primitives peuvent être extraites par diverses techniques, par exemple *clustering* ou croissance de régions. Généralement, il est délicat d'obtenir un algorithme robuste de segmentation 3D.

- les maillages triangulaires.

C'est une représentation générique, dont la résolution peut être adaptée à la complexité de l'entité modélisée d'une part, à la tâche qui va exploiter ce modèle d'autre part.

Notre objectif sera donc de proposer un algorithme qui puisse construire ces deux types de représentations à partir d'un ensemble d'images 3D recalées.

Comment effectuer cette construction ? La structure d'un système de modélisation dépend de la façon

d'organiser chaque étape de traitement à partir des données sensorielles pour produire le modèle désiré. La sélection d'une organisation globale ou incrémentale est liée à l'application et au type des images à traiter. Les diagrammes sont montrés sur la figure 1.6.

- **La modélisation globale** sépare complètement (1) l'étape de l'acquisition et du recalage des vues qui produit un nuage de points 3D et (2) l'étape suivante de construction d'une représentation géométrique à partir de ce nuage.
- **La modélisation incrémentale** exécute dans la même boucle à chaque vue, l'acquisition, le recalage et la fusion des nouvelles données par mise à jour du modèle courant. Les fonctions de recalage et fusion, peuvent être faites soit à partir de l'image 3D courante et du modèle 3D courant, soit après construction d'un modèle local, à partir des modèles local et global.

Généralement la modélisation d'un objet 3D ne nécessite au maximum, que quelques dizaines d'images ; c'est un processus dont on peut détecter la fin ; de ce fait la taille du nuage de points dans lequel toutes les données acquises sont fusionnées, reste d'une dimension finie "raisonnable". Il pourra donc être mémorisé avant construction globale d'une représentation pour cet objet.

Au contraire, pour effectuer la cartographie d'un environnement, il n'est pas possible de garder toutes les images dans un nuage pour générer un modèle a posteriori. La vue courante sert à augmenter le modèle actuel. Une solution intermédiaire peut consister à fusionner les vues dans un modèle local, par exemple pour acquérir le modèle d'une pièce dans un bâtiment.

Dans notre système, nous souhaitons rester flexibles, afin d'appliquer soit une approche incrémentale, soit une approche globale, selon le contexte, le type de données. . .

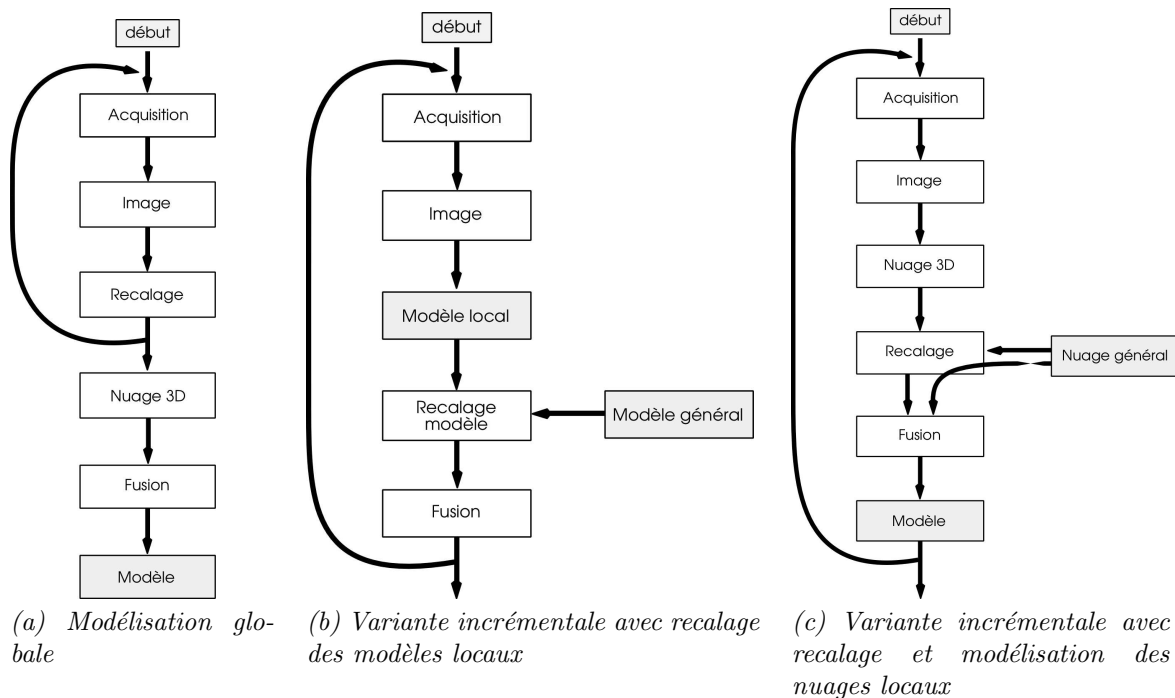


FIG. 1.6 – Structures des systèmes de modélisation.

### 1.2.3 Contrôle de la tâche de modélisation

Nous nous sommes intéressés à la planification globale de la tâche de modélisation. Cette fonction est compatible seulement avec une approche incrémentale de modélisation : après chaque mise à jour du

modèle courant, nous pouvons déterminer en quelle position mettre le capteur pour compléter ce modèle, le but pouvant être

- de minimiser le nombre de vues nécessaires pour modéliser une entité ;
- ou d'atteindre une résolution donnée dans ce modèle. . .

### 1.3 Modélisation 3D dans d'autres domaines.

La modélisation 3D est une thématique qui peut être abordée sous des angles différents, selon le but envisagé ou les types de données à utiliser. Nous évoquons ici deux thématiques proches à notre structure, mais sur lesquelles nous n'avons pas travaillé.

#### Modélisation 3D et SLAM

L'exploration d'environnements par des capteurs montés sur un robot mobile, requiert de traiter simultanément

- de la localisation du robot à partir d'appariements entre données sensorielles courantes et amers détectés lors des vues précédentes et mémorisés dans le modèle courant de l'environnement ;
- et aussi, de mettre à jour ce modèle en y intégrant les données sensorielles courantes, notamment détecter dans ces données, des amers non encore perçus et les mémoriser dans le modèle.

Cette fonction d'exploration a donc pris le nom de SLAM, pour *Simultaneous Localization and Mapping*. Les techniques employées pour traiter de ce problème, utilisent des représentations stochastiques et des techniques de filtrage tenant compte des imprécisions sur les données, typiquement Filtrage de Kalman Etendu, Filtrage Particulaire ...

La thématique SLAM est liée à la navigation d'un robot mobile ; elle exploite une approche incrémentale ; elle a pour objectif la construction d'une carte d'amers, carte souvent éparsée pour limiter la complexité. Elle maintient un modèle des imprécisions sur toutes les données intervenant dans la fonction.

Dans le contexte *Modélisation d'objets*, nous souhaitons rester flexible quand au type d'approches à mettre en oeuvre. Généralement, les modèles de type maillage ne portent pas avec eux des informations sur l'incertitude ou l'imprécision. Enfin ces modèles sont denses, alors que les modèles construits par les fonctions SLAM, sont généralement épars (carte d'amers).

#### Modélisation d'objet et SFM

Une approche de modélisation d'objet qui suscite un grand intérêt depuis plusieurs années dans la communauté Vision, se fonde sur la vision monoculaire. Avec une caméra mobile, il est possible de modéliser une scène 3D, à un facteur d'échelle près ; ce sont les techniques regroupées sous l'appellation *Structure From Motion*. Notons que les performances des techniques de base exploitées par ces méthodes (optimisation non linéaire pour résoudre le problème de l'ajustement des faisceaux, ou *Bundle Adjustment*) ne sont pas à l'heure actuelle, compatibles avec les contraintes liées à la robotique.

Malgré les progrès récents de ces techniques, y compris dans le cadre du projet CAMERA (travaux de J.Santos-Victor et de son équipe à Lisbonne), nous ne les avons pas exploitées.

### 1.4 Cahier des charges de notre système de modélisation.

En conclusion de ce chapitre introductif, nous pouvons spécifier les caractéristiques du système de modélisation d'objet qui pourrait être intégré dans les machines autonomes :

- acquisition avec un capteur monté sur une plateforme mobile, typiquement la stéréovision, car la texture pourra être directement associée aux données 3D sans traitement supplémentaire.
- les fonctions de recalage, fusion des images 3D et construction du modèle doivent pouvoir être exécutées selon une approche globale ou incrémentale.



- la précision d'un modèle doit permettre a posteriori, de localiser l'objet en 3D avec une erreur maximale de quelques millimètres, de sorte que la saisie pourra être planifiée.
- la résolution du modèle doit pouvoir être adaptée, pour pouvoir accélérer des traitements si besoin, en sous-échantillonnant les images 3D.
- le modèle doit permettre d'intégrer différents types de données 3D, acquises par télémétrie, stéréo, etc.
- la méthode de construction doit prendre en compte le bruit de mesure sur les données sensorielles, bruit qui est non uniforme pour des données stéréo, et qui peut être important.

Dans le chapitre suivant, nous décrivons la première fonction nécessaire dans notre chaîne de traitement : le recalage entre les vues.

# Chapitre 2

## Recalage des images 3D

### 2.1 Introduction.

La construction du modèle d'un objet ou d'un environnement, requiert l'acquisition d'un grand nombre d'images 3D, acquises depuis diverses positions du capteur. Chaque image est un ensemble de points 3D, connus dans le repère du capteur ou du support sur lequel il est rigidement lié (platine, bras, plateforme...). L'organisation d'une image 3D dépend de la technique d'acquisition : pour la stéréovision ou la télémétrie laser deux axes, il s'agit d'une organisation matricielle. Deux points 3D voisins dans la matrice, sont portés par des rayons optiques voisins dans l'espace.

Pour chaque image 3D, le **recalage** (*registration*) consiste à déterminer précisément, en quelle position de l'environnement elle a été acquise. Cela revient à exprimer chaque image dans un repère fixe par rapport à l'environnement, typiquement, la position du capteur lors de la première acquisition. Le recalage s'applique en général avant la phase de modélisation, au niveau des images de points 3D ; une fois connue la position d'acquisition de chaque image, l'ensemble des points peut être fusionné dans un **nuage** de points 3D, dans lequel toute structuration liée à l'acquisition est perdue.

Selon la stratégie, le recalage s'effectuera entre images successives, ou entre chaque image et une partie adéquate du nuage résultant de la fusion de toutes les images acquises précédemment.

Le recalage est traité par un algorithme de type ICP, pour *Iterative Closest Points*. La figure 2.1 présente les différentes étapes du processus ICP. Tout d'abord, pour obtenir une convergence plus rapide de cet algorithme, un prétraitement doit être appliqué sur chaque image 3D à recaler, pour sélectionner des points représentatifs sur lesquels ICP sera appliqué : les points de discontinuité 3D, les sommets d'un maillage triangulaire généré à partir de ces contours ou de l'image ou toute autre méthode de sous-échantillonnage.

Ensuite, le processus se compose de 3 parties clairement séparées : l'étape d'appariement de points, génère un ensemble de paires de points communs entre les nuages à recaler. Cette étape est critique, car de cette sélection dépend les résultats suivants. Le temps de calcul associé est aussi le plus important pour l'ensemble des essais effectués. De nombreux critères permettent de trouver les points communs entre les vues, comme par exemple :

- la distance entre points de mesure,
- la comparaison des normales estimées en chaque point d'une part, ou des normales et orientations (spin) d'autre part,
- l'utilisation de polygones associés aux points,
- l'identification des contours passant par ces points...
- la comparaison d'informations photométriques liées aux points (luminance, couleur...).

Dans une seconde étape, si la distance moyenne entre les points appariés est plus grande à un seuil donné, il faut estimer la transformation (donc, une rotation et une translation) entre les repères dans

lesquels sont exprimées ces points. Si les nuages à recaler sont deux images successives, il s'agit d'estimer le mouvement du capteur entre les deux acquisitions; en robotique mobile, on parle alors d'odométrie optique [Mallet 00] *Visual Motion Estimation*. L'estimation de la meilleure transformation liant deux ensembles de points rigidement liés, est obtenue par minimisation d'un critère de moindres carrés : on peut utiliser les méthodes classiques itératives, notamment les approximations adaptées au problème spécifique d'obtenir une rotation et une translation. Il existe des solutions fondées sur SVD ou sur la représentation de la rotation par un quaternion ou un double quaternion entre autres.

La dernière étape consiste en l'application de la transformation sur les points d'un des nuages pour les faire converger vers les points du second. Puis, l'algorithme ICP reboucle vers la première étape, jusqu'à convergence.

La section suivante propose un état de l'art sur cet algorithme de recalage. Puis, en section 2.3, nous donnons une description plus précise de la méthode, avant d'analyser chacune de ces étapes : sélection des points de contrôle en section 2.4, recherche d'appariement entre ces points en section 2.5, estimation de la transformation entre les deux nuages à recaler en section 2.6 et contrôle d'exécution pour accélérer la convergence en section 2.7. Nous décrivons quelques variants en section 2.8. Enfin, en section 2.9, nous évaluons les performances de notre version de ICP sur un ensemble d'images 3D téléchargées depuis le Web ou acquises depuis nos propres capteurs; le but de cette première évaluation est de comparer le comportement de ICP selon le type de points de contrôle.

## 2.2 État de l'art.

La méthode de recalage des images de profondeur nommé algorithme d'ICP, a été introduite initialement en 1991 et 92 par Chen et Medioni [Medioni 91] et Besl et McKay [Besl 92]; cette méthode a été étudiée par de nombreux chercheurs jusqu'à présent et plusieurs variantes ont été proposées. Dans les paragraphes suivants nous présentons les travaux qui nous ont paru les plus significatifs.

La première version de l'algorithme ICP, proposée dans [Medioni 91], utilise les surfaces tangentes pour l'appariement. Par contre, la méthode de Besl et Jain [Besl 92] exploite de manière plus élégante et plus globale, points, lignes, triangles et courbes; par ailleurs, le calcul de la transformation est fondé sur le formalisme des quaternions. Dans ces premières versions, l'algorithme ICP est appliqué pour le recalage fin entre deux nuages de points, donc à partir d'une estimée correcte de la transformation recherchée.

Blais et Levine [Blais 95] sont les premiers à faire un recalage entre plusieurs vues en utilisant l'algorithme d'ICP. Un autre travail préliminaire sur le recalage a été proposé par Kamgar-Parsi, Jones et Rosenfeld [Kamgar-Parsi 91] : recalage de données 3D sur les fonds marins en utilisant les lignes de niveau. Zhang [Zhang 94] adapte cette méthode au recalage de lignes pour la navigation des robots mobiles sur terrain accidenté; une contribution importante est le filtrage statistique afin de rendre plus discriminant le critère d'appariement fondé sur la distance entre points.

A l'Université de Carnegie Mellon, dans l'équipe de M. Hebert, Johnson, Huber et Carmichael [Johnson 97a, Johnson 96, Johnson 99, Zhang 99], travaillent sur la construction incrémentale de modèle dense d'environnements à partir d'images de profondeur acquises par des télémètres laser deux axes (Acuity Research ou Zoeller&Froelich) : ils ont introduit le concept de "spin" afin d'améliorer la technique d'appariement entre points. Ce concept a été utilisé aussi sur des thèmes connexes comme la modélisation de scènes d'intérieur [Hebert 95], ce qui nécessite de traiter de grandes quantités de données [Hebert 98, Carmichael 99, Johnson 95, Johnson 97b]. Cette méthode a été exploitée pour la navigation [Hebert 01] et pour le recalage de formes [Simon 96], [Huber 99] avec des données acquises par différents capteurs. Le problème de l'estimée initiale de la transformation pour qu'ICP converge, est traité dans les travaux sur le recalage automatique par [Higuchi 93], [Huber 01a], [Huber 02], [Huber 01b], [Huber 00], [Huber 03].

À Stanford, Levoy, Curless, Rusinkiewicz et Pulli [Levoy 00, Rusinkiewicz 01, Curless 96, Pulli 99,

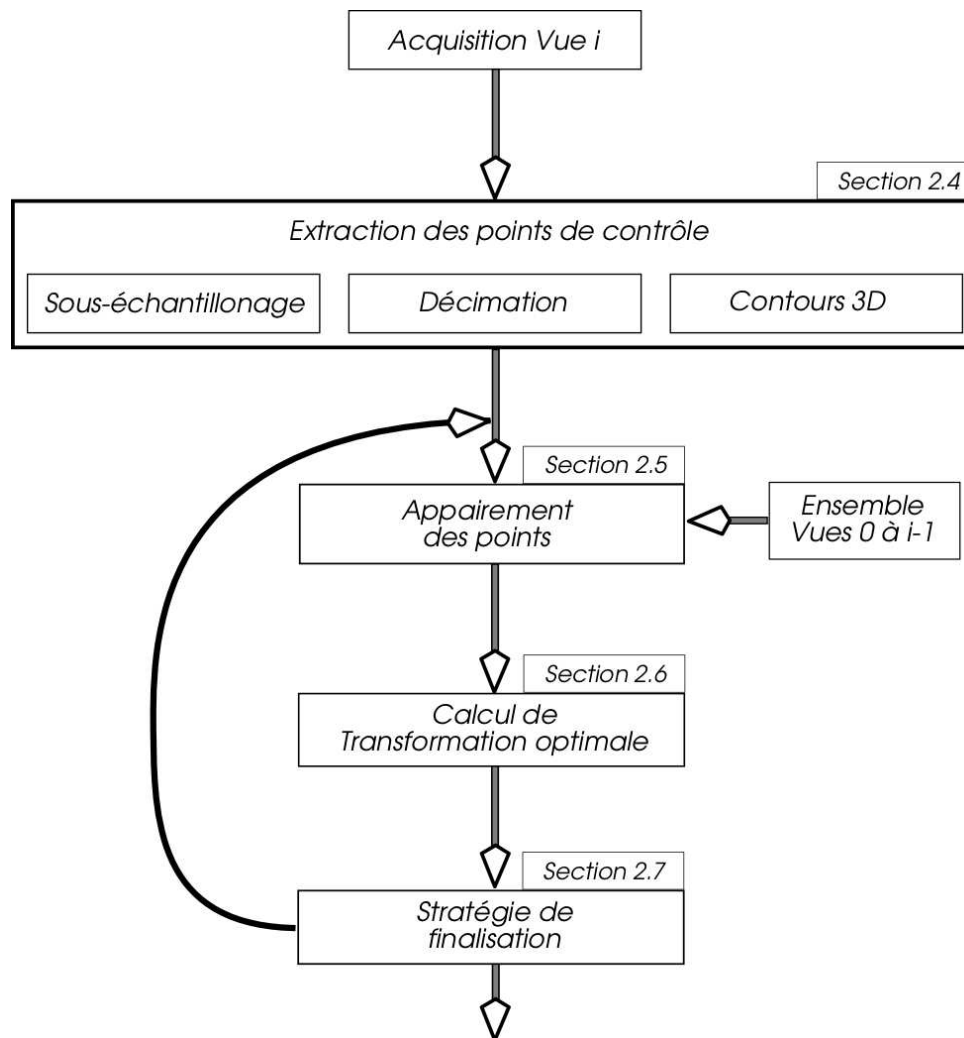


FIG. 2.1 – Diagramme du processus d'ICP.

Pulli 97, Pulli 98, Gelfand 03] se sont intéressés à la reconstruction détaillée de statues à partir de données laser : “The Digital Michelangelo Project” est à l’origine des images du lapin BUNNY et des statues BUDHA et DRAGON, utilisées constamment comme données de test pour la reconstruction et le traitement des images 3D. Ils ont contribué sur la stratégie exploitée dans les méthodes de reconstruction, l’amélioration des performances de la méthode d’ICP et le traitement multi-vues, ce qui nécessite le traitement et la fusion de quantités massives de données. Dans une version multi-vues d’ICP, toutes les transformations par paire sont estimées en même temps.

Fisher, Eggert et Stoddart [Fisher 04, Robertson 02, Rodrigues 02, Eggert 96], [Stoddart 96a, Stoddart 96b] de l’université d’Edinburgh, ont contribué dans le recalage des images de profondeur appliqué au “reverse engineering” et au développement de méthodes multivues pour la fusion et la modélisation ; le capteur 3D qu’ils ont réalisé, appelé *hand-held optical sensor*, est tenu par l’opérateur, qui peut le déplacer autour de l’objet à modéliser ; des capteurs inertiels embarqués donnent les estimées nécessaires pour initialiser ICP.

Penec, Thirion et Ayache de l’INRIA Sophia-Antipolis [Penec 95, Penec 96, Penec 97b, Penec 97a, Granger 02], ont appliqué la méthode ICP en imagerie médicale, pour la reconstruction

des images de résonance magnétique. Leurs contributions portent sur l'exploitation de méthodes robustes pour l'appariement et l'estimation, l'intégration multivues et l'amélioration des performances. Cette méthode procède par appariement de courbes et de surfaces adaptées aux images volumiques .

Citons aussi le "Marching Lines Algorithm" [Thirion 93b, Thirion 92, Thirion 93c, Thirion 96, Thirion 93a, Thirion 95] et des travaux sur le recalage sur objets non-rigides [Feldmar 97, Feldmar 96, Feldmar 94, Gourdon 93].

Dans le même cadre d'application, Sharp à Harvard [Sharp 99, Sharp 01, Sharp 02b, Sharp 02c, Sharp 02a, Sharp 02d] a proposé une méthode de recalage automatique par ICP pour l'intégration multivues, en utilisant divers types de caractéristiques invariantes et le traitement des occultations.

Dorai, Weng et Jain, à MSU [Dorai 97, Dorai 98], ont contribué aux travaux sur la fusion et recalage multivues simultanées.

Au Canada, à l'Université de Laval et VITG, Bergevin, Laurendeau, Godin entre autres ont travaillé sur le recalage multivues [Burtnyk 95, Bergevin 96, Bergevin 95, Tubic 03], sur une version parallèle de l'algorithme ICP [Langis 01], et sur l'optimisation de la stratégie pour la recherche d'appariements [Greenspan 01, Greenspan 03, Godin 01].

Au LAAS-CNRS, nous avons travaillé sur la modélisation d'environnements, l'extraction des discontinuités pour produire des contours 3D [Sappa 02, Sappa 01c, Sappa 01b], la génération de maillage [Bozier 00b] et la navigation en milieu naturel à partir de la stéréovision dense [Jung 04].

Citons d'autres travaux sur ICP dans divers domaines :

Intégration multivues :	[Hung 99], [Nishino 02, Werner 02], [Williams 00, Okatani 02, Masuda 02], [Neugebauer 97, Torr 98].
Appariement de surfaces :	[Benjemaa 97, Xuan 97, Chua 96], [Yamany 99, Yamany 98, Ahmed 97], [Schoenenberger 00, Fischer 99], [Maier 03].
Par caractérisation des invariants :	[Wyngaerd 99], [Wyngaerd 02].
Traitement d'appariements :	[Dalley 02, Sun 03, Keller 03].
Appariement et reconstruction :	[Murino 01, Pritchett 98], [Rodriquez 90], [Yang 98, And 98, Liu 03, Chen 98].
ICP à partir de Stéréo :	[Morency 03, Morency 02].
Appariement d'information de couleur ou lumière :	[Kim 04, Weik 97].
Proposition de variantes de l'algorithme ICP :	[Liu 04, Hähnel 03, Ohta 98], [Zinßer 03, Chetverikov 02, Simon 96], [Luck 00].

## 2.3 Description de la méthode.

La fusion consiste à réunir en un seul modèle global les informations obtenues sur un ensemble de modèles locaux, générés depuis des positions différentes. Un modèle sera donc construit de manière incrémentale. A l'instant  $t$ , le modèle courant intégrant les données acquises jusque là, est appelé  $M_{t-1}$  ; un nouveau point de vue est sélectionné, le capteur  $y$  est positionné et une acquisition produit un modèle local  $m_t$  ; ce modèle doit être recalé avec  $M_{t-1}$ , puis fusionné avec ce modèle pour produire le nouveau modèle global courant  $M_t$ . Ce processus s'achève à l'instant *final*, lorsque le modèle global  $M_{final}$  satisfait un critère d'accomplissement : c'est le modèle final. Dans ces conditions  $M_{final}$  intègre ainsi l'ensemble des informations des différents modèles locaux  $m_0 \dots m_{final}$ .

Nous ne traitons ici que du recalage entre un modèle local  $m_t$  et le modèle global courant  $M_{t-1}$ , ou entre les modèles locaux acquis depuis deux vues successives ( $m_t$  et  $m_{t+1}$ ) ou plus généralement, acquis depuis deux positions depuis lesquelles les champs de vue sont en recouvrement ( $m_i$  et  $m_j$ ). La transformation  $T_t$  entre les positions d'acquisition aux instants  $t - 1$  et  $t$ , peut être estimée par des capteurs proprioceptifs, tels que des codeurs sur les actionneurs qui permettent de déplacer le capteur, ou des capteurs inertiels portés par le capteur 3D (comme dans un *Hand-held Sensor*) ; dans certains cas,

notamment lorsque le capteur est déplacé manuellement dans des grands environnements (modélisation de sites industriels ou de monuments à partir de données laser), cette estimée est évaluée par l'opérateur, donc est très approximative.

Cette estimation de  $T_t$ , n'est généralement pas suffisante pour effectuer directement la fusion de modèles; sans recalage, le modèle global aurait une précision insuffisante. Il faut donc affiner l'estimée initiale par un processus de recalage, qui consiste à mettre en correspondance des attributs significatifs du modèle local et du modèle global courant, et à calculer la transformation 3D minimisant une mesure de différence entre ces attributs. La méthode développée dans nos travaux a les caractéristiques suivantes :

- nos modèles sont des images (local  $m_t$ ) ou des nuages (global  $M_t$ ) de points comme dans [Medioni 91, Zhang 94, Chen 92]
- les attributs mis en correspondance sont des points de contrôle,
- la différence entre les divers points de vue est mesurée par la distance euclidienne entre les points de contrôle appariés.

Le recalage par l'algorithme ICP, est une méthode itérative, qui a pour but d'améliorer l'estimée sur la transformation  $T_t$  à partir de l'estimée initiale  $T_t^0$ . A l'itération numéro  $k$ , la position du point de vue courant est estimée par une transformation  $T_t^k$ . Un ensemble de points de contrôle ( $p_t^i, i = 1 \dots$ ) est extrait sur le modèle local  $m_t$ . Nous leur appliquons la transformation  $T_t^k$  afin de les exprimer dans le repère du modèle global; pour chacun point  $p_t^i$ , on recherche, selon divers critères discriminants, le point  $P_{t-1}^i$  le plus proche sur l'ensemble des points de contrôle déjà fusionnés dans le modèle  $M_{t-1}$ . La transformation  $T_t^k$  peut alors être réestimée à partir de ces appariements :

- on évalue tout d'abord la transformation  $\Delta T^k$  qui permet de superposer, au mieux, les points appariés, par minimisation aux moindres carrés de l'erreur de superposition :

$$\sum_i |\Delta T^k p_t^i - P_{t-1}^i|^2$$

- on met à jour l'estimée de la transformation en faisant la composition de l'estimée précédente avec l'erreur de superposition :

$$T_t^{k+1} = T_t^k \Delta T^k$$

Formellement, les transformations  $T$  et  $\Delta T$  sont ici représentées par des matrices homogènes, tandis que les points  $p$  et  $P$  sont connus par leurs coordonnées homogènes. Ces opérations sont répétées itérativement jusqu'à la superposition satisfaisante des modèles.

Comme indiqué dans la section précédente, de nombreuses optimisations de cet algorithme ont été proposées, notamment dans les travaux de [Hebert 98], afin d'accélérer la convergence et d'en améliorer la précision. En particulier, chaque point est non plus apparié au plus proche de l'autre modèle au sens de la distance euclidienne, mais à celui dont le relief local est le plus proche dans le voisinage considérée. L'appariement est donc calculé par une minimisation de la distance euclidienne d'une part et un calcul de corrélation d'autre part.

Si  $N$  et  $M$  sont les nombres de points de contrôle respectivement extraits du modèle local  $m_t$  et fusionnés dans le modèle global  $M_{t-1}$ , alors la complexité maximale pour la recherche des appariements est  $N.M$  pour chaque itération; l'exploitation de contraintes de voisinage, de visibilité, d'unicité ... permettent de limiter cette complexité. Néanmoins, un facteur clé pour améliorer la performance, est la limitation du nombre  $N$  de points extraits depuis un modèle local (dans notre cas, une image 3D); pour ce faire plusieurs techniques d'extraction de points de contrôle sont présentées dans la section suivante; elles seront comparées en fin de ce chapitre.

## 2.4 Sélection des points de contrôle.

Nos données 3D sont denses; la résolution des images dépend du capteur. Dans le projet CAMERA, nous avons traité des images acquises par un télémètre Z&F : plus de 10 millions de points 3D par image (voir une sous-image en figure 2.2). Il est donc indispensable d'extraire un ensemble de points de contrôle

sur lesquels la méthode ICP sera appliquée. Nous avons comparé différentes méthodes de sélection de points depuis une image 3D.

1. nous avons utilisé d'abord les points de discontinuité extraits directement depuis l'image, répertoriés dans une image binaire appelée "*binary map*". D'autres auteurs exploitent aussi ces points pour extraire des contours 3D (par chaînage des points connexes dans le "*binary map*"), puis des segments de courbe 3D (droite, quadrique, B-spline...) par approximation des contours 3D ; citons en particulier les travaux de Guéziec&Ayache [Guéziec 92] sur le recalage d'images 3D par appariements de contours approximatés par des B-splines.

Dans notre approche, pour éviter les effets des occultations, seuls les points de discontinuité de la normale sont retenus ; ce sont de vrais points de mesure, mais comme ils sont sur les discontinuités, ils peuvent être plus bruités que les points acquis sur les surfaces.

2. nous avons ensuite essayé de construire un maillage triangulaire à partir de l'image, de décimer ce maillage à une résolution réglable, et d'exploiter pour le recalage, les sommets ou les centres de triangles : ils sont directement présents au sein du modèle et sont donc très facilement utilisables. Par contre, selon la méthode de construction du maillage ou de décimation, leur répartition n'est pas uniforme dans l'image, ce qui peut introduire un biais de recalage, non quantifiable.
3. enfin les deux méthodes précédentes ayant un coût non négligeable, nous avons essayé un simple sous-échantillonnage des points de mesure présents dans les images à recalcer.

### 2.4.1 Extraction des contours 3D.

Cette étude a été menée conjointement avec A.Sappa dans le cadre du projet CAMERA, dédié à la modélisation de bâtiments à partir de données de profondeur. Un des défis majeurs en ce domaine, consiste à générer automatiquement des représentations de type ensemble de primitives surfaciques planes, cylindriques, sphériques, ... Aujourd'hui, les outils disponibles dans les entreprises spécialisées permettent uniquement une génération interactive de telles représentations ; dans le cadre du projet CAMERA, nous avons exploité le progiciel LFM (*Light Form Modeller*) de la société UK Robotics, devenue depuis la filiale britannique de la compagnie Z&F. La segmentation robuste en régions reste un problème, malgré un grand nombre de travaux dans la communauté scientifique [Hoover 96]. La méthode présentée ci-dessous a été intégrée dans LFM par A.Sappa [Sappa 01a], et a été exploitée afin d'initialiser des régions élémentaires, dites *region seeds*, à partir desquelles une technique de croissance de régions peut être exécutée pour générer un ensemble fiable de primitives surfaciques.

Pour aborder le problème de la segmentation 3D d'une image, nous nous sommes donc intéressés aux méthodes qui se fondent sur l'extraction des contours 3D. Ces méthodes consistent (1) à rechercher les points de discontinuité dans l'image, puis (2) à connecter ces points pour en extraire des chaînes de

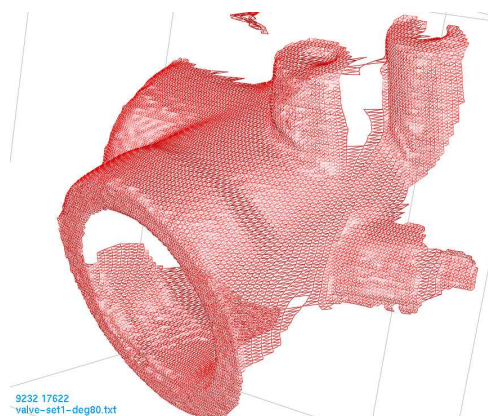


FIG. 2.2 – Données 3D denses d'origine.

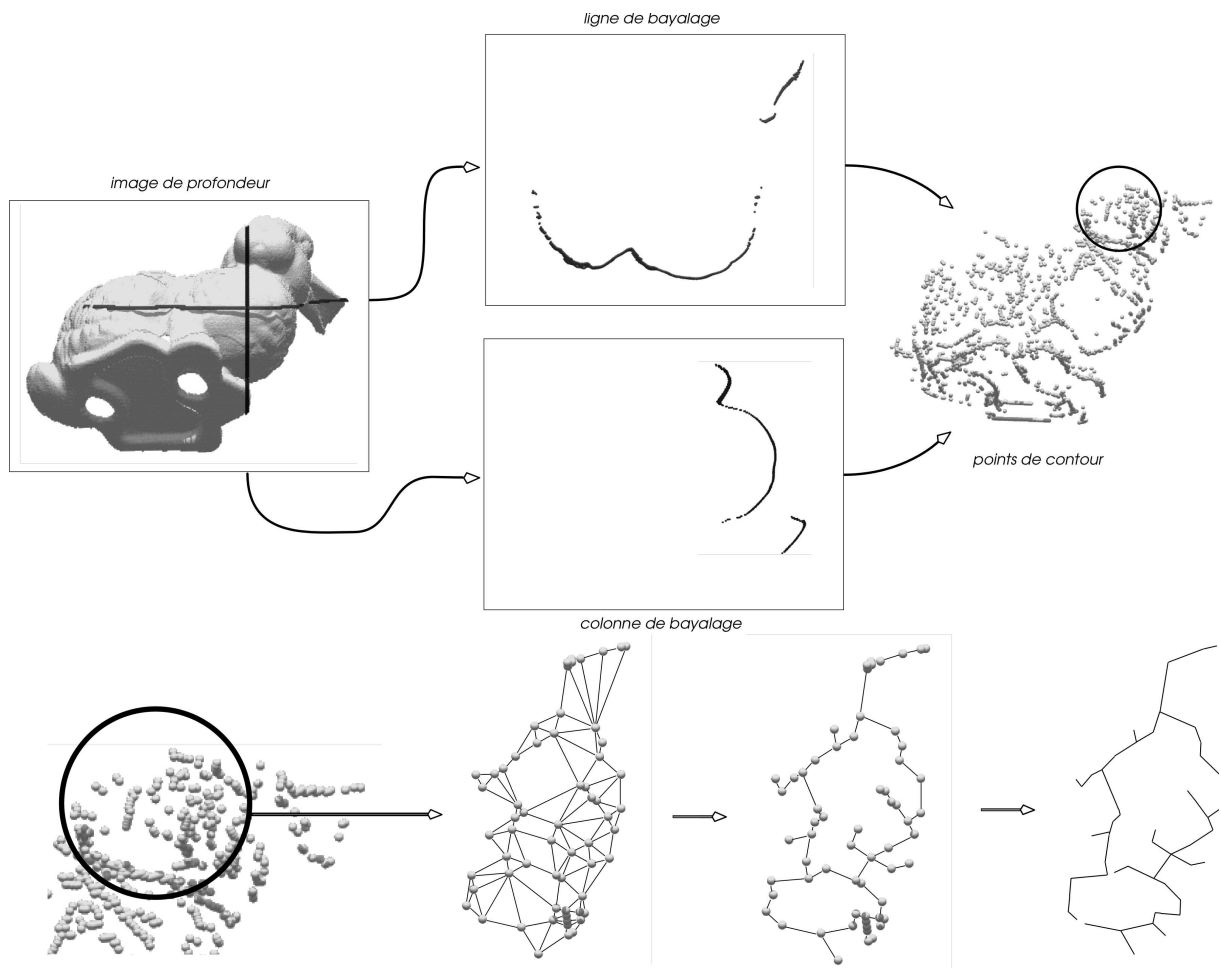


FIG. 2.3 – Etapes de l'extraction des contours 3D.

contours 3D. Dans le cadre de notre travail, nous décrivons uniquement la phase d'extraction des contours 3D.

Notre approche est illustrée sur la figure 2.3. Tout d'abord, nous recherchons les points des discontinuités en analysant successivement chaque ligne et chaque colonne de l'image 3D (en haut à gauche), comme l'a proposé X. Jiang dans [Jiang 99]. Sur chaque ligne ou colonne (en haut au centre), nous détectons d'abord les points de discontinuité de profondeur; puis chaque tronçon entre de tels points, est approximé par des segments de courbe quadrique; les points de raccordement entre ces segments sont les points de discontinuité de normale. Tous les points de contour sont marqués dans une image binaire, appelée *binary map* (en haut à droite). Ensuite, nous construisons un maillage sur ces points (en bas à droite) et recherchons un arbre de recouvrement minimal de ce maillage (en bas au milieu), arbre filtré par des opérateurs morphologiques (en bas à gauche); une arête dans l'arbre est un contour 3D.

### Recherche des points de discontinuité.

Chaque ligne et colonne de l'image 3D est traitée, afin d'y détecter des points de discontinuité; ces points sont ensuite marqués dans une image binaire ou *binary map*, qui sera la donnée d'entrée du traitement suivant.

Plusieurs travaux ont traité de ce type d'approche qui considère l'image comme un ensemble de *scan lines* [Jiang 94, Jiang 98, Bellon 99]. Dans notre configuration où le plan horizontal correspond



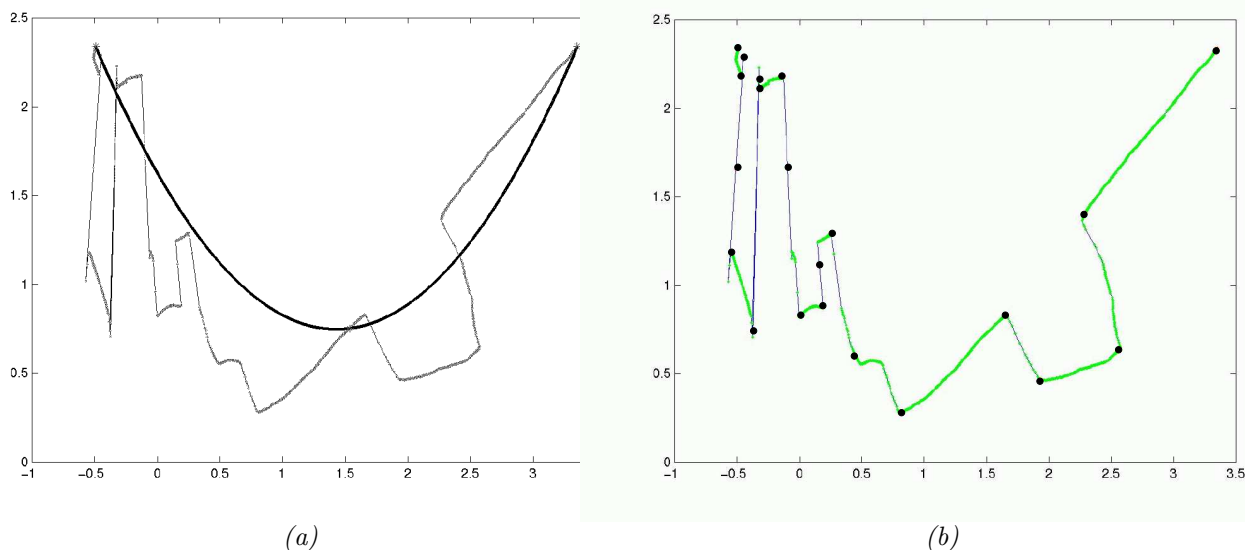


FIG. 2.4 – (a) Une colonne de l’image et l’approximation initiale par une parabole. ; (b) Discontinuités de normale retenues.

approximativement à l’angle site égal à 0, il suffit de traiter deux directions pour saisir tous les points de discontinuité de l’image. Donc, chaque ligne et chaque colonne est successivement traitée comme un vecteur 1D. Sur ce vecteur, nous réalisons deux opérations : tout d’abord, nous détectons les discontinuités de profondeur (ou *jump edge*) et filtrons les points aberrants, souvent proches de ces ruptures de surface : dans le cas de la télémétrie laser, ce sont souvent des points de moyennage, obtenus quand le faisceau laser chevauche une discontinuité de profondeur.

Puis, pour chaque séquence de points comprise entre deux discontinuités de profondeur, nous calculons une approximation par un ensemble de segments quadriques (pour simplifier, uniquement paraboliques), les points de connexion entre deux segments sont des discontinuités de normale (ou *crease edge*). Cette segmentation utilise la technique classique de la corde, ou du découpage itératif de la séquence initiale. Les positions des discontinuités de profondeur et de normale sont ensuite marquées dans la *binary map*.

Sur la figure 2.4, nous présentons les points de contour retenus par notre algorithme pour une colonne de l’image. Dans la plupart des cas, un point de discontinuité sera trouvé deux fois, lors des traitements de la ligne et de la colonne auxquels il appartient : néanmoins il convient de traiter les deux orientations, car lors du parcours des lignes, un contour situé dans le plan de balayage laser ne serait pas détecté. Après le parcours des lignes et des colonnes, dans la *binary map*, tous les points des arêtes physiques seront marqués (voir la figure 2.7 générée à partir de la sous-image présentée sur la figure 2.6).

Notons que cette recherche des discontinuités permet de s’affranchir en grande partie, du bruit dans les images : soit des artefacts propres aux discontinuités de profondeur, soit du bruit sur les points 3D internes aux surfaces, car l’approximation par une quadrique a un effet de lissage sur les données brutes.

### Extraction des contours 3D.

A partir de l’image binaire des points de discontinuité, nous construisons tout d’abord un maillage triangulaire, en appliquant des contraintes de Delaunay en 2D (dans le plan image), pour limiter le nombre de triangles dégénérés. Puis nous filtrons toutes les arêtes de ce maillage qui ont une longueur supérieure à un seuil variable. De ce fait, les arêtes du maillage connectent uniquement des points proches sur les discontinuités.

Nous transformons ce maillage en un graphe : un sommet du maillage donne un sommet du graphe, une arête du maillage donne une arête du graphe, étiquetée par la distance 3D entre les sommets qu’elle lie. Chaque composante connexe de ce graphe de connexité est ensuite traitée par l’algorithme de Kruskal [Rosen 90], afin de rechercher l’arbre de recouvrement minimal (ou *Minimum Spanning Tree*). Cet

algorithme donne le chemin de coût minimal reliant les points de contour.

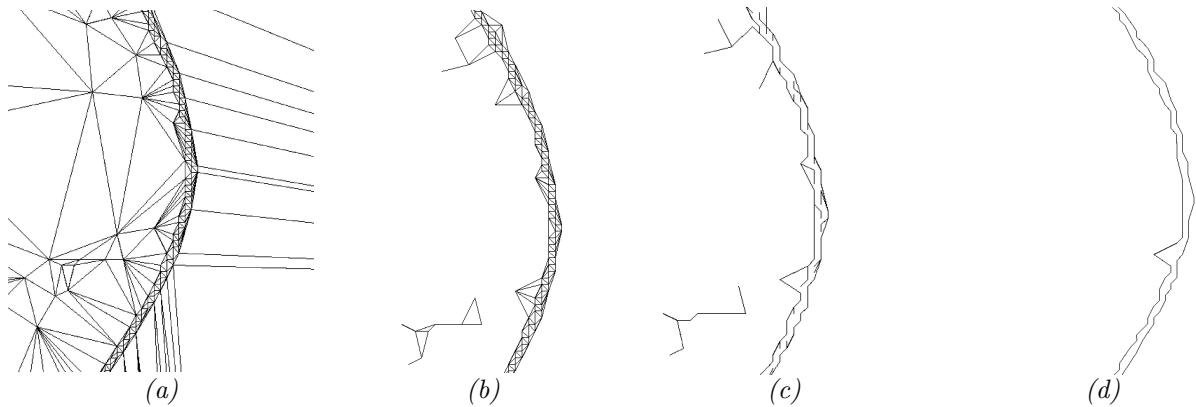


FIG. 2.5 – Vues de détail sur l'extraction des contours 3D.

Sur la figure 2.5, nous illustrons cette deuxième étape sur un contour correspondant à une discontinuité de profondeur. La figure (a) est un zoom sur une arête et la partie du maillage triangulaire complet qui s'y rattache. La figure (b) présente le résultat du filtrage des arêtes sur ce maillage ; il existe deux composantes connexes ; seules restent connectés les sommets connexes sur les contours. La figure (c) donne les arbres de recouvrement de coût minimal sur ces deux sous-graphes, avec un arbre isolé correspondant à des fausses discontinuités, et deux arbres quasiment parallèles pour les deux côtés de la discontinuité de profondeur : l'algorithme de Kruskal n'a pas retenu de chemin joignant des points éloignés qui sont de part et d'autre de la discontinuité. Sur la figure (d) enfin, l'arbre isolé a été filtré car il ne comportait pas un nombre suffisant de sommets, et les deux arbres principaux ont été simplifiés par des opérateurs de type érosion-dilatation, ce qui a permis la suppression des barbules. Nous pouvons ensuite extraire deux contours, correspondant aux deux côtés de la discontinuité de profondeur.

### Essais.

Cet algorithme a été testé sur des images de haute résolution acquises sur une église. Sur les figures 2.7 et 2.8, notre méthode est appliquée sur l'image panoramique de taille 1400 lignes par 8000 colonnes ; nous présentons uniquement les résultats sur la sous-image centrée sur l'autel, présentée sur la figure 2.6. De nombreux points de contour sont détectés, soit pour des discontinuités de normale sur les murs, soit pour des discontinuités de profondeur liées à des objets posés au sol (chaises, cierges autour et sur l'autel, barrière devant l'autel, ...).

Les contours 3D extraits par notre méthode sont localisés sur ces discontinuités et le filtrage des contours permettent de simplifier le résultat. Notons que ce filtrage devrait être amélioré pour éviter la suppression d'arêtes 3D réelles (les deux arêtes verticales à droite de la sous-image, devant la barrière).

### 2.4.2 Construction d'un maillage triangulaire irrégulier.

La deuxième méthode proposée pour extraire d'une image 3D, un nombre réduit de points de contrôle, exploite une modélisation locale sous la forme d'un maillage. Les maillages peuvent s'adapter à tout type de surface, donc quelle que soit la forme ou la topologie. Nous souhaitons ici

- construire seulement un maillage depuis une image 3D, problème simple par rapport au problème de la modélisation incrémentale de l'environnement, traité dans notre équipe, par J.Bozier [Bozier 00a, Bozier 00b].
- extraire des points qui soient des points de mesure, ce qui exclut les méthodes fondées sur des surfaces actives ou déformables (voir le chapitre suivant), du reste trop complexes pour nos besoins, puisqu'elles permettent de construire un maillage à partir d'un nuage de points 3D.

Dans une image 3D, les relations de voisinage entre points, sont explicites : en connectant les points entre eux, on peut obtenir à peu de coût, un maillage cohérent (sans repliement ou triangle malformé).



FIG. 2.6 – Sous-image centrée sur l'autel, d'une image de profondeur acquise dans une église.

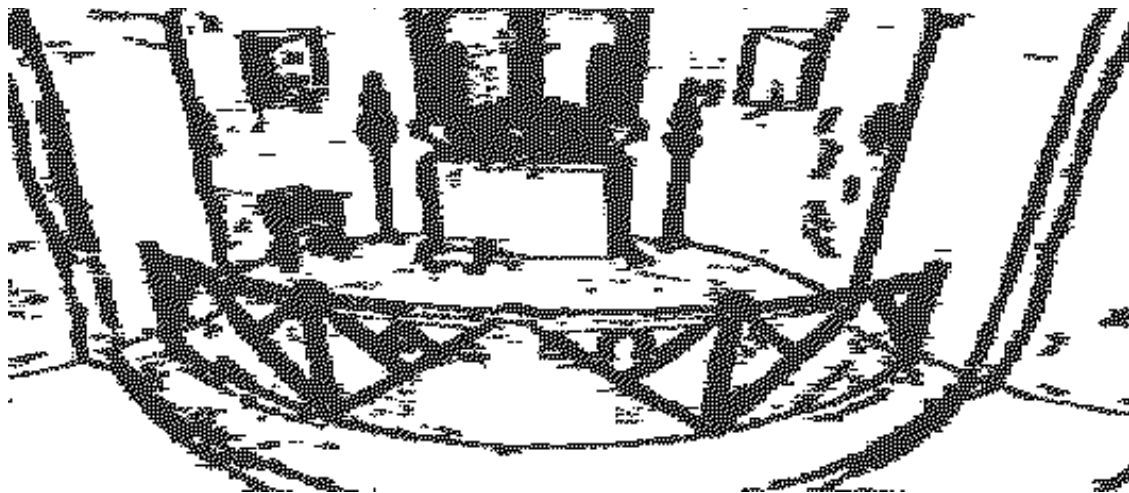


FIG. 2.7 – Ensemble des points de contour extraits sur la sous-image centrée sur l'autel

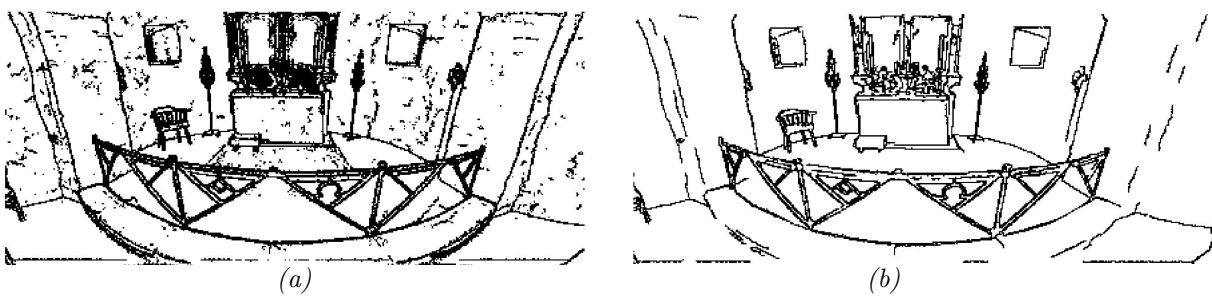


FIG. 2.8 – Résultat de la segmentation en contours : (a) arbre de recouvrement non filtré; (b) arbre de recouvrement filtré .

Nous utilisons donc une méthode adaptative, de type *split & merge*, à partir d'un maillage initial qui recouvre la scène perçue. Les approches adaptatives se fondent donc sur deux mécanismes : la séparation (*split*) et la fusion (*merge*), qui s'applique itérativement au maillage en cours de construction

- La séparation, à partir d'un maillage grossier, consiste à découper itérativement les triangles jusqu'à l'obtention d'un maillage suffisamment "proche" des points de mesure.
- La fusion, à partir d'un maillage très fin, consiste à regrouper les triangles en régions homogènes. Différents critères d'homogénéité peuvent être utilisés selon les besoins de modélisation : planéité, courbure limite admissible, ... Les algorithmes de *décimation* de maillage [Ciampalini 97] exploitent ce mécanisme de fusion : ils procèdent par suppression successive de sommets ou d'arêtes.

La séparation part donc d'un maillage initial très simple [Floriani 88] (deux triangles, par exemple), qui est découpé jusqu'à intégration de toutes les mesures, ou jusqu'à satisfaire un critère minimal de "recouvrement" de ces points de mesure. Au contraire, la fusion ou la *décimation* partent d'un maillage initial très segmenté, souvent un maillage régulier généré directement à partir des points de mesure [Hoppe 94, Sequeira 97]. Dans les deux cas, le maillage résultat est irrégulier, s'appuie sur les points de mesure et possède une résolution variable, qui peut être choisie. Par exemple, un maillage peut être décimé jusqu'à ce qu'il ait un nombre donné de sommets.

Nous pouvons exploiter l'une de ces deux méthodes pour générer un maillage à partir d'une image 3D. Pour la fusion, nous avons utilisé un logiciel Domaine Public.

Pour la séparation, nous exploitons la méthode de construction d'un modèle local d'une scène par un maillage, méthode proposée dans sa thèse par J.Bozier. Nous allons rappeler par la suite les différentes étapes de cette méthode. Cet algorithme s'applique à des images 3D, donc en  $2D^{1/2}$ , c'est à dire en 2D dans le repère image ( $u, v$ ) et en 3D dans le repère réel ( $x, y, z$ ). Le maillage local est donc construit en 2D tout en considérant les informations 3D pour les différents calculs de positionnement.

### Création d'un maillage simple.

Utilisant un algorithme basé sur la technique du "split", il faut partir d'un maillage le plus simple possible, facilement différentiable des données sensorielles pour éviter les artefacts de modélisation, et capable d'accueillir l'ensemble de ces données.

Pour cela, nous partons d'un maillage initial formé de deux triangles et de quatre sommets virtuels (voir figure 2.9), ayant une profondeur très supérieure à la profondeur maximale de l'image 3D ; ces sommets, positionnés en 2D aux coins de l'image, seront supprimés ultérieurement. Ce maillage initial représente donc l'horizon. Chaque point de l'image est ensuite rattaché au triangle unique dans le repère image (donc en 2D) auquel il appartient.

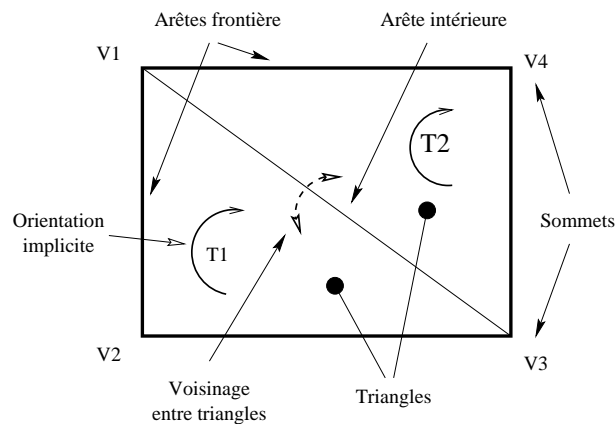


FIG. 2.9 – Description du maillage initial.

### Insertion des contours 3D.

Afin de générer un maillage qui s'appuie sur les discontinuités de la scène, nous pouvons exploiter un ensemble de segments 3D qui ne seront pas modifiés par la suite. Ceci permet d'"accrocher" le maillage local sur des arêtes 3D de la scène extraites par ailleurs, par exemple par l'approche présentée dans la section précédente : extraction des points de discontinuité, de chaînes de contour, approximées par un ensemble de segments 3D. Certains auteurs [Ruggeri 01] ont proposé des méthodes d'affinement de maillage, qui permettent de retrouver de telles arêtes a posteriori, une fois qu'un maillage complet a été construit.

Dans notre cas, les segments sont insérés dans le maillage initial, ce qui provoque un premier découpage. Chaque segment est inséré en tant qu'arête dans le maillage en cours de construction. Ce traitement illustré sur la figure 2.10 doit respecter la topologie du maillage quelque soit les contours extraits de l'image. A chaque modification du maillage, les points de mesure sont redistribués de manière unique, entre les nouveaux triangles.

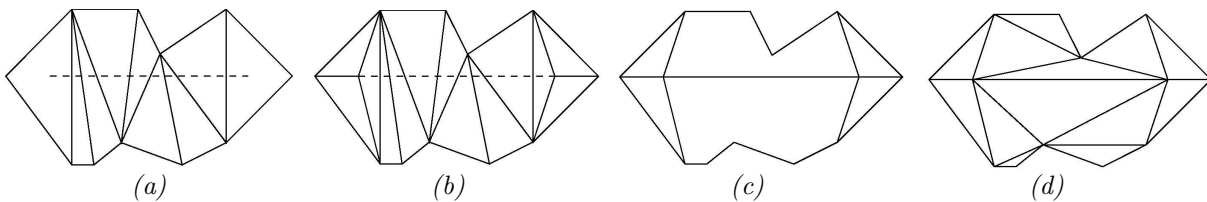


FIG. 2.10 – Processus d'insertion d'un contour : (a) segment à insérer, (b) insertion des extrémités, (c) destruction des triangles, (d) remaillage.

### Segmentation du maillage.

A ce stade, nous disposons d'un maillage initial qui contient l'ensemble des données sensorielles. Nous appliquons à ce maillage un algorithme itératif de séparation afin de le faire converger vers les données sensorielles. Le modèle obtenu est ainsi hiérarchique. Chaque triangle ayant une taille dépendante de la nature de la surface qu'il modélise : plus elle est plane, plus les dimensions du triangle est importante. Le découpage du maillage dépend de plusieurs critères : la forme, la taille des triangles obtenus et l'erreur quadratique de l'estimation d'un ensemble de points dans un plan.

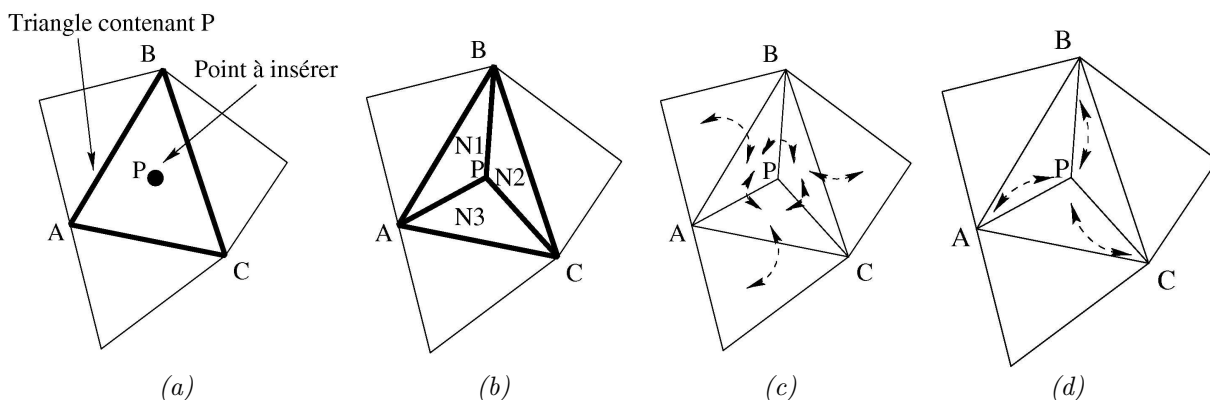


FIG. 2.11 – Insertion d'un sommet dans un triangle : (a) point à insérer, (b) création des nouveaux triangles  $N1, N2, N3$ , mise à jour des relations de voisinage inter-triangles en (c), inter-sommets en (d).

Pour réaliser cette opération, nous étudions les triangles un à un : pour le triangle examiné, le plan support est déterminé par les approximations à partir de tous les points 3D qui lui sont associés ; ensuite,



FIG. 2.12 – Filtrage successif d'un maillage : (a) maillage après le filtrage des facettes à l'horizon, (b) maillage après le filtrage de la densité et filtrage topologique.

nous déterminons s'il doit être découpé ou non par un critère de distance variable : un triangle est découpé si l'erreur d'approximation plane est supérieure à une distance  $D$  dépendante de la taille du triangle. Lors du redécoupage d'un triangle, un nouveau sommet du maillage est créé sur le point de mesure le plus proche du centre de gravité. Trois nouveaux triangles étant formés, nous appliquons un critère de Delaunay simplifié pour sélectionner la triangulation la moins dégénérée. Enfin, les points de mesure du triangle initial seront répartis dans les nouveaux triangles (voir figure 2.11). Cette opération sera répétée tant que des triangles ne vérifieront pas le critère d'homogénéité.

### Filtrage du modèle.

Le modèle obtenu à cette étape doit être ensuite filtré pour supprimer des artefacts de modélisation qui proviennent :

- du maillage initial : ses sommets virtuels étant placés à l'horizon, un certain nombre de facettes artificielles vont s'y appuyer, ne contenant aucune donnée sensorielle.
- des trous de données : ceux-ci peuvent avoir pour origine des surfaces spéculaires (capteur laser), des zones non corrélées car non texturées (stéréo-corrélation), ou tout simplement une absence de données (le ciel dans une image d'extérieur). Leur détection est réalisée grâce à un seuillage de la densité de données sur les facettes, avec un seuil calculé selon la transformation entre la facette et le capteur.
- des caractéristiques topologiques du maillage : le maillage contient des triangles artificiels au niveau des discontinuités de profondeur. Ces facettes étant le plus souvent dégénérées, leur détection est réalisée en étudiant le rapport  $surface/(périmètre * périmètre)$ , nombre sans dimension donc indépendant de la taille du triangle, caractérisant "l'équilatéralité" du triangle (plus le triangle est dégénéré, plus ce nombre est petit).

Un exemple de filtrage est donné sur la figure 2.12. Finalement sur la figure 2.13, nous présentons deux résultats de simplification.



FIG. 2.13 – Exemple de la construction d'un maillage.

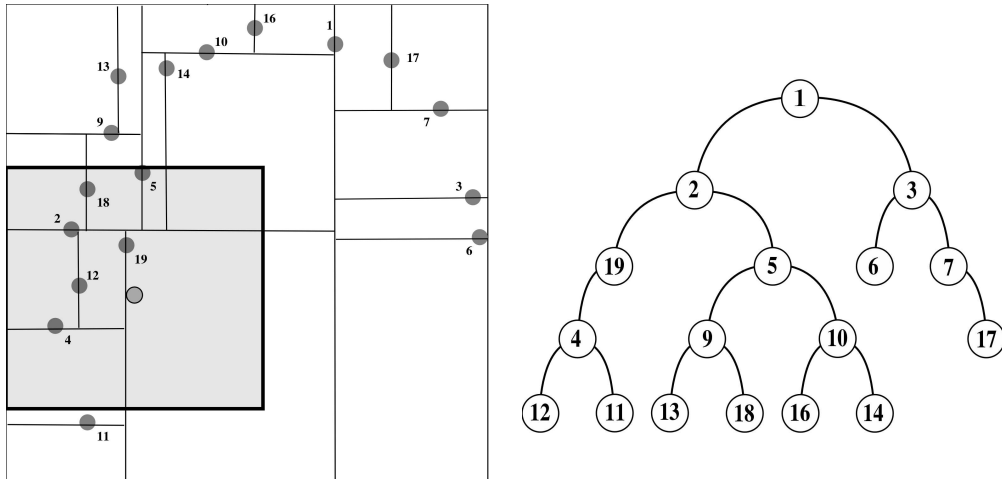


FIG. 2.14 – Organisation dans l'espace d'un k-d tree.

## 2.5 Recherche d'appariements entre points de contrôle.

Nous exploitons l'une des méthodes proposées dans la section précédente, pour extraire d'une image 3D, modèle local noté  $m_t$ , une liste de  $N$  points de contrôle, exprimés dans le repère capteur, noté  $\mathcal{R}_t$ . Cette image ou ce modèle local numéro  $t$ , doit être recalée par ICP, par rapport au modèle global courant  $M_{t-1}$ , fusion de tous les modèles locaux de 0 à  $t - 1$ ; les  $M$  points de contrôle extraits de ces modèles, sont exprimés dans un repère global, noté  $\mathcal{R}_0$ , car il s'agit typiquement du repère capteur pour  $m_0$ .

A l'itération  $k$  de ICP, nous transformons les points de  $m_t$  par  $T_t^k$  pour les exprimer dans  $\mathcal{R}_0$ . Puis, pour chaque point  $p_t^i$ , nous recherchons un appariement dans le modèle global courant : le point  $P_{t-1}^j$  de  $M_{t-1}$  le plus proche de  $p_t^i$  est donc utilisé. Ce processus est lourd et directement proportionnel au nombre de points dans chaque ensemble et le degré de complexité est  $N.M$ .

### 2.5.1 Structure des données.

Cette étape de mise en correspondance des points, est la plus longue en temps de calcul. Nous pouvons diminuer ce temps, en limitant le nombre de points extraits depuis chaque modèle local, en exploitant

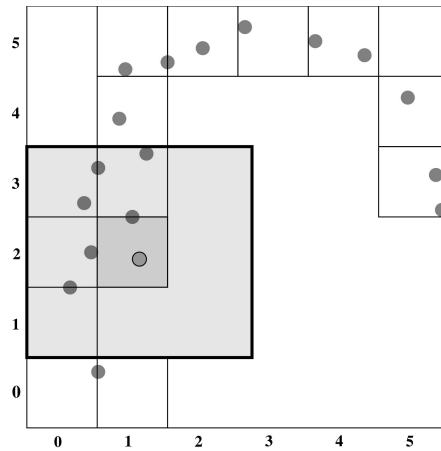


FIG. 2.15 – Organisation de l'espace par clusters.

des critères discriminants d'appariement, ou en améliorant la stratégie de recherche.

Les structures des données proposées pour optimiser les traitements sur des informations tridimensionnelles exploitent une discrétisation de l'espace : il s'agit entre autres, des grilles uniformes ou "clusters", des "octrees", des "k-d trees" et des "BSP-trees" ("Binary Space Partitioning").

Pour cette stratégie, nous avons utilisé des "clusters" ou unités de division de l'espace [Chetverikov 02]. L'idée est d'éviter la comparaison de tous les points d'un ensemble avec les points de l'autre ensemble : les points sont comparés seulement dans le voisinage d'un nombre donné de clusters.

Souvent, la structure la plus utilisée pour faire une comparaison entre deux collections de points, est le *k-d tree*. Créé par Bentley [Bentley 75], le *k-d tree* est un arbre binaire de recherche dans lequel chaque noeud représente une partition de l'espace de dimension  $k$ . La racine est l'espace entier et les fils sont des sous-espaces avec des régions uniques mutuellement exclusives.

L'utilisation d'un *k-d tree* remplace la recherche des points les plus proches par une recherche dans un arbre binaire qui a une complexité de  $N \log(N)$ . A chacun des noeuds de l'arbre un test décide de quel coté de l'hypersurface se trouve le point le plus proche.

Les difficultés principales pour utiliser cette structure sont :

- pour chaque point de  $m_t$  et à chaque itération, on doit recommencer à parcourir l'arbre à partir du noeud racine ;
- la création (pour  $m_0$ ) ou la mise à jour (à chaque fusion d'un modèle  $m_t$  dans  $M_{t-1}$ ) du *k-d tree* est lourde.

La figure 2.14 représente un *k-d tree* à deux dimensions : (1) à gauche, l'organisation dans l'espace d'un *k-d tree* pour une collection de points 2D ; (2) à droite, la structure d'arbre du *k-d tree*. Le point pour lequel on recherche un appariement, et le voisinage de recherche se trouve en gris. La recherche des points proches se limite au noeud 2 et ses fils.

Par contre, les *clusters* divisent l'espace en cubes de dimension constante ; c'est une structure similaire à un tableau en 3 dimensions de cellules volumiques, ou *voxels*, tableau appelé souvent *voxel map*. Vu leur uniformité, les *clusters* ont une structure plus simple que le *k-d tree*. La vitesse de création est meilleure et la recherche de points proches s'accomplit par l'inspection directe à travers des indices 3D. L'inconvénient est le besoin de définir la dimension du *cluster* ; il ne paraît pas simple de gérer une *voxel map* multi-résolution, sans dupliquer toute la structure. Chaque cellule doit être divisée en 8 pour obtenir le niveau de résolution plus fin (comme cela est fait pour la construction d'un *octree*).

La figure 2.15 présente des *clusters* à deux dimensions : afin de comparer qualitativement *k-d trees* et *clusters*, nous avons représenté la même collection de points de la figure 2.14 avec une organisation par *clusters*. Le point et le voisinage de recherche se trouve aussi en gris. La recherche des points proches est faite dans les *clusters* occupés : (0,1),(0,2),(0,3),(1,2) et (1,3).

L'organisation en *clusters* utilisée pour notre méthode d'appariement, se fonde sur les deux caractéristiques suivantes (figure 2.18) :

- L'espace partagé par les deux ensembles de points à comparer, est divisé en *clusters* d'une taille fixe donnée. Ces *clusters* gardent les indices des *clusters* de chaque collection de points. Ils sont indexables doublement :
  - par une matrice tridimensionnelle, qui sert à obtenir facilement le voisinage pour un *cluster* sélectionné,
  - et par une liste de *clusters* occupés, pour éviter de balayer des *clusters* vides de la matrice tridimensionnelle en cherchant l'appariement pour un point donné.

Cette double indexation est nécessaire pour limiter la quantité de mémoire nécessaire pour gérer un tableau de *clusters* de grande taille.

- Chaque ensemble de points est plongé dans une organisation de *clusters* de la même taille que les *clusters* précédents. Ces *clusters* gardent les indices de points correspondants à la subdivision de l'espace.



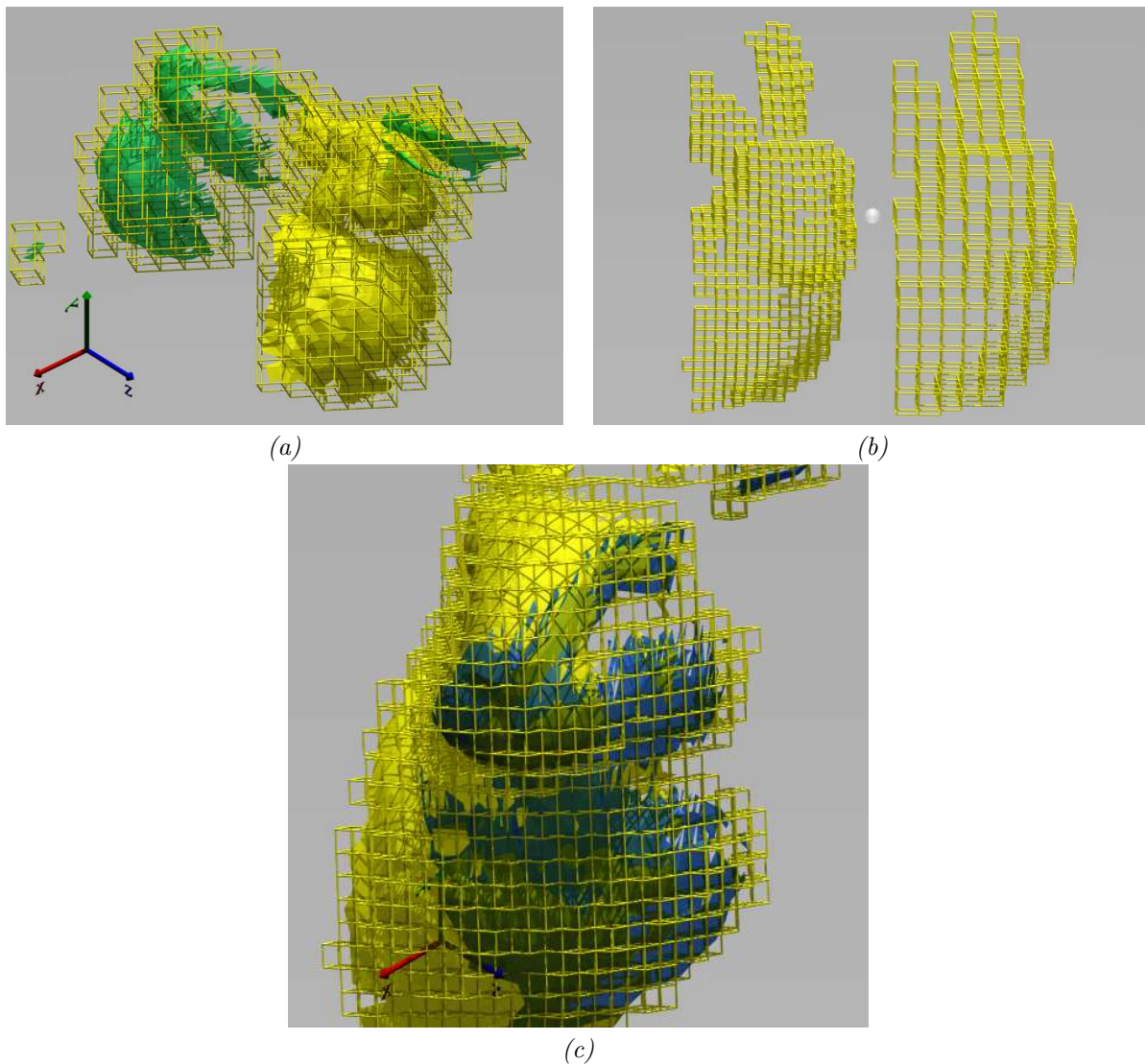


FIG. 2.16 – Recalage de deux vues avec une organisation en *clusters*.

### 2.5.2 Processus.

Le processus d'appariement en exploitant une organisation de l'espace en *clusters*, a pour objectif la comparaison des deux ensembles de points  $p_0$  et  $p_1$  avec une recherche d'un rayon  $r$  autour de chaque *cluster*. Comme tout l'espace est divisé par *clusters* la position approximée de chaque point est connue ; de ce fait une comparaison entre points locaux est suffisante.

L'algorithme est illustré sur la figure 2.16, avec le recalage de deux images de *Bunny*. En (a), l'algorithme commence avec une organisation en *clusters* de grande dimension adaptée à l'espace utilisé par les deux vues. En fonction de l'erreur de recalage, donc de la distance entre les images, la dimension des *clusters* change. En (b), nous montrons pour une image, les deux espaces de *clusters* : espace initial à droite (peu résolu) et espace final à gauche (plus fin). Enfin, en (c) apparaît la position finale des vues avec une organisation de *clusters* plus dense.

A chaque recherche des points de l'ensemble  $p_1$  dans un voisinage d'un point de l'ensemble  $p_0$ , nous commençons avec la révision de la liste de *clusters* globaux occupés  $Bul[]$  qui contient l'index 3D  $i, j, k$

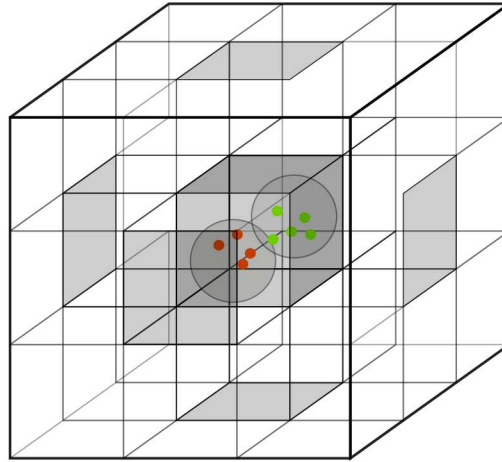


FIG. 2.17 – Cluster  $B[i][j][k]$  avec un voisinage de recherche d'un cluster

de la structure de *clusters* globales dans l'espace  $Bu[][][]$  et les indices des *clusters* de chaque ensemble  $B_0[]$  et  $B_1[]$ .

Les structures de données mises en oeuvre pour le gestion des *clusters*, sont représentées en figures 2.17 et 2.18 :

- en 2.17, un *cluster* global avec les *clusters* en recouvrement des deux ensembles de points.
- en 2.18, un élément  $x$  de la liste  $Bul$  des *clusters* occupés pointe sur un *cluster* global  $B[i][j][k]$  qui contient les indicateurs des *clusters* des ensembles 0 et 1 dans les listes  $B_0[]$  et  $B_1[]$ . A son tour, chaque élément de ces listes de *clusters* par ensemble, pointe sur ses points présents à l'intérieur du *cluster*  $i, j, k$ .

Avec les indices 3D  $i, j, k$  du *cluster* 3D, on calcule les coordonnées de ses *clusters* voisins séparés d'une distance  $r$ . Nous balayons ces *clusters* voisins dans l'espace 3D, et trouvons les points de chaque ensemble avec la référence aux listes  $B_0[]$  et  $B_1[]$  gardés dans chaque *cluster*. La complexité de l'algorithme est  $O(N_0 \cdot (1 + 2r)^3 \frac{N_1}{n^3})$  versus  $O(N_0 \cdot N_1)$  d'une comparaison directe entre points, ou  $(1 + 2r)^3$  est le nombre de *clusters* à visiter à chaque itération et  $\frac{N_1}{n^3}$  est la densité moyenne de points par *cluster*.

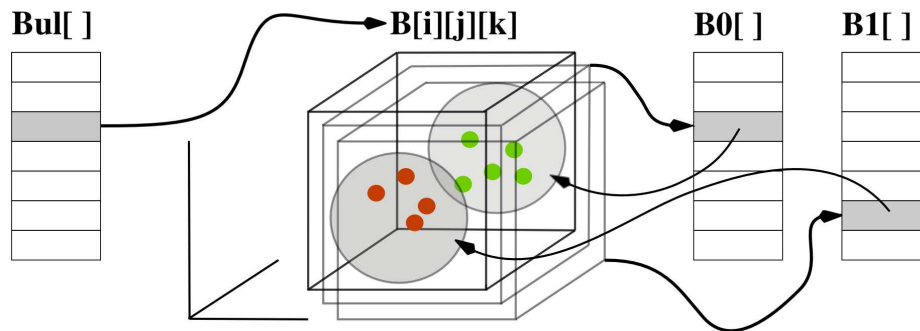


FIG. 2.18 – Structure de données décrivant une organisation de l'espace par *clusters*.

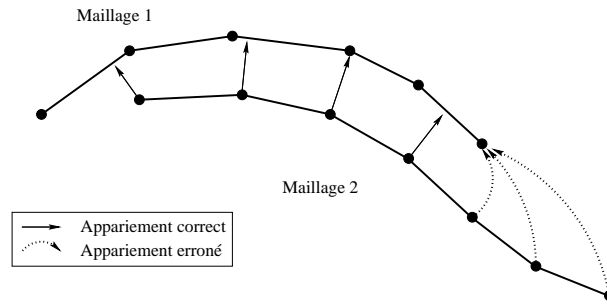


FIG. 2.19 – Filtrage des appariements : points extrêmes d'un des modèles

### 2.5.3 Filtrage.

Avant de passer à l'étape d'estimation de l'erreur de superposition entre les points appariés, un filtrage préliminaire est nécessaire pour éliminer autant que possible les appariements aberrants ; des méthodes robustes d'estimation (du type RANSAC) permettraient aussi de limiter l'impact de ces *outliers*, mais seraient plus coûteuses en temps de calcul.

Deux méthodes heuristiques de filtrage peuvent être appliquées :

- les couples d'appariement dont un des points se trouve sur le bord d'un modèle sont éliminés, car ils sont le plus souvent faux [Turk 94], comme illustré sur la figure 2.19.
- une contrainte d'unicité peut être vérifiée, afin que tout point de  $m_t$  ou  $M_{t-1}$  ne participe qu'à un seul appariement.

Le filtre d'unicité sert à ne conserver que les appariements qui ont la meilleure distance minimale ; cela interdit des juxtapositions de points  $p_i$  et  $p_j$  de  $m_t$  avec un même point  $P$  de  $M_{t-1}$ . Dans le processus d'appariement, qui calcule les distances minimales d'un point d'un ensemble  $m_t$  par rapport aux points d'un deuxième ensemble  $M_{t-1}$ , une pile de profondeur donnée  $K_{pile}$  permet de conserver dans la boucle sur chaque point de  $m_t$ , les  $K_{pile}$  points de  $M_{t-1}$  les plus proches. Au sommet de la pile se trouve le minimum absolu pour chaque point. L'objectif de l'algorithme est d'avoir des paires uniques entre points des deux ensembles, sans réduire le nombre d'appariements. Nous devons effectuer un nombre  $k$  d'itérations pour arriver à une unicité pure ; le nombre maximal d'itérations est la profondeur maximale des piles  $K_{pile}$ .

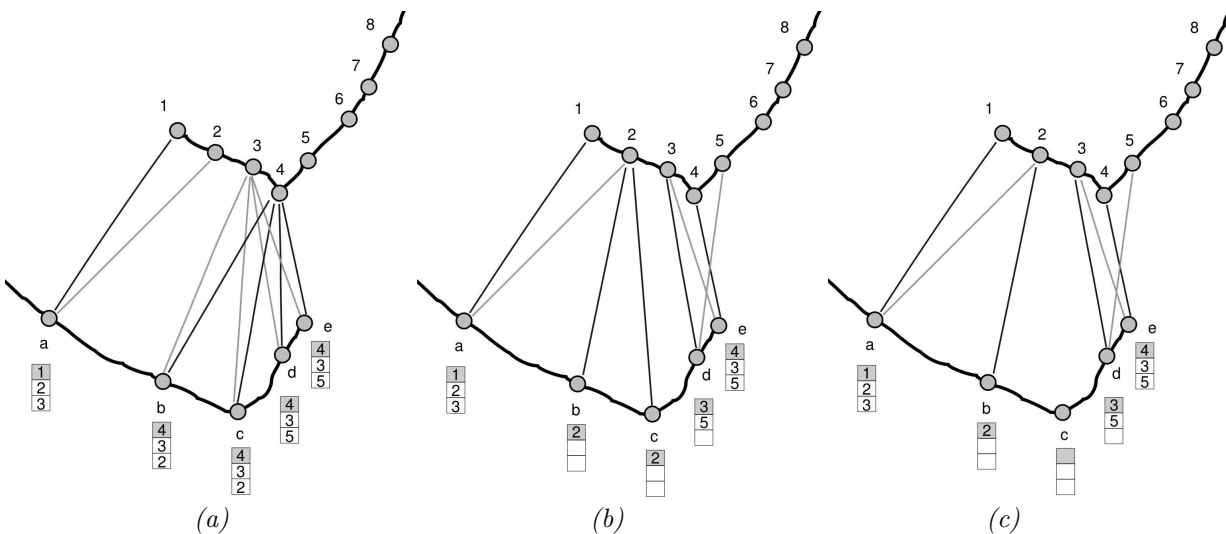


FIG. 2.20 – Filtrage d'unicité avec un empilement doté de 3 minima.

A l'itération 0, nous mémorisons pour chaque point de  $m_t$ , 3 minima, qui ont satisfait le critère des distances au niveau du filtrage, le minimum absolu se trouve alors au sommet de chaque pile. Si quelques points de  $m_t$  ont leurs sommets de pile qui pointent sur le même point de  $M_{t-1}$ , alors la contrainte d'unicité n'est pas respectée; il faut itérer. La première itération trouve les paires en conflit et garde en chaque cas, les appariements qui présentent la distance la plus courte. Les points de  $m_t$  qui ont perdu, dépilent et se trouvent donc appariés avec le minimum suivant de leurs piles. Les itérations suivantes répètent le processus jusqu'à obtenir des piles sans aucun conflit : le minima de chaque point est unique. Eventuellement, certains points peuvent avoir perdu tous leurs minima.

C'est le cas sur l'exemple illustré en figure 2.20. Nous observons les itérations, avec les minima principaux montrés en noir et les secondaires en gris.

- En (a), après la recherche des appariements, les minima trouvés pour les points **b**, **c**, **d** et **e** partagent le même point à distance minimale, le point 4. Une redistribution est nécessaire.
- En (b), après deux itérations; dans l'itération 1, les points **d**, **c** et **b** perdent le sommet de leurs piles, car c'est le point **e** qui est à distance minimale du point 4; à l'itération 2, **b**, **c** et **d** partagent encore le même point à la distance minimale, le point 3. **b**, **c** sortent le minimum suivant de leurs piles pour éviter une incompatibilité avec **d** qui est le plus proche du point 3.
- En (c), à l'itération 3, **b** et **c** partageaient le même minimum avec le point 2; c'est **b** le plus proche. **c** n'a plus de candidat dans sa pile et restera donc sans appariement.

Nous utilisons ensuite une technique de filtrage statistique, inspirée des travaux de Zhang [Zhang 94]), qui consiste à filtrer les appariements en fonction de la distribution des distances entre les points appariés. Les appariements sont filtrés de manière itérative et ne sont conservés à chaque itération  $k$  que les couples  $(p_t^i, P_{t-1}^j)$  ayant une distance inférieure à un seuil  $D_{max}^{k-1}$ . Le seuil initial  $D_{max}^0$  est choisi à une valeur élevée, multiple de la résolution moyenne du capteur. Ce seuil est mis à jour à chaque itération en fonction de la moyenne  $\mu$  des distances entre points appariés et de l'écart type  $\sigma$  de ces distances de la façon suivante :

- si  $\mu < D_{max}^{I-1}$  alors  $D_{max}^I = \mu + 3.\sigma$
- sinon si  $\mu < 3.D_{max}^{I-1}$  alors  $D_{max}^I = \mu + 2.\sigma$
- sinon si  $\mu < 6.D_{max}^{I-1}$  alors  $D_{max}^I = \mu + \sigma$
- sinon  $D_{max}^I = \xi$ ,  $\xi$  étant le premier minimum en fonction de  $\mu$  dans l'histogramme des distances.

Quatre cas sont considérés, correspondant à un ensemble d'appariements

- très homogène ; 99% des appariements sont conservés.
- homogène : seulement 5% des appariements sont rejetés.
- peu homogène : environ un tiers des appariements sont rejetés.
- mauvais : un grand nombre d'appariements sont rejetés.

Le nombre d'itérations doit être un compromis entre le nombre d'appariements conservés et la dispersion des distances entre points appariés

#### 2.5.4 Algorithme complet.

Finalement, une illustration sur l'algorithme d'appariement avec l'utilisation des espaces de *clusters* et des piles pour conserver plusieurs associations potentielles pour un point, est présenté sous forme d'un organigramme dans la figure 2.21.

Un pseudo-code est présenté en Annexe B.

A chaque itération  $k$ , nous avons donc décrit comment obtenir un ensemble cohérent d'appariements entre points de  $m_t$  et points de  $M_{t-1}$ . Nous allons pouvoir évaluer l'erreur de superposition entre ces points appariés.

## 2.6 Estimation de la transformation 3D.

A cette étape, nous possédons un ensemble de  $N$  appariements  $(\vec{p}_i, \vec{p}_i')$ . Nous utilisons alors une procédure d'estimation de la transformation 3D :  $\mathbf{R}, \vec{t}$ , minimisant une erreur définie par la moyenne des distances euclidiennes entre les points appariés.

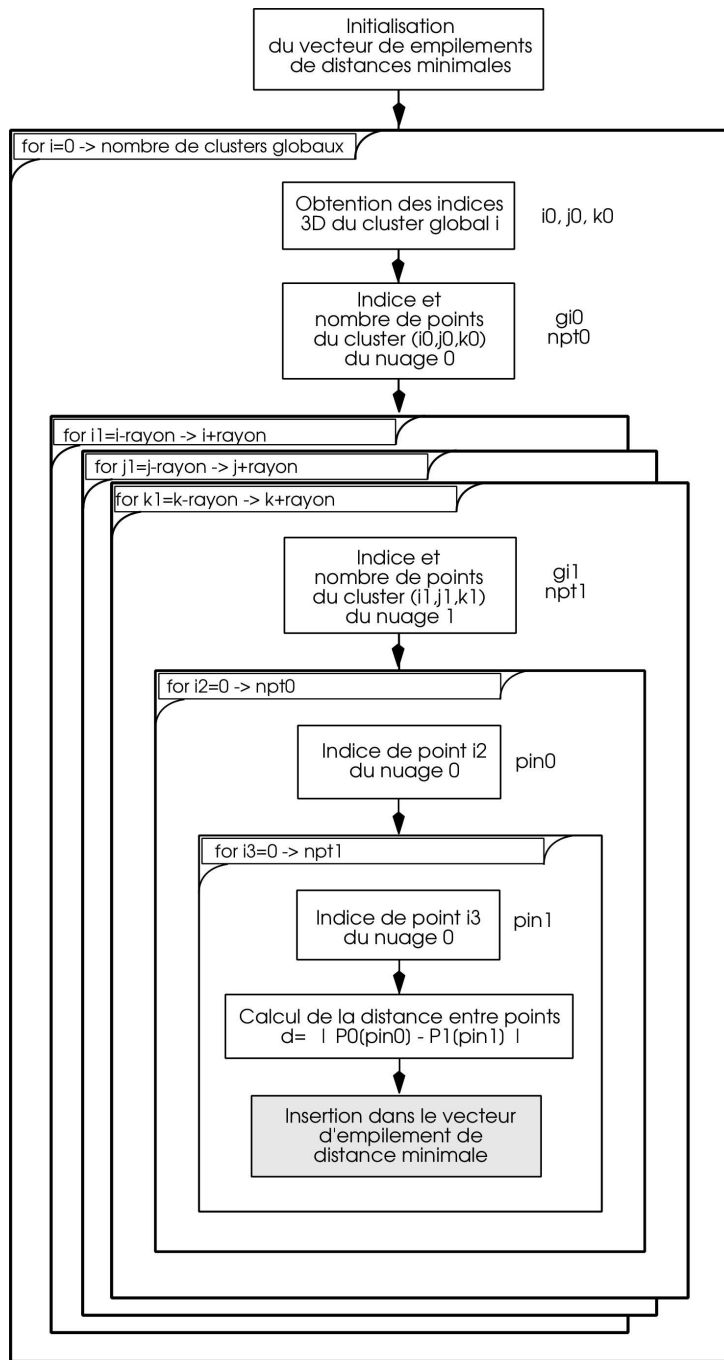


FIG. 2.21 – L’algorithme d’appariement basé sur des espaces de clusters et des piles de distances minimales.

La quantité à minimiser est :

$$f = \sum_{i=0}^{N-1} [\vec{p}'_i - (\mathbf{R}\vec{p}_i + \vec{t})]^2 \quad (2.1)$$

où les  $N$  points  $\vec{p}_i$  de l’ensemble 1 exprimés dans un repère  $s_1$ , sont appariés avec les points  $\vec{p}'_i$  de l’ensemble 2, exprimés dans un repère  $s_2$ . La figure 2.22 présente deux séries de points appariés (zones

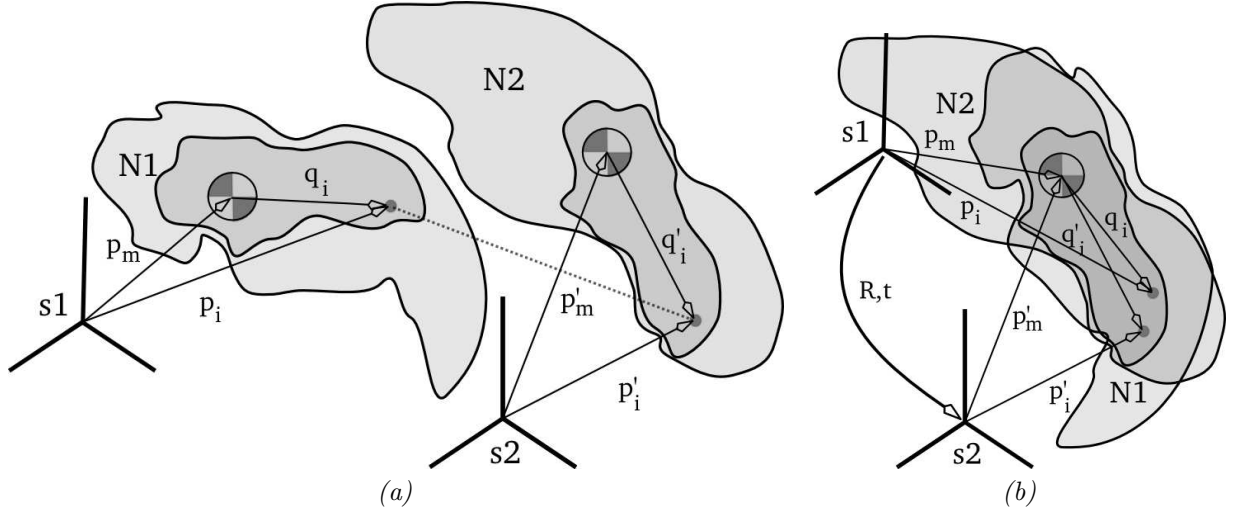


FIG. 2.22 – Deux ensembles de points appariés, avant et après recalage.

sombres à l'intérieur de chaque nuage) dans leurs systèmes de référence respectifs  $s_1$  et  $s_2$ , et la paire des points  $p_i$  et  $p'_i$ . En (a), les deux ensembles avant estimation de la transformation  $\mathbf{R}, \vec{t}$  qui est l'erreur courante de superposition ; en (b), les deux ensembles, après estimation de la transformation et application de celle-ci sur  $N1$  pour superposer au mieux les deux nuages.

Ce problème de minimisation peut être résolu par un algorithme itératif du type descente du gradient, mais plusieurs auteurs ont proposé des méthodes directes présentées dans cette section. Les méthodes connues sont le calcul par SVD [Arun 87], par les quaternions [Horn 87] et par les quaternions doubles [Zhang 94].

### 2.6.1 Estimation par SVD.

Après avoir effectué les appariements entre deux groupes de points, comment estimer la meilleure transformation (rotation et translation) entre les deux en minimisant la fonction 2.1 ? La méthode d'estimation par SVD ("singular value decomposition") a été décrite par [Arun 87] et [Haralick 89]. Le processus mathématique fondé sur SVD est illustré ci-dessous.

La stratégie à utiliser est de séparer le problème en deux, une partie qui traite la translation et l'autre la rotation.

La translation se sépare du problème en calculant le centre de masse des points appariés de chaque nuage, qui est :

$$\vec{p}_m = \frac{1}{N} \sum_{i=0}^{N-1} \vec{p}_i \quad \vec{p}'_m = \frac{1}{N} \sum_{i=0}^{N-1} \vec{p}'_i \quad (2.2)$$

Nous décrivons les points de chaque nuage à partir de leur centre de masse respectif :

$$\vec{q}_i = \vec{p}_i - \vec{p}_m \quad \vec{q}'_i = \vec{p}'_i - \vec{p}'_m \quad (2.3)$$

Après application de la transformation  $\mathbf{R}, \vec{t}$ , qui minimise 2.1, sur le nuage 1, on peut dire que les deux centres de masse seront égaux. Donc, le centre de masse du nuage 1 vu depuis le système  $s_2$  sera égal au centre de masse du nuage 2 (voir la figure 2.22 b) :

$$\frac{1}{N} \sum_{i=0}^{N-1} (\mathbf{R}\vec{p}_i + \vec{t}) = \mathbf{R}\vec{p}_m + \vec{t} = \vec{p}'_m \quad (2.4)$$

Exprimons la fonction à minimiser  $f$  de l'équation 2.1, en intégrant les centres de masse (équation 2.3) et en prenant en compte la relation qui relie ces centres 2.4 :

$$\begin{aligned}
 f &= \sum_{i=0}^{N-1} [\vec{q}_i' + \vec{p}_m' - (\mathbf{R}(\vec{q}_i + \vec{p}_m) - \vec{t})]^2 \\
 &= \sum_{i=0}^{N-1} [\vec{q}_i' + \vec{p}_m' - \mathbf{R}\vec{q}_i - \mathbf{R}\vec{p}_m - \vec{t}]^2 \\
 &= \sum_{i=0}^{N-1} [\vec{q}_i' - \mathbf{R}\vec{q}_i]^2
 \end{aligned} \tag{2.5}$$

De cette manière, notre fonction à minimiser amène à l'application de la rotation  $\mathbf{R}$ , uniquement.

Développons l'équation pour une variation des coordonnées à l'intérieur :

$$\begin{aligned}
 (\vec{q}_i' - \mathbf{R}\vec{q}_i)^T (\vec{q}_i' - \mathbf{R}\vec{q}_i) &= \vec{q}_i'^T \vec{q}_i' - \vec{q}_i'^T (\mathbf{R}\vec{q}_i) - (\mathbf{R}\vec{q}_i)^T \vec{q}_i' + (\mathbf{R}\vec{q}_i)^T (\mathbf{R}\vec{q}_i) \\
 &= \vec{q}_i'^T \vec{q}_i' - \underbrace{\vec{q}_i'^T (\mathbf{R}\vec{q}_i) + (\mathbf{R}\vec{q}_i)^T \vec{q}_i'}_{2\vec{q}_i'^T \mathbf{R}\vec{q}_i} + \vec{q}_i'^T \mathbf{R}^T \mathbf{R}\vec{q}_i \\
 &= \vec{q}_i'^T \vec{q}_i' - 2\vec{q}_i'^T \mathbf{R}\vec{q}_i + \vec{q}_i'^T \vec{q}_i
 \end{aligned} \tag{2.6}$$

Etant donné que :

$$\mathbf{R}^T \mathbf{R} = \mathbf{I} \quad \mathbf{R}^T = \mathbf{R}^{-1} \quad \vec{a}^T \vec{b} = \vec{b}^T \vec{a} \tag{2.7}$$

nous obtenons le terme dont l'extrémum est recherché devient :

$$\begin{aligned}
 f &= \sum_{i=0}^{N-1} \vec{q}_i'^T \mathbf{R}\vec{q}_i \\
 &= \text{Trace} \left( \sum_{i=0}^{N-1} \mathbf{R}\vec{q}_i \vec{q}_i'^T \right) = \text{Trace}(\mathbf{R}\mathbf{H})
 \end{aligned} \tag{2.8}$$

où  $\mathbf{H}$  prend la forme suivante :

$$\mathbf{H} \equiv \sum_{i=0}^{N-1} \vec{q}_i \vec{q}_i'^T$$

Exprimons  $\mathbf{H}$  à l'aide de la décomposition SVD ("Singular Value Descomposition") :

$$\mathbf{H} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^T \tag{2.9}$$

Dans l'égalité précédente les matrices  $\mathbf{U}$  et  $\mathbf{V}$  sont orthonormales et  $\mathbf{\Lambda}$  est une matrice diagonale positive contenant les valeurs propres (ou singulières). Si nous obtenons

$$\mathbf{X} = \mathbf{V}\mathbf{U}^T \quad \Rightarrow \quad \mathbf{X}\mathbf{H} = \mathbf{V}\mathbf{U}^T \mathbf{U}\mathbf{\Lambda}\mathbf{V}^T = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T \tag{2.10}$$

$\mathbf{X}\mathbf{H}$  est symétrique positif. Or, nous savons que :

$$\text{Trace}(\mathbf{X}\mathbf{H}) \geq \text{Trace}(\mathbf{B}\mathbf{X}\mathbf{H}) \tag{2.11}$$

pour toute matrice  $\mathbf{B}$  orthonormale 3x3. Dans ces conditions  $\mathbf{X}$  maximise la fonction 2.8.

Finalement nous pouvons exprimer les résultats :

$$\mathbf{R} = \mathbf{V}\mathbf{U}^T \tag{2.12}$$

$$\vec{t} = \vec{p}_m' - \mathbf{R}\vec{p}_m \tag{2.13}$$

Cette technique d'estimation d'une transformation 3D est certainement la plus utilisée dans la littérature. C'est cette méthode qui est intégrée dans notre logiciel. Nous avons toutefois analysé les approches fondées sur (1) la représentation des rotations par des quaternions, et (2) la représentation des transformations par des quaternions doubles.

Ces analyses sont décrites dans les deux sections ci-dessous.

## 2.6.2 Estimation par quaternions doubles.

Cette méthode a été introduite par [Walker 91] et popularisée par Z.Zhang dans [Zhang 94], pour résoudre la minimisation de la transformation entre les points appariés. Elle est basée sur la représentation des transformations par les quaternions doubles à la différence de la méthode SVD décrite ci-dessus, pour laquelle elles sont représentées par des matrices.

Le processus mathématique est montré ci-dessous. Les fondements sur les quaternions et les quaternions doubles sont donnés dans les annexes A.1 et A.2.

Comme pour la méthode SVD, l'équation à minimiser est 2.1 :

$$f = \sum_{i=0}^{N-1} [\vec{p}'_i - (\mathbf{R}\vec{p}_i + \vec{t})]^2$$

Nous utilisons la formule A.43 pour exprimer la transformation à estimer avec le quaternion double  $\hat{\mathbf{q}} = \mathbf{r} + \epsilon \mathbf{s}$ , décrit en A.2 :

$$\mathbf{R}\vec{x} + \vec{t} \Rightarrow \mathbf{X} \mathbf{R}_{row}(\mathbf{r})^T \mathbf{L}_{row}(\mathbf{r}) + \mathbf{s} \mathbf{R}_{row}(\mathbf{r})^T$$

ou bien

$$\mathbf{R}\vec{x} \Rightarrow \mathbf{X} \mathbf{R}_{row}(\mathbf{r})^T \mathbf{L}_{row}(\mathbf{r}) \quad (2.14)$$

$$\vec{t} \Rightarrow \mathbf{s} \mathbf{R}_{row}(\mathbf{r})^T \quad (2.15)$$

Nous développons l'expression, mais sans extraire les centres de masse, comme cela est fait dans la méthode SVD.

$$\begin{aligned} f_i &= (\vec{p}'_i - (\mathbf{R}\vec{p}_i + \vec{t}))^T (\vec{p}'_i - (\mathbf{R}\vec{p}_i + \vec{t})) \\ &= \vec{p}'_i{}^T \vec{p}'_i - \vec{p}'_i{}^T (\mathbf{R}\vec{p}_i + \vec{t}) - (\mathbf{R}\vec{p}_i + \vec{t})^T \vec{p}'_i + (\mathbf{R}\vec{p}_i + \vec{t})^T (\mathbf{R}\vec{p}_i + \vec{t}) \\ &= \vec{p}'_i{}^T \vec{p}'_i - \vec{p}'_i{}^T \mathbf{R}\vec{p}_i - \vec{p}'_i{}^T \vec{t} - \vec{p}_i{}^T \mathbf{R}^T \vec{p}'_i - \vec{t}^T \vec{p}'_i + (\mathbf{R}\vec{p}_i + \vec{t})^T (\mathbf{R}\vec{p}_i + \vec{t}) \\ &= \vec{p}'_i{}^T \vec{p}'_i - 2\vec{p}'_i{}^T \mathbf{R}\vec{p}_i - 2\vec{p}'_i{}^T \vec{t} + \vec{p}_i{}^T \mathbf{R}^T \mathbf{R}\vec{p}_i + 2\vec{t}^T \mathbf{R}\vec{p}_i + \vec{t}^T \vec{t} \\ &= \vec{p}'_i{}^T \vec{p}'_i + \vec{p}_i{}^T \vec{p}_i + \vec{t}^T \vec{t} - 2\vec{t}^T \vec{p}'_i + 2\vec{t}^T \mathbf{R}\vec{p}_i - 2\vec{p}'_i{}^T \mathbf{R}\vec{p}_i \end{aligned}$$

Ensuite nous utilisons la représentation par un quaternion double avec les équations 2.14 et 2.15 et arrangeons les notations :

$$\begin{aligned} f_i &= (P'_i P'_i{}^T + P_i P_i{}^T) + \mathbf{s} \mathbf{R}_{row}(\mathbf{r})^T (\mathbf{s} \mathbf{R}_{row}(\mathbf{r})^T)^T \\ &\quad - 2P'_i (\mathbf{s} \mathbf{R}_{row}(\mathbf{r})^T)^T + 2P_i \mathbf{R}_{row}(\mathbf{r})^T \mathbf{L}_{row}(\mathbf{r}) (\mathbf{s} \mathbf{R}_{row}(\mathbf{r})^T)^T - 2P_i \mathbf{R}_{row}(\mathbf{r})^T \mathbf{L}_{row}(\mathbf{r}) P'_i{}^T \\ &= (P'_i P'_i{}^T + P_i P_i{}^T) + \mathbf{s} \mathbf{R}_{row}(\mathbf{r})^T \mathbf{R}_{row}(\mathbf{r}) \mathbf{s}^T \\ &\quad - 2P'_i \mathbf{R}_{row}(\mathbf{r}) \mathbf{s}^T + 2P_i \mathbf{L}_{row}(\mathbf{r}) \mathbf{R}_{row}(\mathbf{r})^T \mathbf{R}_{row}(\mathbf{r}) \mathbf{s}^T - 2P_i \mathbf{R}_{row}(\mathbf{r})^T \mathbf{L}_{row}(\mathbf{r}) P'_i{}^T \end{aligned}$$

qui donne finalement

$$f_i = (P'_i P'_i{}^T + P_i P_i{}^T) + \mathbf{s} \mathbf{s}^T - 2 \mathbf{r} (\mathbf{L}_{row}(P'_i) - \mathbf{R}_{row}(P_i)) \mathbf{s}^T - 2 \mathbf{r} \mathbf{L}_{row}(P_i)^T \mathbf{R}_{row}(P'_i) \mathbf{r}^T$$

ou bien

$$f = K + \mathbf{s} \mathbf{s}^T + \mathbf{r} C_2 \mathbf{s}^T + \mathbf{r} C_1 \mathbf{r}^T \quad (2.16)$$

Les quaternions doubles doivent satisfaire deux contraintes rappelées en annexe, dans l'équation A.27. Ces contraintes doivent être prises en compte dans la fonction à minimiser.

$$f = K + \mathbf{s} \mathbf{s}^T + \mathbf{r} C_2 \mathbf{s}^T + \mathbf{r} C_1 \mathbf{r}^T + \lambda_1 (\mathbf{r} \mathbf{r}^T - 1) + \lambda_2 (\mathbf{r} \mathbf{s}^T) \quad (2.17)$$



où  $\lambda_1$  et  $\lambda_2$  sont des multiplicateurs de Lagrange, et avec

$$\begin{aligned}
 C_1 &= -2 \sum_{i=0}^{N-1} \mathbf{L}_{row}(P_i)^T \mathbf{R}_{row}(P_i) \\
 C_2 &= -2 \sum_{i=0}^{N-1} [\mathbf{L}_{row}(P'_i) - \mathbf{R}_{row}(P_i)] \\
 K &= \sum_{i=0}^{N-1} [(P'_i P_i'^T + P_i P_i^T)] \quad (2.18)
 \end{aligned}$$

Dans ces formulations,  $\mathbf{L}_{row}(P_i)$ ,  $\mathbf{L}_{row}(P'_i)$ ,  $\mathbf{R}_{row}(P_i)$ ,  $\mathbf{R}_{row}(P'_i)$  sont des matrices obliques symétriques, grâce au zéro de la partie scalaire des quaternions  $P_i$  et  $P'_i$ . Les matrices  $C_1$  et  $C_2$  sont :

$$\begin{aligned}
 C_{1i} &= \mathbf{L}_{row}(P_i)^T \mathbf{R}_{row}(P_i') = \begin{bmatrix} -\mathbf{K}(\check{\mathbf{p}}_i) & \check{\mathbf{p}}_i^T \\ -\check{\mathbf{p}}_i & 0 \end{bmatrix} \begin{bmatrix} -\mathbf{K}(\check{\mathbf{p}}_i') & -\check{\mathbf{p}}_i'^T \\ \check{\mathbf{p}}_i' & 0 \end{bmatrix} \\
 &= \begin{bmatrix} \mathbf{K}(\check{\mathbf{p}}_i)\mathbf{K}(\check{\mathbf{p}}_i') + \check{\mathbf{p}}_i^T \check{\mathbf{p}}_i' & \mathbf{K}(\check{\mathbf{p}}_i)\check{\mathbf{p}}_i'^T \\ \check{\mathbf{p}}_i \mathbf{K}(\check{\mathbf{p}}_i') & \check{\mathbf{p}}_i \check{\mathbf{p}}_i'^T \end{bmatrix} = \begin{bmatrix} \check{\mathbf{p}}_i'^T \check{\mathbf{p}}_i + \check{\mathbf{p}}_i^T \check{\mathbf{p}}_i' - \check{\mathbf{p}}_i' \check{\mathbf{p}}_i^T \mathbf{I} & (\check{\mathbf{p}}_i' \wedge \check{\mathbf{p}}_i)^T \\ \check{\mathbf{p}}_i' \wedge \check{\mathbf{p}}_i & \check{\mathbf{p}}_i' \check{\mathbf{p}}_i'^T \end{bmatrix} \quad (2.19)
 \end{aligned}$$

$$C_{2i} = \mathbf{L}_{row}(P_i') - \mathbf{R}_{row}(P_i) = \begin{bmatrix} \mathbf{K}(\check{\mathbf{p}}_i') + \mathbf{K}(\check{\mathbf{p}}_i) & -\check{\mathbf{p}}_i'^T + \check{\mathbf{p}}_i^T \\ \check{\mathbf{p}}_i' - \check{\mathbf{p}}_i & 0 \end{bmatrix} \quad (2.20)$$

La matrice  $\mathbf{K}$  est définie par l'expression A.6 de l'annexe. On voit que  $C_1$  est une matrice symétrique, par contre  $C_2$  est oblique symétrique.

$$C_1^T = C_1 \quad C_2^T = -C_2 \quad (2.21)$$

Pour trouver les minima de la fonction  $f$ , les dérivées partielles sur les variables  $\mathbf{r}$  et  $\mathbf{s}$  doivent être égales à zéro :

$$\frac{\partial f}{\partial \mathbf{r}} = C_2 \mathbf{s}^T + (C_1 + C_1^T) \mathbf{r}^T + 2\lambda_1 \mathbf{r}^T + \lambda_2 \mathbf{s}^T = 0 \quad (2.22)$$

$$\frac{\partial f}{\partial \mathbf{s}} = 2 \mathbf{s}^T + C_2^T \mathbf{r}^T + \lambda_2 \mathbf{r}^T = 0 \quad (2.23)$$

Multiplions par  $\mathbf{r}$  l'équation précédente :

$$0 = 2 \mathbf{r} \mathbf{s}^T + \mathbf{r} C_2^T \mathbf{r}^T + \lambda_2 \mathbf{r} \mathbf{r}^T = \mathbf{r} C_2^T \mathbf{r}^T + \lambda_2 \Rightarrow \lambda_2 = 0 \quad (2.24)$$

ce qui nous permet de sortir  $\mathbf{s}$  de l'équation 2.23 :

$$2 \mathbf{s}^T + C_2^T \mathbf{r}^T = 0 \Rightarrow \mathbf{s}^T = -\frac{1}{2} C_2^T \mathbf{r}^T \quad (2.25)$$

De l'équation 2.22 nous obtenons

$$-\frac{1}{2} C_2 C_2^T \mathbf{r}^T + (C_1 + C_1^T) \mathbf{r}^T + 2\lambda_1 \mathbf{r}^T = 0 \Rightarrow \lambda_1 \mathbf{r}^T = \frac{1}{2} (\frac{1}{2} C_2 C_2^T - C_1 - C_1^T) \mathbf{r}^T$$

$$\mathbf{M} = -\frac{1}{2} (C_1 + C_1^T - \frac{1}{2} C_2 C_2^T) \quad (2.26)$$

$$\lambda_1 \mathbf{r}^T = \mathbf{M} \mathbf{r}^T \quad (2.27)$$

qui montre que le quaternion  $\mathbf{r}$  est un vecteur propre de la matrice  $\mathbf{M}$ .

Si ces résultats sont remplacés dans la fonction à minimiser, nous arrivons à

$$\begin{aligned}
 f &= K + \frac{1}{4} \mathbf{r} \mathbf{C}_2 \mathbf{C}_2^T \mathbf{r}^T - \frac{1}{2} \mathbf{r} \mathbf{C}_2 \mathbf{C}_2^T \mathbf{r}^T + \mathbf{r} \mathbf{C}_1 \mathbf{r}^T + \mathbf{r} \lambda_1 \mathbf{r}^T - \lambda_1 \\
 &= K - \frac{1}{4} \mathbf{r} \mathbf{C}_2 \mathbf{C}_2^T \mathbf{r}^T + \mathbf{r} \mathbf{C}_1 \mathbf{r}^T + \mathbf{r} \left( \frac{1}{4} \mathbf{C}_2 \mathbf{C}_2^T - \frac{1}{2} \mathbf{C}_1 - \frac{1}{2} \mathbf{C}_1^T \right) \mathbf{r}^T - \lambda_1 \\
 f &= K - \lambda_1
 \end{aligned} \tag{2.28}$$

La fonction  $f$  est donc minimale pour la valeur propre  $\lambda_1$  maximale.

La transformation 3D,  $\mathbf{R}$  et  $\vec{t}$ , peut être calculée à partir du quaternion double  $\hat{\mathbf{q}} = \mathbf{r} + \epsilon \mathbf{s}$ , comme cela est démontré dans l'annexe dans les équations A.17 et A.42.

$$\begin{aligned}
 \mathbf{R} &= (w^2 - \check{\mathbf{q}} \check{\mathbf{q}}^T) \mathbf{I} + 2w \mathbf{K}(\check{\mathbf{q}}) + 2\check{\mathbf{q}}^T \check{\mathbf{q}} \quad ; \quad \mathbf{r} = (w, \check{\mathbf{q}}) \\
 \mathbf{t} &= \mathbf{s} \mathbf{R}_{row}(\mathbf{r})^T
 \end{aligned}$$

### 2.6.3 Estimation par quaternions.

Proposé par [Horn 87] et utilisé par [Besl 92], ce calcul de la transformation 3D est un croisement entre les deux méthodes précédentes, car il est basé sur les quaternions, comme dans la méthode précédente décrite en 2.6.2, mais en dissociant la translation, comme dans la méthode SVD décrite en 2.6.1. Donc, nous obtenons une description du problème mathématique, proche de celle exploitée par SVD, sauf que la rotation restante est représentée par un quaternion.

De la même façon comme dans la section 2.6.1, la fonction à minimiser 2.1 est donc simplifiée en exploitant les centres de masse des nuages 2.2 pour arriver à l'équation 2.5, où la translation a été séparée. Après avoir développer, l'expression 2.6 donne :

$$f = \sum_{i=0}^{N-1} [ \vec{q}_i'^T \vec{q}_i' + \vec{q}_i^T \vec{q}_i - 2\vec{q}_i'^T \mathbf{R} \vec{q}_i ] \tag{2.29}$$

Nous représentons la matrice de rotation avec le quaternion  $\mathbf{r}$  en utilisant 2.14 et 2.15 ; arrangeons les notations, comme dans la section précédente

$$\begin{aligned}
 f_i &= (Q_i' Q_i'^T + Q_i Q_i^T) - 2Q_i \mathbf{R}_{row}(\mathbf{r})^T \mathbf{L}_{row}(\mathbf{r}) Q_i'^T \\
 &= (Q_i' Q_i'^T + Q_i Q_i^T) - 2 \mathbf{r} \mathbf{L}_{row}(Q_i)^T \mathbf{R}_{row}(Q_i') \mathbf{r}^T
 \end{aligned} \tag{2.30}$$

Le résultat est 2.16 mais sans les expressions du quaternion  $\mathbf{s}$  :

$$f = K + \mathbf{r} \mathbf{C}_1 \mathbf{r}^T \tag{2.31}$$

$$\mathbf{C}_1 = -2 \sum_{i=0}^{N-1} \mathbf{L}_{row}(P_i)^T \mathbf{R}_{row}(P_i') \tag{2.32}$$

La restriction de quaternion unitaire, nécessaire pour un quaternion qui représente une rotation, doit être incluse dans la fonction à minimiser :

$$f = K + \mathbf{r} \mathbf{C}_1 \mathbf{r}^T + \lambda_1 (\mathbf{r} \mathbf{r}^T - 1) \tag{2.33}$$

Calculons la dérivée pour trouver les extrêmes :

$$\frac{\partial f}{\partial \mathbf{r}} = (\mathbf{C}_1 + \mathbf{C}_1^T) \mathbf{r}^T + 2\lambda_1 \mathbf{r}^T = 0 \tag{2.34}$$

qui donne avec 2.21

$$\lambda_1 \mathbf{r}^T = \mathbf{M} \mathbf{r}^T \tag{2.35}$$

Le quaternion  $\mathbf{r}$  est la valeur propre de la matrice  $\mathbf{M} = -\mathbf{C}_1$ .

$$\mathbf{M} = 2 \sum_{i=0}^{N-1} \begin{bmatrix} \vec{q}_i' \vec{q}_i'^T + \vec{q}_i \vec{q}_i'^T - \vec{q}_i'^T \vec{q}_i \mathbf{I} & \vec{q}_i' \wedge \vec{q}_i \\ (\vec{q}_i' \wedge \vec{q}_i)^T & \vec{q}_i'^T \vec{q}_i \end{bmatrix} \quad (2.36)$$

Nous pouvons écrire cette expression d'une autre façon, si l'expression  $\mathbf{E} = \sum_{i=0}^{N-1} \vec{q}_i' \vec{q}_i'^T$  vient de la matrice de covariance croisée. Finalement

$$\mathbf{M} = 2 \begin{bmatrix} \mathbf{E} + \mathbf{E}^T - \text{Trace}(\mathbf{E})\mathbf{I} & \Delta^T \\ \Delta & \text{Trace}(\mathbf{E}) \end{bmatrix} \quad (2.37)$$

avec  $\Delta$  qui regroupe les éléments non-diagonaux de  $\mathbf{D} = \mathbf{E} - \mathbf{E}^T$  :  $\Delta = [D_{2,3} \ D_{1,3} \ D_{1,2}]$ .

Finalement, pour obtenir la transformation nous convertissons le quaternion  $\mathbf{r}$  en une matrice de rotation avec la formule A.18 et la translation avec 2.13, comme dans la méthode SVD.

## 2.7 Stratégies d'itérations avec modification du comportement de l'algorithme.

Une fois trouvée la transformation qui exprime l'erreur d'alignement ou de superposition entre deux nuages de points, elle est appliquée au nuage de points  $N_1$  qui se rapproche ainsi du nuage statique  $N_2$ .

La méthode s'exécute à nouveau en recherchant des appariements entre points du nuage transformé et du nuage statique, et en estimant une nouvelle erreur d'alignement. La méthode est itérée, jusqu'à ce que l'erreur d'alignement, qui en fait est l'évaluation de la fonction de minimisation, soit inférieure à un seuil choisi par l'utilisateur.

Dans notre version de la méthode ICP, nous avons combiné plusieurs stratégies pour effectuer les itérations selon la phase du recalage :

- la phase d'approche entre les deux nuages à recaler, qui peuvent être assez éloignés l'un de l'autre si la transformée initiale  $T_0$  est peu précise.
- ou la phase d'affinement, qui est un traitement précis et détaillé.

La figure 2.23 montre un exemple du changement de stratégies pendant un processus d'ICP. La commutation entre les stratégies, peut être contrôlée en analysant

- l'erreur d'alignement qui doit diminuer et devenir inférieure au seuil d'arrêt,
- le taux de partage des espaces de *clusters* entre les nuages de points, taux qui augmente quand les nuages s'alignent,
- ou la dérivée de l'erreur d'alignement qui approche de zéro à la fin du processus.

Le taux de partage de clusters est une mesure de recouvrement des ensembles de points. Dans cet exemple, la stratégie 0 fonctionne jusqu'à l'itération 75. Ensuite la stratégie 1 tourne entre l'itération 76 et 103 et la stratégie 2 s'arrête à la itération 120. Nous notons une forte fluctuation du taux de partage de clusters entre la stratégie 1 et 2, qui est produite par la *reclustérisation* de l'espace, afin de traiter plus efficacement les nuages plus proches. *Reclustériser* consiste à plonger les nuages dans un nouvel espace de *clusters*, à la résolution plus fine que le précédent.

Les trois stratégies sont les suivantes :

- **Stratégie 0 :**

la méthode doit faire un recalage grossier des deux nuages. La priorité est de trouver assez d'appariements pour traiter de l'estimation de la transformation inter-vues. Une stratégie complexe et judicieuse de mise en correspondance des points n'est pas utile à ce niveau, mais le filtre statistique de Z.Zhang est appliqué seulement à ce niveau de calcul pour conserver un ensemble statistiquement cohérent d'appariements.

Le voisinage de recherche dans les espaces de *cluster* doit être compris entre 3 et 4 et ceci en fonction de la distance séparant les nuages. Le critère d'unicité n'est pas mis en oeuvre.

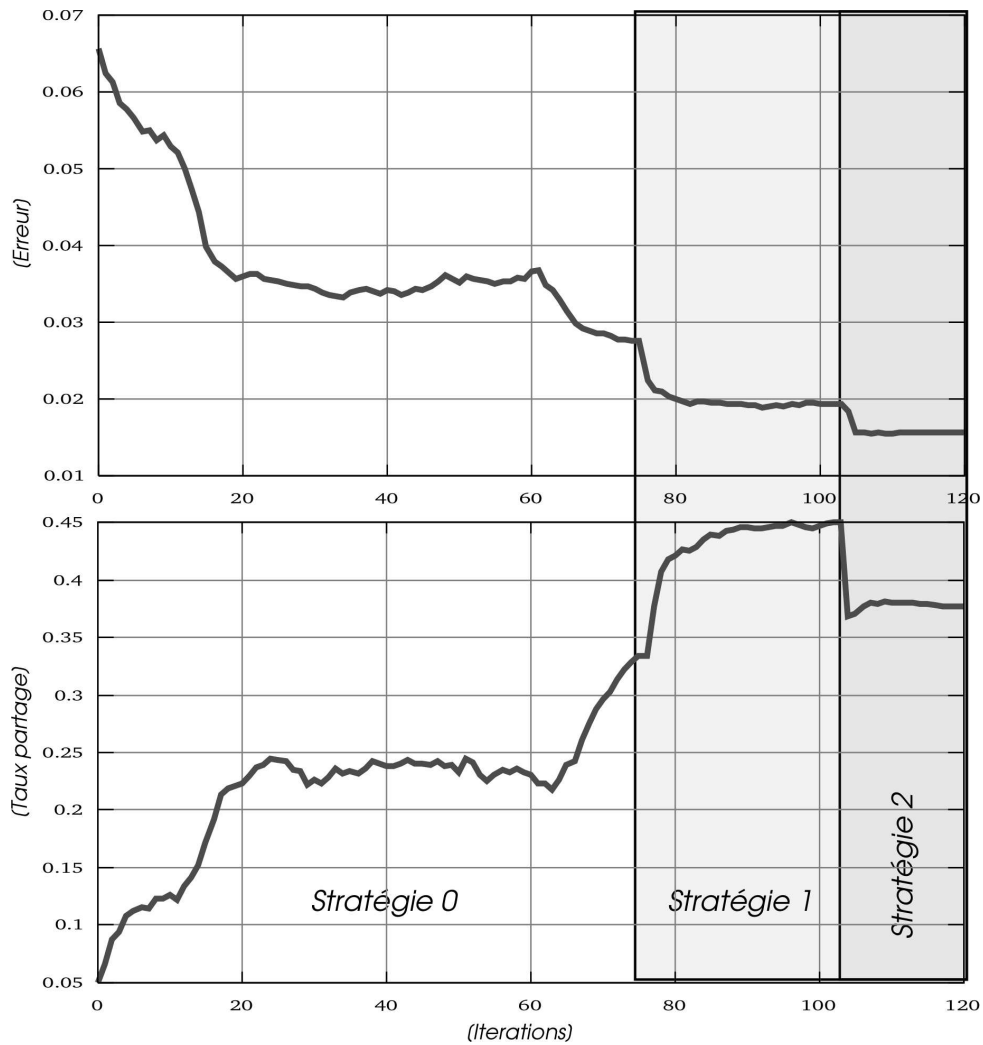


FIG. 2.23 – le changement de stratégies pendant un processus d'ICP.

- **Stratégie 1 :**  
si la dérivée sur l'erreur de recalage converge vers 0, le nuage ne bouge plus et l'algorithme change de stratégie : le voisinage des *clusters* est forcé à la valeur 2 et le filtre d'unicité est appliqué avec une profondeur de pile égal à 4.
- **Stratégie 2 :**  
finalement il faut se donner les conditions d'un recalage précis : l'espace de *clusters* est refait de nouveau pour prendre en compte la distance actuelle diminuée entre des nuages, car nous utilisons un nombre de *clusters* proportionnel à la longueur de l'arête de l'espace occupé par les deux nuages, qui a diminuée au niveau de cette étape. Par conséquent une reformulation des *clusters* donne une taille moindre de groupes de points dans chaque cellule, ce qui augmente la précision et la vitesse de la convergence.  
En plus, le voisinage se réduit à 1 *cluster* et l'empilement est réglé à la valeur 3.

Finalement, l'algorithme est arrêté par analyse de la dérivée de l'erreur d'alignement entre les nuages à recalcer. L'algorithme d'I.C.P. complet est illustré sur la figure 2.24. Le pseudo-code est donné dans l'annexe B.

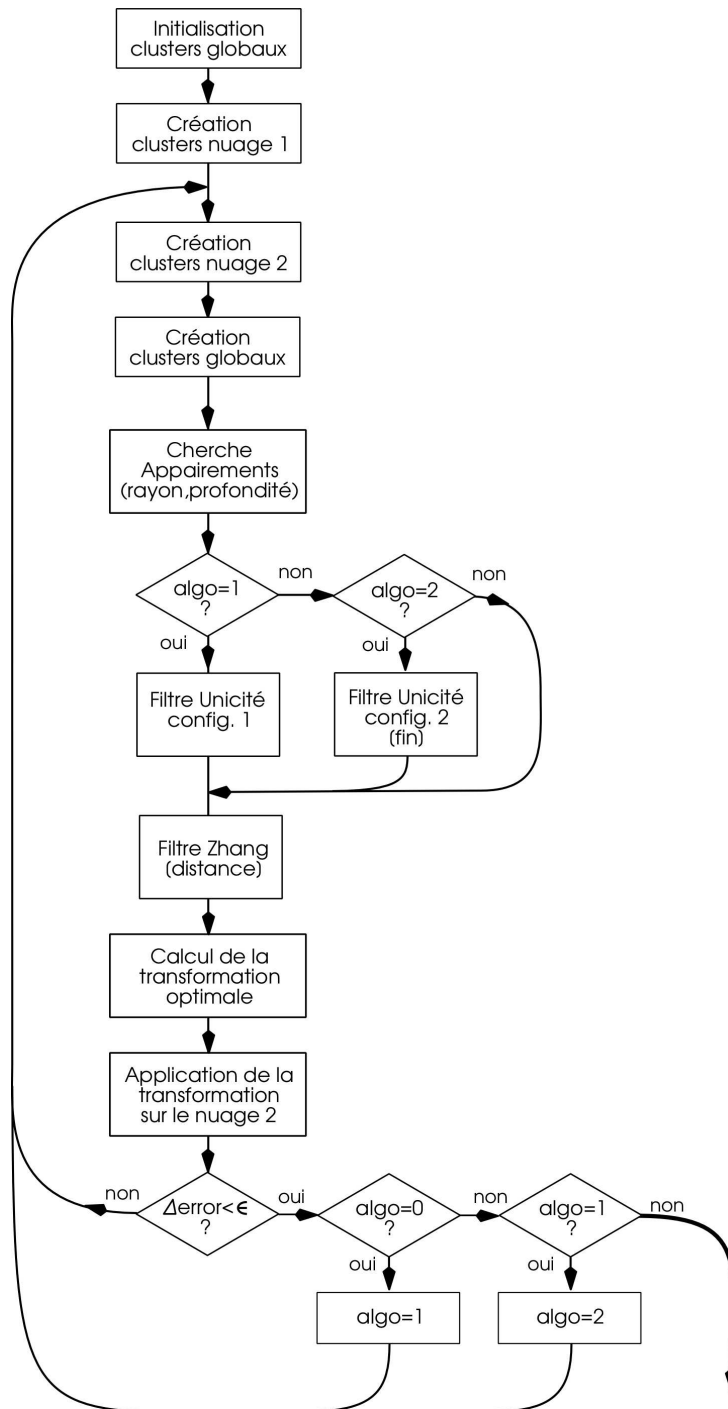


FIG. 2.24 – L’algorithme d’ICP.

## 2.8 Variantes dans l’application de la méthode d’ICP.

Nous avons décrit l’algorithme ICP qui permet de recalcr deux ensembles de points. Mais que sont ces ensembles ? Si nous disposons de  $N$  vues successives sur une scène, prises depuis des positions différentes, mais avec recouvrement des champs de vue, l’algorithme ICP peut être appliqué de plusieurs façons sur cet ensemble de vues.

Certaines stratégies peuvent rendre le recalage plus robuste, cela en améliorant deux étapes :

- la sélection de la transformation initiale et
- l'élimination des minima locaux proches du minimum global dans la convergence de la méthode.

Le principe des variantes décrites ci-dessous, consiste à multiplier les exécutions de ICP de vue à vue, puis entre les nuages obtenus par fusion partielle entre vues successives, cela pour améliorer chaque fois les transformées initiales, et donc, éliminer des mauvais appariements.

La figure 2.25 montre quelques variantes possibles en utilisant d'abord un recalage par paires, qui devient à la fin un recalage multi-vues.

- a) recalage incrémental classique ; que nous avons décrit ci-dessus : fusion successive de  $m_t$  et  $M_{t-1}$ .
- b) cycle de recalages entre trois vues ou plus, qui servent à corriger chaque transformation par paire en utilisant la transformation circulaire de la façon suivante :

$$T_{AB} = (T_{BC}T_{CA})^{-1}, T_{BC} = (T_{CA}T_{AB})^{-1} \text{ et } T_{CA} = (T_{AB}T_{BC})^{-1}.$$

La fusion des vues est faite après avoir appliqué quelques iterations d'ICP pour corriger les transformations initiales.

- c) recalage entre la fusion d'un cycle avec les autres vues comme dans le recalage incrémental.
- d) ce type de recalage qui mélange cycles et vues, n'est généralement pas possible en pratique, car il n'est pas faisable de créer des cycles entre toutes les vues, vu les recouvrements limités entre champs de vue.
- e) cycle de recalage entre cycles fusionnés. Cette configuration peut être une alternative au recalage incrémental qui peut réduire l'erreur finale.

De plus, pour optimiser les temps d'exécution, la séparation des recalages par groupes permet d'utiliser une configuration de cluster d'ordinateurs pour le calcul en parallèle.

Un exemple de recalage par cycles est possible avec l'ensemble d'images acquises sur l'objet BUNNY. La structure de recalage par paire est illustrée sur le graphe des recouvrements inter-vues présenté en figure 2.26 : deux vues sont connectés si au moins 30% de champ de perception et commun . Ces images en particulier, du fait de recouvrements multiples entre les champs de vue, permettent un recalage entre de nombreuses paires acquises de plusieurs directions, ce qui permet de générer de nombreux cycles d'ICP.

## 2.9 Vérification quantitative.

Nous proposons dans cette section, de comparer le comportement de l'algorithme ICP selon le type de points de contrôle extraits depuis les images d'origine : points de contour ou sommets de maillage.

Nous faisons une évaluation systématique en utilisant plusieurs groupes d'images de profondeur ; ces images (Bunny, Curvblock...) ont été utilisées par de nombreux chercheurs à des fins de comparaison ou évaluation d'algorithmes ; elles comportent des composantes structurées et de forme libre. Nous avons exploité les bases de données d'images de "Stanford 3D scanning repository" [3Dscanrep sh] et "OSU (MSU/WSU) data base" [3DDB sh] pour obtenir ces images.

Chaque image contient une vue à partir d'une position différente du capteur vis-à-vis d'un objet. Pour certaines séquences la position du capteur est connue avec une précision acceptable, qui sera utilisée comme vérité terrain pour comparer les résultats obtenus avec l'algorithme ICP.

Les résultats présentés dans cette section ont été obtenus à l'aide des séquences d'images sur les objets suivants :

- objets non structurés : BUNNY (12 images) en figure 2.27, DRAGON (6 images) en figure 2.28, HAPPYBUDDHA (6 images) en figure 2.30 et DOUGHBOY (8 images) en figure 2.31 à droite.
- objets structurés : BLOCK2 (5 images) en figure 2.32 à gauche, CURVBLOCK (5 images) en figure 2.31 à gauche, WYE (2 images) en figure 2.29 et VALVE (8 images) en figure 2.32 à droite.

Dans cette évaluation nous ne nous sommes pas intéressés à la fusion incrémentale, parce que l'objectif est de comparer le comportement de l'algorithme ICP avec différents types de points de contrôle, les sommets de maillage et les points de contour.

Ci-dessous, nous décrivons la méthodologie proposée et suivie pour cette comparaison, puis nous présentons les critères retenus en disposant de plusieurs images d'illustration. Finalement les critères sont

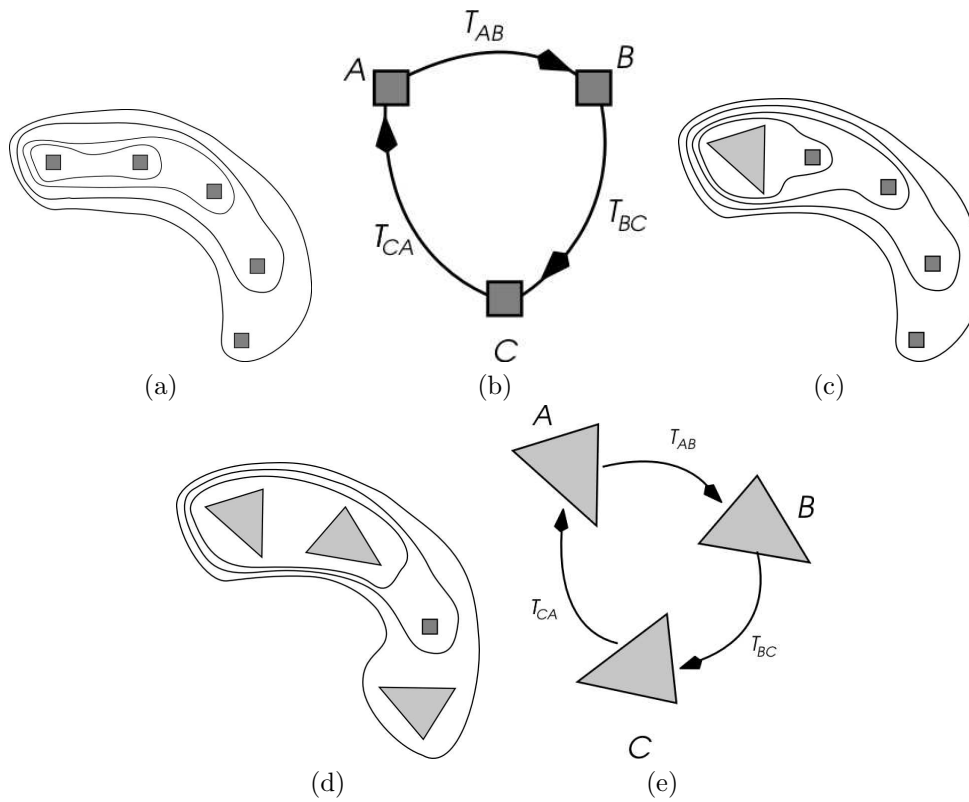


FIG. 2.25 – Variantes d'applications dICP.

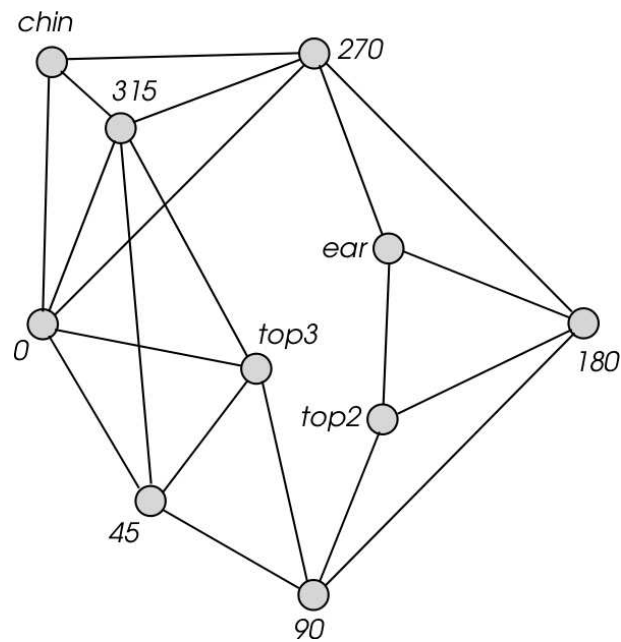


FIG. 2.26 – Exemple de recalages cycliques triangulaires sur l'ensemble d'images de BUNNY.

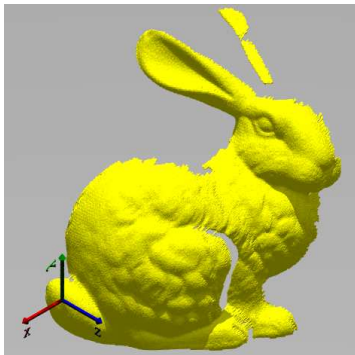


FIG. 2.27 – BUNNY.

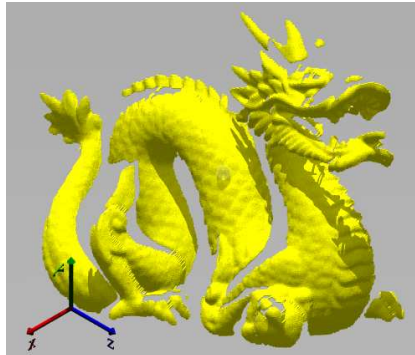


FIG. 2.28 – DRAGON.

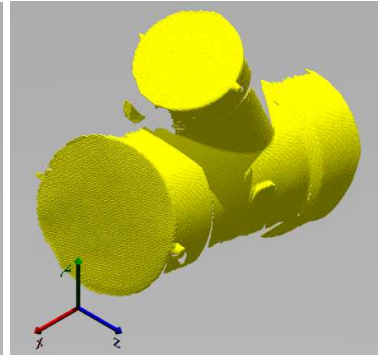


FIG. 2.29 – WYE.



FIG. 2.30 – HAPPYBUDDHA.

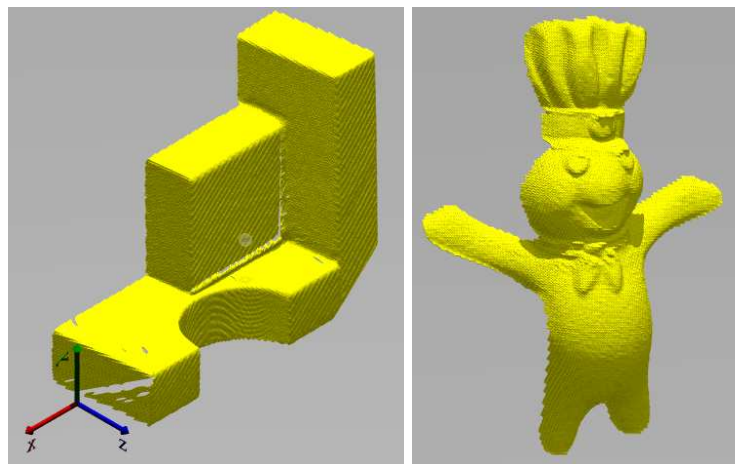


FIG. 2.31 – CURVBLOCK et DOUGHBOY.

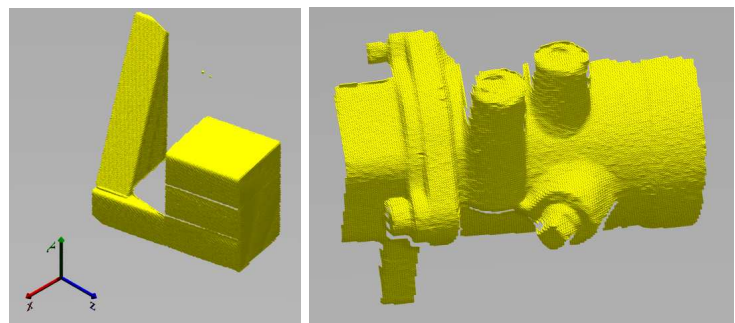


FIG. 2.32 – BLOCK2 et VALVE.

examinés l'un après l'autre avec une centaine d'essais de recalage en considérant huit groupes d'images ; les résultats statistiques globaux sont présentés et discutés à la fin de cette section.

### 2.9.1 Méthodologie de comparaison.

Le processus de recalage dépend de plusieurs facteurs. En conséquence pour comparer les recalages soit sur les points de maillage soit sur les points de contour, nous proposons une analyse de la robustesse de l'algorithme ICP en faisant varier les critères suivants :



- les types de surfaces présentes dans les scènes : surfaces de forme libre (typiquement, des sculptures) ou structurés (typiquement, des pièces industrielles, ayant des facettes polygonales, cylindriques),
- les conditions initiales,
- la résolution des nuages de points à recalcr,
- le taux de recouvrement entre les vues.

La comparaison des deux alternatives pour sélectionner des points de contrôle, a été effectuée à partir de ces quatre critères. La robustesse du recalage est étudiée en détail, en analysant, pour chaque méthode d'extraction des points de contrôle, comment elle se comporte en fonction des critères résolution, conditions initiales, recouvrement des vues et type de scènes.

Par contre, l'analyse du temps d'exécution en fonction de l'erreur de recalage, consiste à observer les résultats de recalage sur points de contour ou points de maillage, mais en considérant les mêmes choix pour les autres critères. Ainsi chaque fois que nous mentionnons un test de recalage, nous nous référons à la comparaison des deux recalages, un avec maillage et l'autre avec contours, exécutés avec les mêmes conditions de résolution et conditions initiales.

Pour ce faire, nous avons préparé une base de données de paires de représentations en maillage et contours, dans des conditions de résolution exactement comparables.

Robustesse et performance d'une fonction de recalage sont mesurées sur plusieurs éléments :

- La convergence ou non convergence. En cas de convergence, nous notons le nombre des itérations et la monotonie de l'erreur d'ICP.
- L'erreur finale du recalage par ICP qui correspond à la distance moyenne entre les points appariés.
- L'erreur de l'estimée finale de la transformation entre vues, par rapport à la vérité terrain quand elle est connue.
- Le temps de calcul du processus global, qui comprend l'extraction des points de contrôle jusqu'à l'obtention des résultats de recalage.

## 2.9.2 Robustesse par rapport aux conditions initiales.

### Conditions initiales et types d'objet

Le recalage des images de profondeur est un problème d'optimisation. L'algorithme d'ICP est fondé sur l'hypothèse qu'une transformation estimée  $\mathbf{T}_0$  entre deux vues est connue à l'avance. Une fonction de coût est définie, qui mesure la qualité de l'alignement obtenu avec  $\mathbf{T}_0$ , entre les surfaces en recouvrement partiel contenues dans chaque vue. Ces vues sont recalées en calculant la transformation 3D qui minimise la fonction de coût.

La question est de trouver une transformation approximative appropriée pour que la méthode converge. Comment estimer la précision requise sur l'approximation initiale  $\mathbf{T}_0$  de la transformation entre deux vues? Dans la plupart des cas, il n'y a aucune information sur cette valeur : comme la transformation initiale idéale n'est pas connue, la réponse à cette question repose donc sur une analyse du comportement expérimental.

Nous avons tout d'abord voulu comparer les approches Contour et Maillage pour l'ensemble des images de test, avec des conditions initiales choisies par l'utilisateur, les mêmes pour les deux approches.

- pour les images acquises sur des surfaces de forme libre, la technique basée sur les contours a réussi dans 76% des cas, tandis que la méthode basée sur le maillage triangulaire a du succès dans 92% des cas.
- au contraire dans les images acquises sur des scènes structurées, la technique basée sur les contours a réussi dans 95% des cas, par comparaison aux 85% obtenus par la procédure avec le maillage triangulaire.

Les cas où aucune des deux méthodes n'ont réussi à recalcr les vues, (environ un cas sur deux) ne sont pas considérés. Comme prévu pour ce critère de comparaison, le recalage basé par contour donne de meilleurs résultats pour les objets structurés du fait que sur ces objets, l'information pertinente est sur les contours; dans ces cas, la décimation de maillage doit créer des sommets au milieu des faces, qui ne correspondent en rien à des points discriminants.

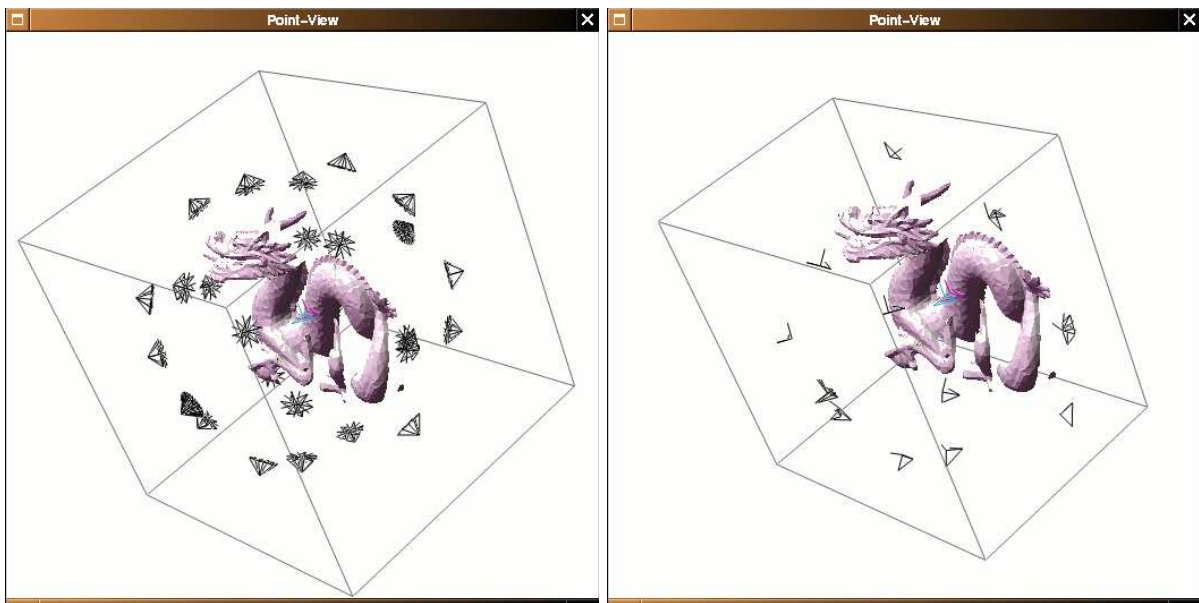


FIG. 2.33 – Exemple 1, DRAGON : A gauche, toutes les positions initiales, présentés comme des croix. A droite, les positions qui ont produit une bonne convergence pour les tests fondés sur les contours.

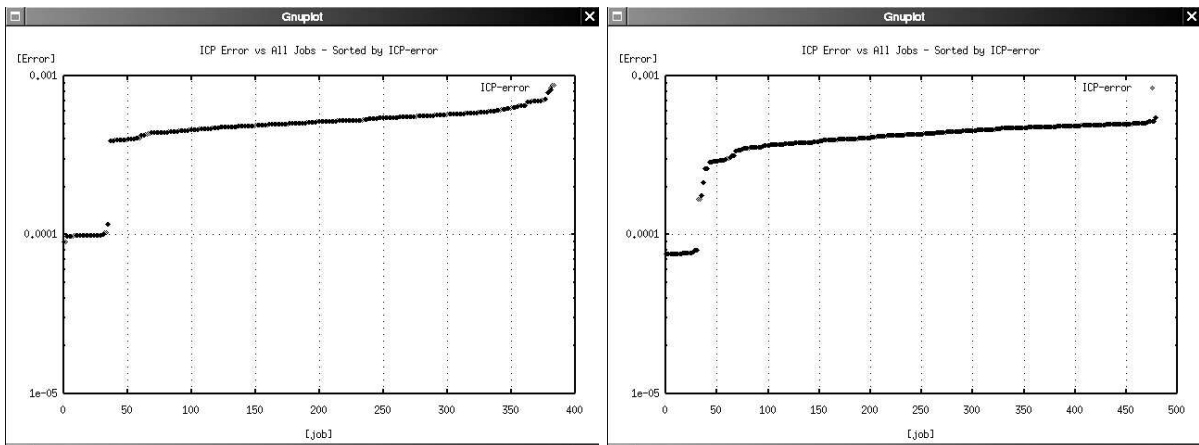


FIG. 2.34 – Exemple 1, DRAGON : toutes les erreurs d'ICP, à gauche, pour les tests basés sur les contours et à droite, pour les tests basés sur les sommets de maillage.

### Etude de la plage de stabilité par rapport aux conditions initiales.

A présent, notre objectif est d'évaluer le secteur de stabilité pour la transformation  $\mathbf{T}_0$ , plus exactement estimer la différence permise maximale entre la transformation initiale et les transformées finales  $\mathbf{T}$  dénotant une rotation  $\mathbf{R}$ , et une translation  $\vec{t}$ . De la description de la méthode d'ICP, l'erreur sur la translation initiale  $\vec{t}_0$  n'a aucun effet sur le temps de convergence; cette erreur sera convertie dans un ex-centrage pour chaque distance entre les points appariés, et par conséquent sur l'erreur de  $\varepsilon_0$ . Cette erreur de translation ne peut pas causer par elle-même des erreurs d'appariement, et sera réduite pendant les premières itérations.

Au contraire, une erreur importante sur la rotation initiale  $\mathbf{R}_0$  peut permettre beaucoup d'appariements faux avec des erreurs uniformes, donc non filtrés par la méthode statistique; de ce fait, une erreur importante sur la rotation initiale peut produire soit une non convergence, soit, plus grave, une solution fautive pour la méthode d'ICP, donc une mauvaise fusion d'une image dans le modèle construit.

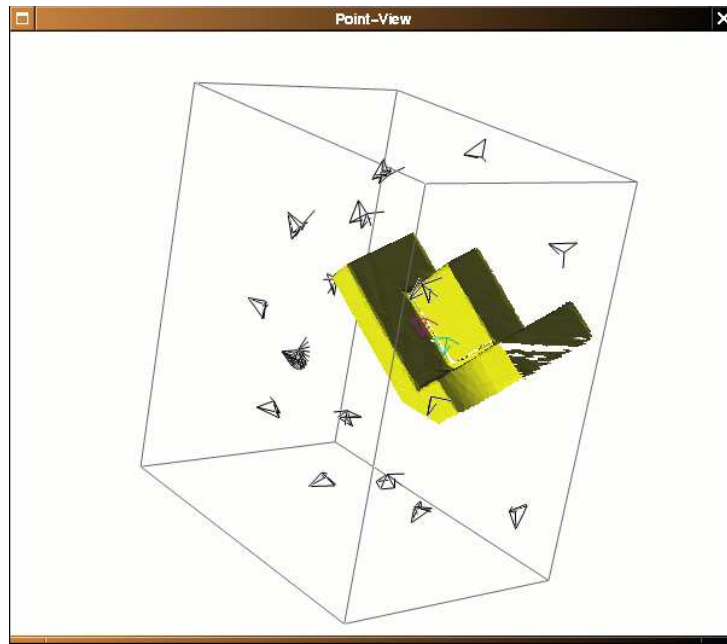


FIG. 2.35 – Exemple 2, CURVBLOCK : les positions qui ont produit une bonne convergence pour les tests à partir de la méthode du maillage.

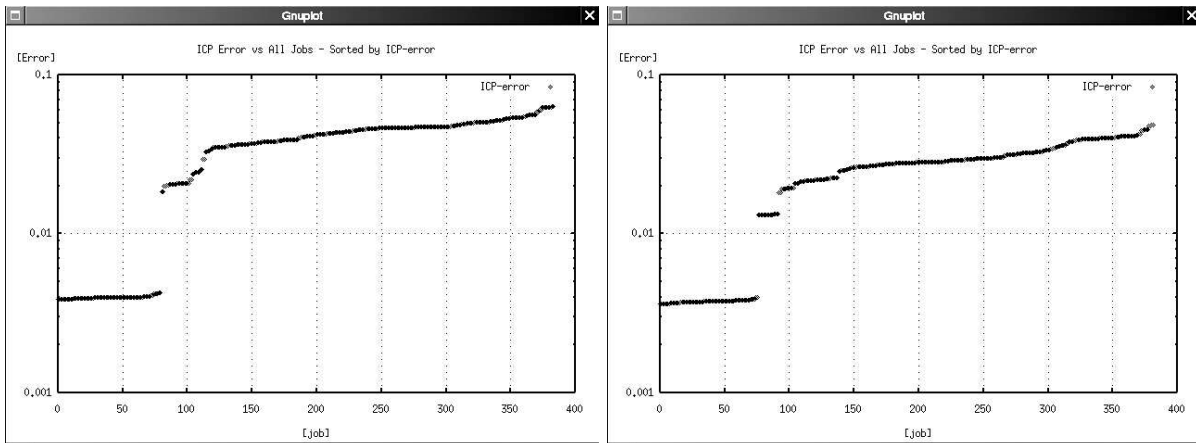


FIG. 2.36 – Exemple 2, CURVBLOCK : toutes les erreurs d'ICP, à gauche, pour les tests obtenus à l'aide de la technique des contours et à droite, pour les tests à partir de la méthode des maillages.

Notre objectif est donc d'évaluer le comportement de l'algorithme ICP -convergence ou non convergence- en fonction des paramètres initiaux de rotation. Un algorithme à deux étapes a été mis en application :

- Pour chaque angle d'Euler une liste de  $N$  valeurs initiales, uniformément réparties autour de la circonférence de 360 degrés, est calculée.
- Ensuite dans la deuxième étape, toutes les combinaisons possibles des valeurs précédentes pour les angles d'Euler, sont obtenues, combinées avec une translation de  $\vec{t} = (0, 0, 0)$  et employées pour définir la position de départ pour nos méthodes d'ICP. Cela conduit à tester au total,  $N^3$  transformations initiales pour l'algorithme ICP (typiquement,  $N$  est égal à 10) basé sur le recalage de points de contour, puis sur les sommets de maillage triangulaire adaptatif.

Toutes les positions initiales sont présentées sur une sphère centrée sur l'objet DRAGON sur la figure 2.33 à gauche ; le résultat de l'analyse pour ICP sur les points de contour est présenté à droite ; ne sont conservées que les transformations qui ont conduit à la convergence. Les graphiques présentés en figure 2.34 montrent

les valeurs de l'erreur ICP triées de la plus petite (quand l'algorithme converge) à la plus grande (quand il diverge) ; à gauche pour les 1000 exécutions sur les points de contour, à gauche pour les 1000 exécutions sur les points de maillage.

La ligne droite, à proximité du côté gauche des graphiques représente la convergence correcte dans le test. Pour cet objet, cette convergence est obtenue seulement pour 35 transformées initiales sur les 1000 testées.

Après avoir vérifié toutes les combinaisons possibles, le recalage basée sur points de contour donne un taux de succès de 15% pendant que le recalage basée sur le maillage triangulaire arrive a un taux de 14% pour les objets de forme libre. Les résultats pour les positions initiales, qui donnent une convergence correcte, sont illustrés sur les figures 2.33 et 2.37 sur les objets BUNNY et DRAGON.

Par contre, pour des objets structurés, le recalage a conduit à un succès avec un taux de 20% pour les contours et 19% pour le maillage, comme nous le présentons sur la figure 2.35. Les cas où aucune des deux méthodes ne convergent à un résultat positif sont invalidées.

Une caractéristique intéressante apparaît sur les graphiques d'erreur des figures 2.34, 2.36 et 2.38 : ils donnent plusieurs informations

- la différence entre les bonnes estimées initiales, qui conduisent à une convergence correcte, et les mauvaises, qui mènent à une erreur, est clairement identifiable. Nous pouvons envisager un test de vraisemblance sur le résultat ICP, et ainsi détecter les erreurs dues à une mauvaise initialisation.
- quand l'algorithme converge, il donne à la fin la même erreur d'alignement, et la même solution finale. Ce phénomène est constaté dans la figure 2.38 : à gauche, les erreurs des exécutions d'ICP classées par ordre croissant ; à droite, dans le même ordre, les erreurs de l'estimation du quaternion. L'analyse de ce graphique montre que, pour le plateau supérieur de l'erreur ICP (exécutions 100 à 500), l'estimation est fautive (minimum local) ; pour le plateau inférieur, elle est juste et l'erreur d'estimation est corrélée à l'erreur d'ICP.
- par contre, comme vu sur les figures 2.33, 2.37 et 2.35 qui montrent les estimées initiales qui mènent à la convergence, ces estimées ne forment pas un ensemble cohérent dans l'espace des rotations.

A propos du temps de calcul utilisé pour cette évaluation, une exécution d'ICP a besoin de quelques secondes (cela dépend du nombre de points de l'image à comparer). Pour réduire le temps de calcul pour nos essais de 1000 exécutions d'ICP approximativement par objet, avec deux méthodes différentes, nous avons utilisé activer plusieurs exécutions en parallèle, sur 2, 4 ou 8 CPUs.

### 2.9.3 Robustesse par rapport à la résolution des images.

Cette comparaison a été exécutée en traitant les maillages triangulaires décimés à différentes valeurs et les représentations par contours produites avec plusieurs erreurs d'approximation. Les conditions initiales ont été les mêmes pour chaque représentation et elles sont choisies pour assurer la convergence de la méthode de recalage.

Les représentations à recalculer sont formées de la façon suivante. Initialement nous préparons les contours des images originales à différentes résolutions en utilisant l'erreur d'approximation comme paramètre de contrôle. Ensuite le maillage triangulaire est produit par décimation à partir de l'image originale en imposant le nombre de sommets obtenu pour la même image, avec l'approche extraction des contours.

Tout d'abord la méthode d'ICP a été appliquée au groupe des images du DRAGON ; voir la figure 2.39, qui montre les représentations pour le maillage en haute et basse résolution ; nous avons exécuté une évaluation similaire sur les points de contour. Les résultats sont résumés sur le tableau 2.1. Pour les différentes résolutions, nous avons noté que l'ICP basé sur les sommets de maillage converge toujours plus vite et réussi à appairer une quantité supérieure de points. Pour le recalage sur points de contour, nous mentionnons l'erreur d'ICP avant convergence (pour l'itération finale du recalage sur les sommets) et l'erreur d'ICP après convergence. Selon la résolution ou le type d'objet, c'est l'une ou l'autre de ces méthodes qui donnera la meilleure précision.

Ce résultat est confirmé par l'analyse statistique finale, ou la représentation par maillage démontre sa supériorité pour des objets non structurés. Néanmoins un point positif pour les deux méthodes est

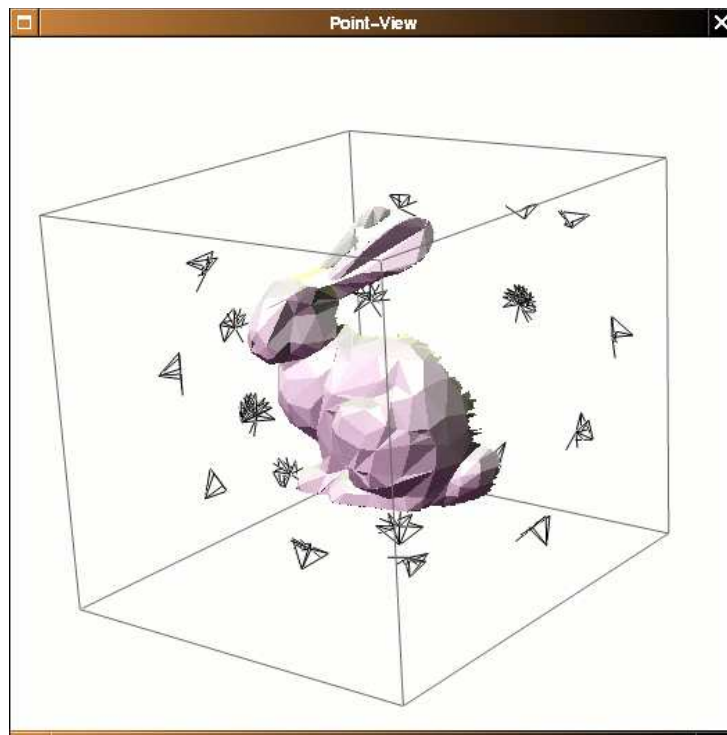


FIG. 2.37 – Exemple 3, BUNNY : les positions qui ont produit une bonne convergence pour les tests basés sur maillage.

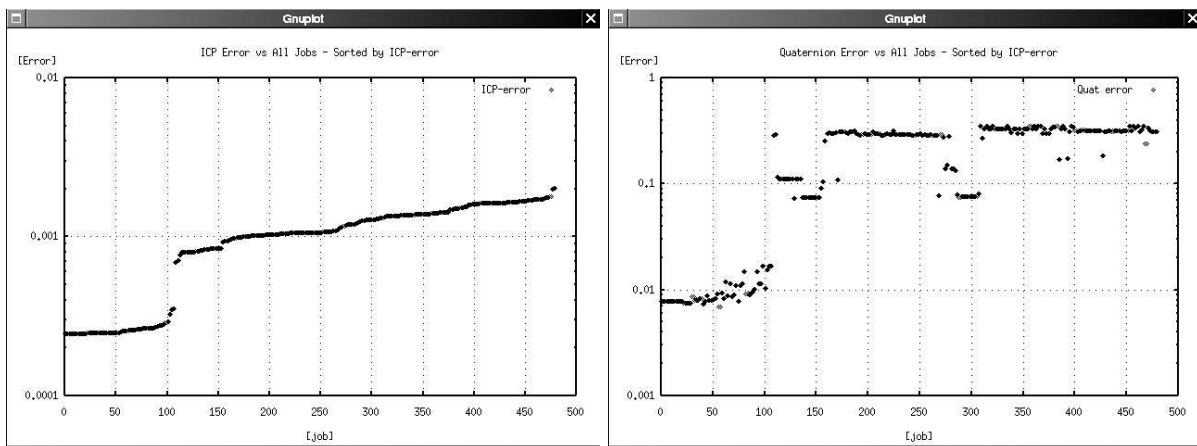


FIG. 2.38 – Exemple 3, BUNNY : à gauche, toutes les erreurs d'ICP et à droite, l'erreur comparée à la position connue pour les tests à partir de la méthode du maillage.

que l'ICP peut réussir à recalibrer des nuages de points avec une résolution très basse, jusqu'à 350 points dans nos essais. Si c'est nécessaire pour l'application étudiée, un recalibrage grossier pourrait fournir un état initial acceptable pour activer un recalibrage précis en utilisant des images de plus haute résolution.

Cette stratégie a été évaluée et son intérêt confirmé par des temps de calcul, requis pour recalibrer deux images de profondeur

- en utilisant seulement la résolution la plus élevée,
- en exploitant deux niveaux de résolution : un recalibrage grossier avec une résolution inférieure, et puis une amélioration en utilisant la résolution la plus haute.

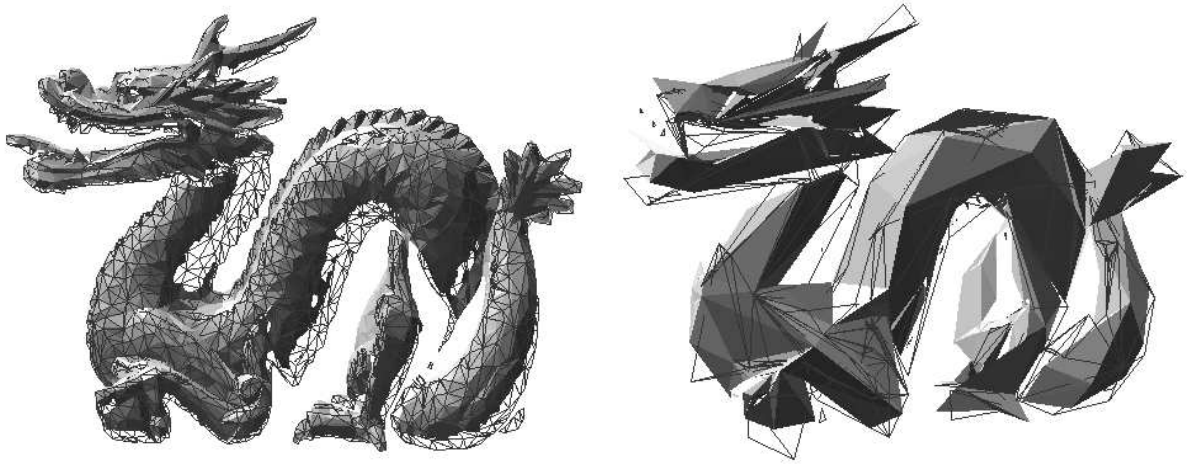


FIG. 2.39 – Évaluation de la méthode d’ICP par rapport à la résolution de l’image : le recalage utilise le maillage à haute (gauche) et à basse résolution (droite).

Le recalage en une seule étape a un temps d’exécution 500 fois plus important que le temps de calcul pour le processus divisé en deux étapes.

Puis, en utilisant l’objet HAPPYBUDDHA, le recalage des images à partir d’ensembles de points avec différentes résolutions, a été évalué. La figure 2.40, à droite, présente l’évolution de l’erreur d’ICP le long des itérations en appliquant la méthode d’ICP basée sur le maillage et des représentations de contour, avec différentes combinaisons des résolutions :

- recalage entre deux images de haute résolution d’environ 5000 points, désigné par H-H.
- recalage entre deux images de basse résolution d’environ 2000 points, désigné par L-L.
- recalage entre deux images l’une de haute, l’autre de basse résolution, désigné par H-L.

Les deux méthodes ont montré une robustesse acceptable pour ce critère. En principe, les techniques devraient converger plus rapidement quand une des représentations à recalier a un nombre de points réduit ; en fait,

- pour le recalage des sommets de maillage, les trois courbes sont très étroites et les nombres d’itéra-

Image1 : nb. points	3365	3265	1835	1774	720	695	509	331
Image2 : nb. points	3851	3723	1903	1836	894	857	668	427
<b>Recalage contours</b>								
No. itérations jusqu’à convergence	50	60	47	56	66	58	100	39
% points appariés	37%	41%	38%	27%	39%	38%	22%	30%
erreur ICP	0.00010	0.00010	0.00036	0.00025	0.00012	0.00031	0.00091	0.00077
erreur ICP jusqu’à convergence	0.000046	0.00010	0.00033	0.000044	0.00010	0.00031	0.00047	0.00078
<b>Recalage maillage</b>								
No. itérations jusqu’à convergence	31	53	27	19	55	49	35	35
% points appariés	36%	60%	56%	32%	60%	56%	41%	37%
erreur ICP	0.00017	0.000073	0.00019	0.00019	0.000070	0.00018	0.00055	0.00071
<b>Comparaison</b>								
No. minimal d’itérations	maillage	maillage	maillage	maillage	maillage	maillage	maillage	maillage
Meilleure précision	contour	maillage	maillage	contour	maillage	maillage	contour	maillage

TAB. 2.1 – Comparaison des deux méthodes par rapport à la résolution, pour l’objet de forme libre DRAGON.

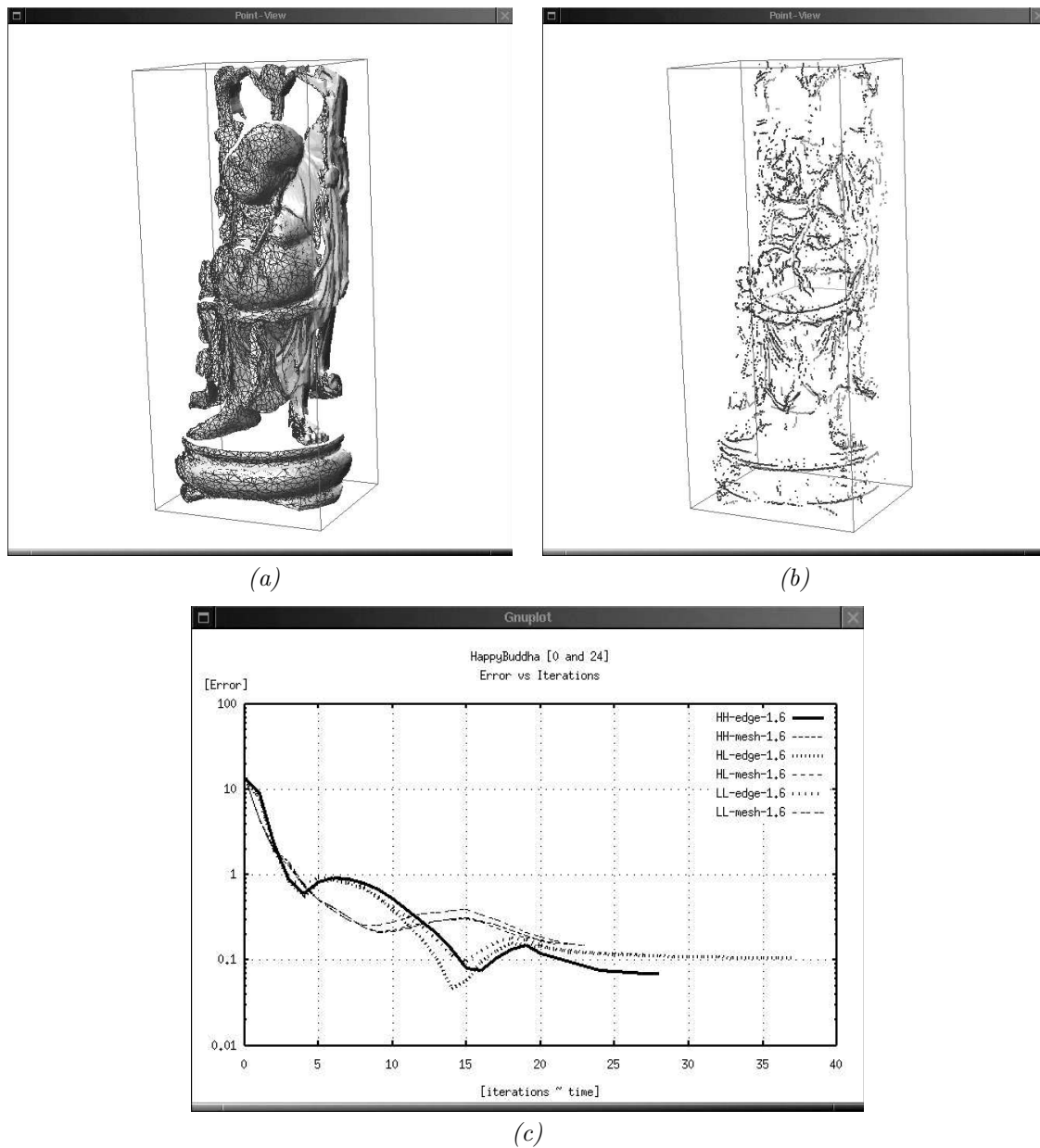


FIG. 2.40 – Evaluation de la méthode d’ICP avec des résolutions différentes. (a) recalage par maillage, (b) recalage par contours, (c) l’erreur d’ICP pour les combinaisons de résolutions.

tion sont égaux ;

- pour le recalage des contours, la convergence est plus rapide avec la haute résolution, grâce au nombre supérieur des appariements de points trouvés.

A la convergence, les erreurs d’ICP sont presque équivalentes, même si la technique des contours est légèrement meilleure pour cet objet qui a un relief complexe.

#### 2.9.4 Robustesse par rapport aux taux de recouvrement des vues.

L’évaluation de la robustesse en ce qui concerne le taux de recouvrement entre les deux vues à recalier, a été faite en utilisant deux objets : l’objet de forme libre DOUGHBOY et l’objet de forme structurée VALVE, pour lesquels nous avons exploité des séquences de vues prises en faisant effectuer au capteur, un

mouvement circulaire autour d'un axe de rotation centré sur l'objet, avec un pas de déplacement constant.

Sur les séquences des figures 2.42 et 2.43, la superposition des images est montrée pour chaque paire de recalage qui a donné une convergence correcte avec la vue prise à l'angle zéro. Dans les deux cas, les 8 figures supérieures représentent les recalages faits avec les sommets de maillage et les inférieures avec les points de contours. Pour l'objet DOUGHBOY, les degrés indiqués sont les angles de rotation autour du vecteur  $\vec{v} = (-0.03, -0.86, -0.50)$ , qui est la normale à la table d'acquisition ; pour l'objet VALVE, angles de rotation autour du vecteur  $\vec{v} = (-0.01, -0.93, -0.35)$ . La figure 2.41 donne une vue détaillée : à gauche, recalage sur DOUGHBOY, avec une différence d'angles de  $40^\circ$  entre les deux vues, et à droite, recalage sur VALVE, avec une différence d'angles de  $60^\circ$  entre les deux vues. Dans les deux cas, un recalage sur les sommets de maillage : une des vues est texturée, l'autre non.

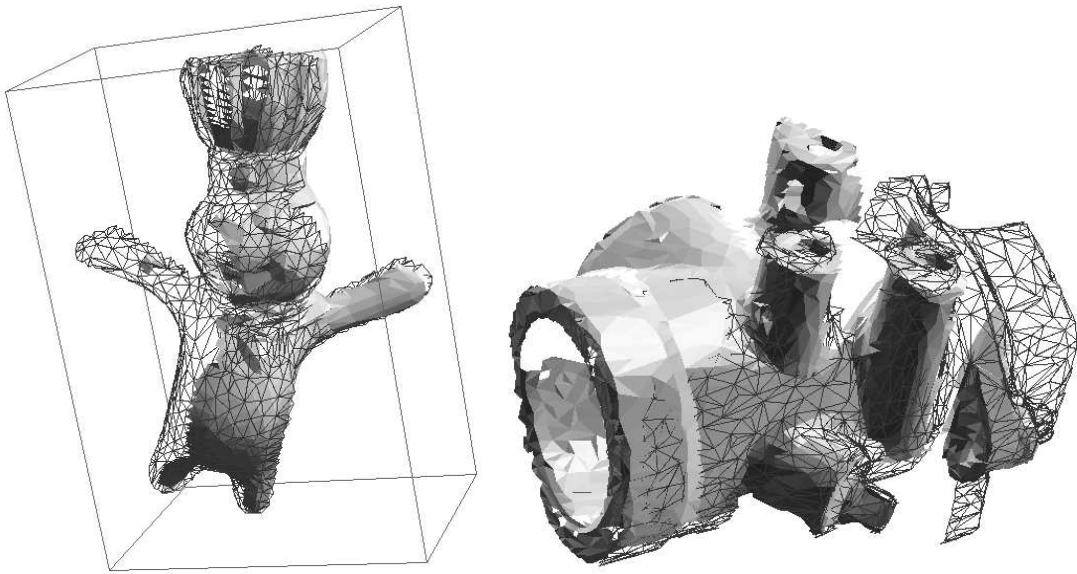


FIG. 2.41 – *Evaluation de la méthode d'ICP par rapport au taux de recouvrement entre nuages de points.*

Pour les deux objets, nous avons essayé de recalibrer une vue de référence acquise pour  $0^\circ$  d'angle de rotation, avec d'autres vues acquises chaque  $20^\circ$  de la position de référence, par exemple, à  $20^\circ$ ,  $40^\circ$ ,  $60^\circ$ ... jusqu'à ce que la non convergence soit atteinte. Cet événement se produit pour les deux représentations, points de contour ou points de maillage, à  $100^\circ$  et  $300^\circ$  pour la séquence de DOUGHBOY et  $80^\circ$  et  $280^\circ$  pour la séquence de VALVE. La vérité terrain (non disponible sur le site web) pour chaque vue est obtenue avec une exécution d'ICP précise en exploitant les images complètes.

Les tables 2.2 et 2.3 récapitulent tous les résultats pour les deux exemples ; quand les unités ne sont pas précisées, cela signifie que l'unité métrique (cm ? mm ?) pour les images 3D, n'est pas précisée sur le site Web. Nous pouvons noter que le taux de *clusters* appariés donne une mesure du recouvrement entre les vues, qui est plus représentative que le taux de points appariés. La convergence de la méthode d'ICP est limitée approximativement, à un taux de recouvrement minimal d'une valeur de 15%.

Dans la plupart des cas, les images des points de contour, qui ont un nombre inférieur de points, ont donné des résultats meilleurs si on compare le résultat obtenu avec la vérité terrain, donc par rapport à la position exacte donnée avec les images originales. Cette comparaison est donnée par les erreurs sur  $\vec{v}$  et  $\theta$ . Par contre l'erreur d'ICP est plus grande que ceux obtenus avec les sommets du maillage triangulaire. Ce comportement est normal car l'erreur d'ICP à la fin de la convergence diminue plus quand le nombre de points est plus important.



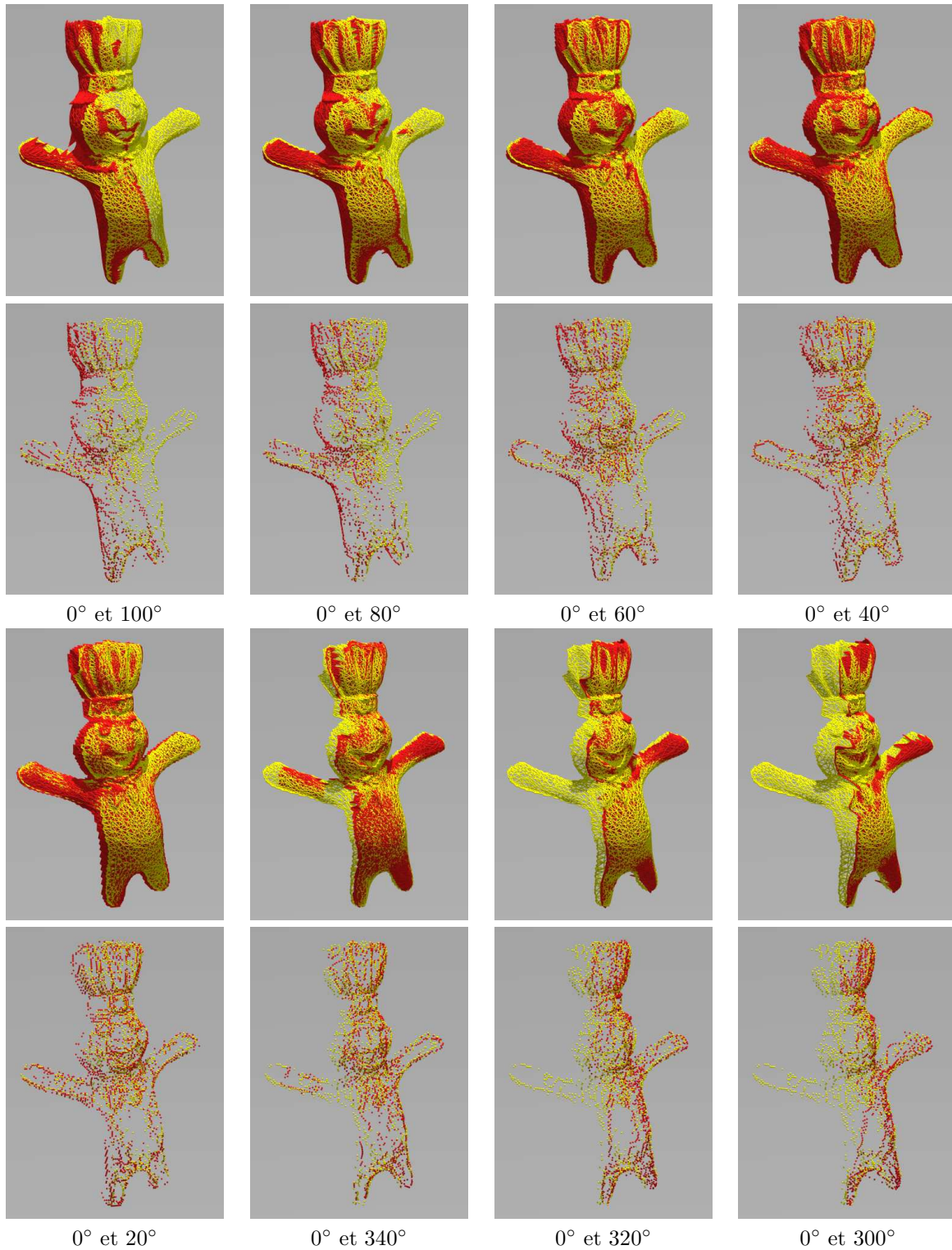


FIG. 2.42 – Evaluation de la méthode d'ICP par rapport au taux de recouvrement entre les nuages de points pour l'objet DOUGHBOY.

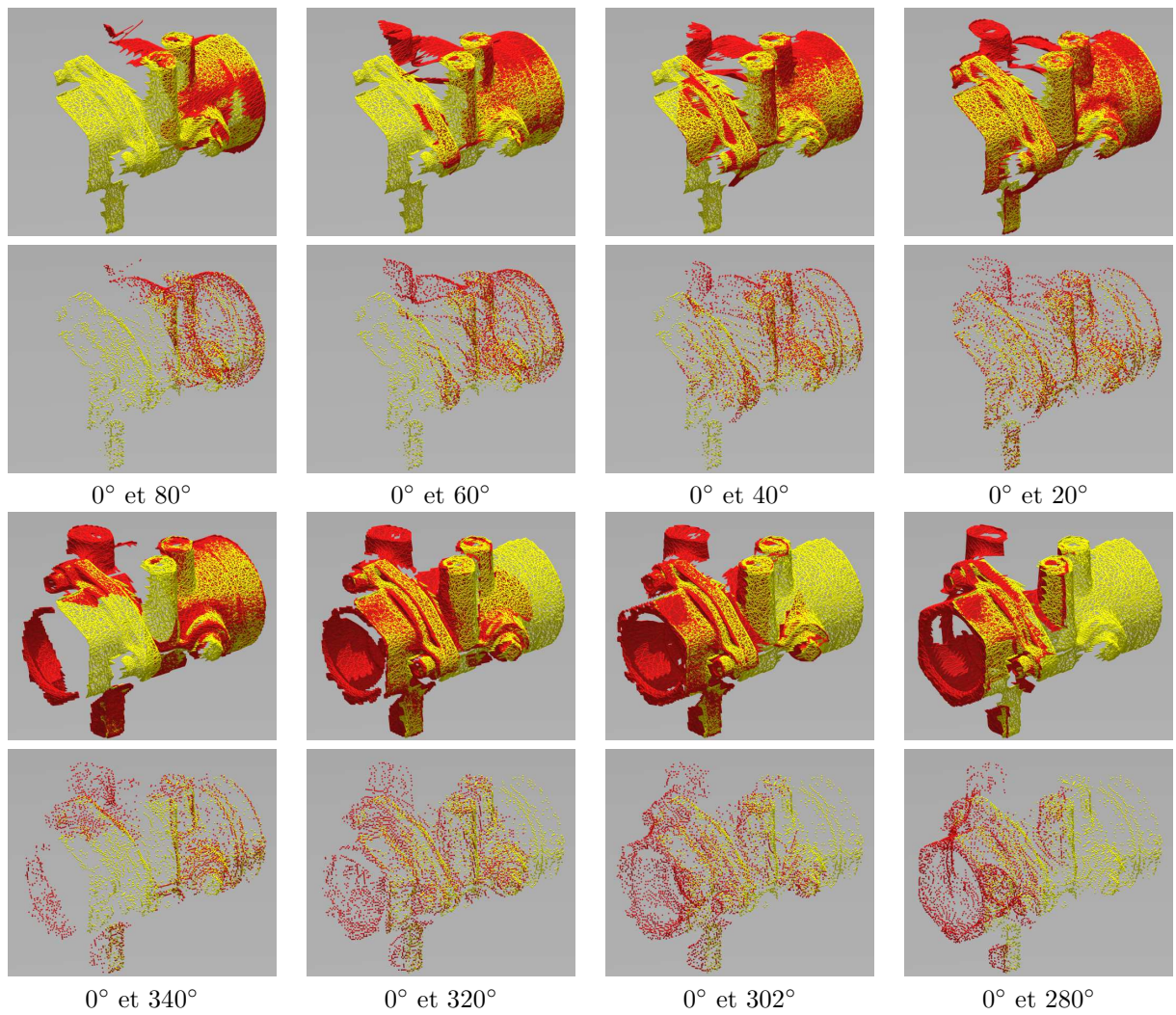


FIG. 2.43 – Evaluation de la méthode d'ICP par rapport du taux de recouvrement entre les nuages de points pour l'objet VALVE.

	Images recalés	0° – 100°	0° – 80°	0° – 60°	0° – 40°	0° – 20°	0° – 340°	0° – 320°	0° – 300°
<b>Recalage maillage</b>	Points nuage 1	2337	2495	2666	2734	2802	2626	2532	2324
	No. itér. jusqu'à converg.	90	176	35	150	74	49	76	105
	<i>Clusters</i> partagés	23.5%	29.4%	35.5%	46.1%	53.8%	55.7%	41.5%	35.4%
	Points nuage 0=2730	40.3%	50.7%	61.7%	71.1%	81.2%	78.4%	66.3%	56.4%
	erreur d'ICP	0.02079	0.01803	0.01553	0.01307	0.01139	0.01193	0.01428	0.01645
	erreur de $\vec{v}$	0.01108	0.01023	0.01090	0.01089	0.02048	0.02549	0.01692	0.01391
	erreur de $\theta$ [deg]	0.28067	0.45658	0.62380	0.44398	0.39519	0.38993	0.70112	0.98555
<b>Recalage contours</b>	Points nuage 1	1364	1459	1663	1811	1903	1591	1628	1488
	No. itér. jusqu'à converg.	80	161	27	133	69	59	127	236
	<i>Clusters</i> partagés	14.9%	19.3%	25.7%	32.6%	41.4%	38.1%	33.9%	28.1%
	Points nuage 0=1688	30.4%	40.7%	58.7%	68.0%	79.3%	71.3%	60.8%	47.5%
	erreur d'ICP	0.03631	0.02838	0.02391	0.01987	0.01631	0.01878	0.02295	0.02546
	erreur de $\vec{v}$	0.00441	0.00270	0.01544	0.01527	0.01611	0.02686	0.00235	0.00415
	erreur de $\theta$ [deg]	0.14037	0.11860	0.39765	0.37625	0.33093	0.37396	0.38509	0.11369

TAB. 2.2 – Comparaison des deux méthodes par rapport au taux de recouvrement pour l'objet "sculpturé" DOUGHBOY sur la figure 2.42.

	Images recalés	0° – 80°	0° – 60°	0° – 40°	0° – 20°	0° – 340°	0° – 320°	0° – 302°	0° – 280°
<b>Recalage maillage</b>	Points nuage 1	3056	3768	3981	4658	3295	3728	3438	3387
	No. itér. jusqu'à converg.	100	81	104	35	59	36	107	103
	<i>Clusters</i> partagés	21.5%	32.2%	50.5%	62.1%	41.7%	38.9%	31.0%	18.3%
	Points nuage 0=4141	27.4%	39.7%	58.6%	76.2%	49.4%	50.8%	35.8%	22.1%
	erreur d'ICP	0.21521	0.03766	0.02586	0.01957	0.02838	0.02835	0.03768	0.06236
	erreur de $\vec{v}$	0.007608	0.01102	0.01451	0.00903	0.00024	0.00356	0.00234	0.00385
	erreur de $\theta$ [deg]	0.58172	0.02928	0.11498	0.09120	0.24402	0.05398	0.29215	0.55983
<b>Recalage contours</b>	Points nuage 1	3024	3598	3285	3547	2346	3099	3312	3443
	No. itér. jusqu'à converg.	94	60	178	54	72	44	93	81
	<i>Clusters</i> partagés	20.1%	30.8%	42.7%	54.8%	32.9%	36.6%	28.2%	17.1%
	Points nuage 0=2910	26.9%	44.2%	60.3%	75.9%	45.6%	55.2%	42.0%	24.8%
	erreur d'ICP	0.06262	0.04916	0.03458	0.02726	0.03879	0.03514	0.04350	0.06963
	erreur de $\vec{v}$	0.001805	0.00826	0.01463	0.01051	0.01162	0.00695	0.00691	0.01208
	erreur de $\theta$ [deg]	0.66780	0.01348	0.05342	0.05138	0.16546	0.09960	0.01449	0.19405

TAB. 2.3 – Comparaison des deux méthodes par rapport au taux de recouvrement pour l'objet structuré VALVE sur la figure 2.43.

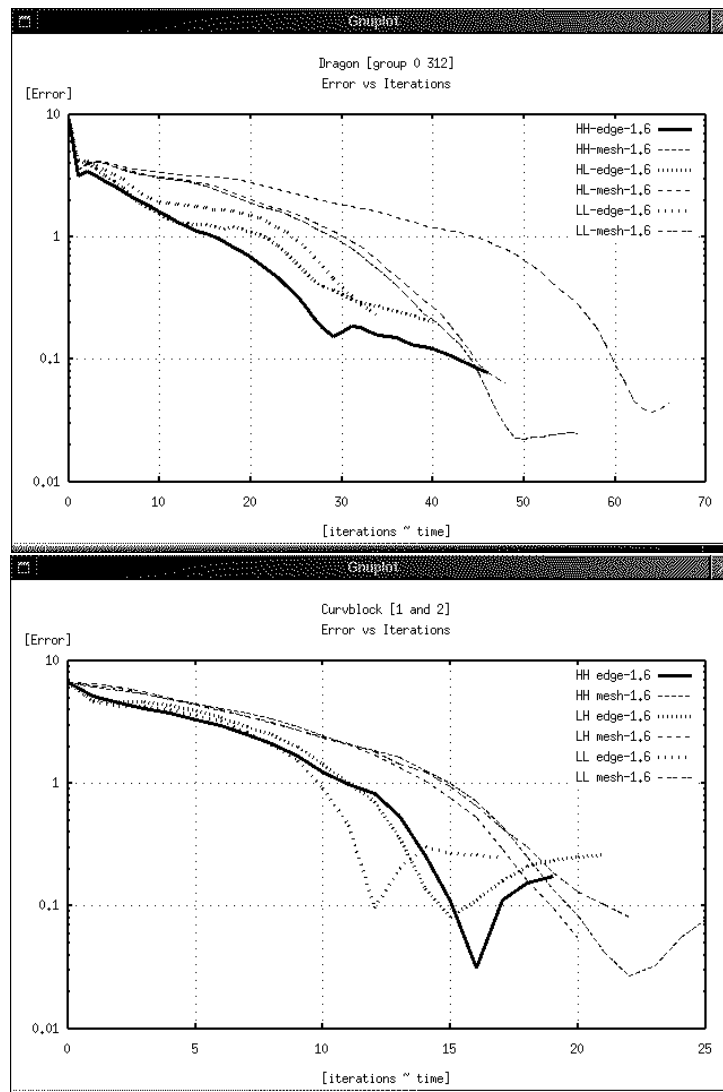


FIG. 2.44 – Diminution de l’erreur par rapport à la vérité terrain au cours des itérations pour les objets DRAGON et CURVBLOCK.

### 2.9.5 Nombre d’itérations en fonction de l’erreur de recalage.

Dans cette section, nous nous intéressons à la vitesse de convergence, directement liée au temps d’exécution. Dans cette évaluation, deux des critères précédents ont été considérés ensemble, la résolution (nombre des points de contrôle des ensembles donnés en entrée de ICP) et les conditions initiales (estimée initiale  $\mathbf{T}_0$ ). Puis, en utilisant les ensembles de points calculés dans les sections précédentes, un algorithme effectue d’une manière aléatoire une combinaison résolution/conditions initiales pour les représentations qui seront recalées. Ces valeurs sont considérées deux fois, en premier lieu pour le recalage fondé sur un maillage et en second lieu pour le recalage fondé sur les contours.

Les graphiques montrés sur la figure 2.44, représentent la diminution de l’erreur 3D le long des itérations d’ICP : le recalage des deux images de CURVBLOCK avec deux résolutions en utilisant les points de contour ou les sommets de maillage (en bas), les mêmes pour deux images de DRAGON (en haut). Les représentations à recalier contiennent la même quantité de points et leurs positions originales ont été les mêmes dans tous les exemples. La diminution des erreurs est plus régulière pour les images de

DRAGON; quelques discontinuités dans cette évolution, apparaissent (plus souvent pour les images sur CURVBLOCK). Cela correspond aux changements de stratégie sur le filtrage des appariements : quand le critère d'unicité est examiné afin de rejeter la majorité des appariements faux, l'erreur a tendance à croître de nouveau.

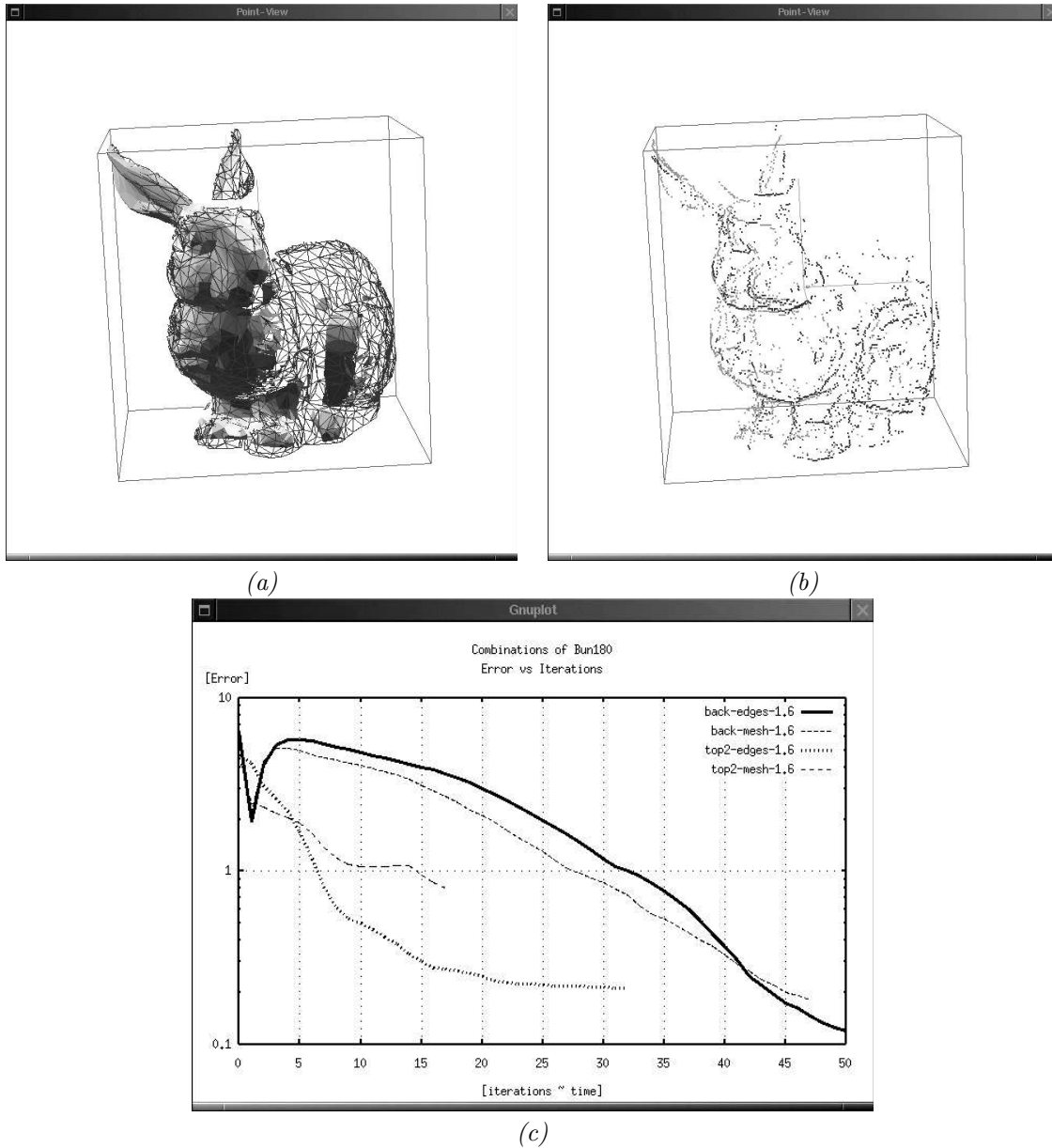


FIG. 2.45 – Évaluation des méthodes d’ICP sur plusieurs images de BUNNY : recalage avec sommets de maillage (a), recalage avec points de contour (b). Comportement de l’erreur d’ICP (c).

Sur le figure 2.45, les graphiques montrent les recalages de trois vues faites individuellement, paires par paires pour BUNNY : la vue de face de l’objet est recalée avec une vue supérieure (“top”) et encore la même vue de face avec une image arrière (“back”). Dans cet exemple, la technique de recalage basée sur les points de contours donne un résultat meilleur, car les points de discontinuité ont décrit les détails particuliers de la surface de cet objet d’une façon plus exacte. Pour le recalage de la vue de face avec la vue arrière, la diminution de l’erreur d’ICP pour les deux méthodes varie lentement car le taux de

recouvrement entre les vues est plus faible et par conséquent, l’erreur 3D finale est supérieure pour l’autre paire d’images.

### 2.9.6 Evaluation globale.

En plus des exemples présentés dans les sections précédentes, d’autres ensembles d’images de profondeur ont été considérés (4 ensembles sur des objets de forme libre et 4 ensembles sur des objets structurés). Plus de cent résultats d’ICP ont été comparés, en utilisant des images de plusieurs objets, avec différentes résolutions ou conditions initiales. Les résultats statistiques globaux pour chaque ensemble d’images, sont montrés sur la table 2.4.

Objet	CURVBLOCK	BLOCK2	VALVE	WYE	BUNNY	HAPPY	DRAGON	DOUGHBOY
Nombre de tests	23	10	7	2	8	20	28	7
Nombre de résolution	2	2	1	1	1	2	8	1
Non converg. maillage	1	4	1	0	0	1	3	1
Non converg. contour	1	0	1	0	0	7	7	1
Erreur moyen maillage	0.4336	2.7862	na	na	0.3657	0.4136	0.3793	na
Erreur moyen contour	0.2104	0.1738	na	na	0.3809	0.4721	0.5991	na

TAB. 2.4 – Résultats globaux sur tous les objets.

La convergence d’ICP est efficace dans la plupart des cas avec les points de contour pour les objets structurés (BLOCK2 par exemple), et avec les sommets de maillage pour les objets de forme libre (HAPPYBUDDHA par exemple). La même différence a été notée pour l’erreur 3D finale, elle est calculée par comparaison avec la vérité terrain, donc avec la position idéale de la vue, quand elle est disponible. Cette erreur montre que le résultat final est plus près de la position idéale pour le recalage par les points de contour pour les objets structurés que pour le recalage par les sommets de maillage pour les objets de forme libre.

Avec les objets structurés, les sommets supplémentaires produits par l’algorithme de décimation, à l’intérieur des facettes de l’objet, ne sont pas situés à la même position dans les différentes vues ; ainsi les appariements avec ces points ou bien sont rejetés par le filtre statistique (et, pour quelques images, l’ICP ne peut pas converger, ou il converge avec plus d’itérations), ou bien ils sont préservés, mais en rendant le résultat final d’ICP moins précis.

Par contre pour les objets de forme libre, les points de contour sont situés sur des discontinuités de courbure, et leurs positions ne sont pas assez stables entre les différentes vues, de sorte que l’algorithme ICP converge plus rapidement avec les sommets de maillage qu’avec les points de contour.

Un autre avantage potentiel des représentations basées sur les points de contour, vient de l’étiquetage des points de discontinuité. Sans employer comme dans [Hebert 98], un vecteur d’attributs pour décrire le relief local autour du point d’intérêt 3D, le résultat d’ICP peut être amélioré en utilisant la nature de la discontinuité : profondeur, orientation ou visibilité (frontières de l’image ou lignes d’occultation) :

- pour les objets non polyédriques, il est meilleur d’enlever les points de discontinuité de profondeur avant l’exécution d’un recalage : sur la figure 2.46, à gauche, deux images de WYE ont été recalées en utilisant seulement les discontinuités de normale, et à droite sur la même figure tous les points de discontinuité ont été considérés. Dans cet exemple (nous pouvons généraliser pour toutes les scènes contenant des objets structurés non-polyédriques comme les surfaces cylindriques par exemple), la convergence est plus rapide et précise en utilisant seulement les points de variation des normales (plis dans la surface), parce que les points de variation de la profondeur sur un objet cylindrique, appartiennent aux frontières extérieures de cette image, mais ne sont pas invariants selon le point de vue.

Pour le recalage avec les contours des objets de forme libre, figures 2.40 et 2.45, seulement des discontinuités de normale ont été considérées.

- un tel filtrage n’est pas possible pour les sommets de maillage, puisque la fonction de décimation n’indique pas la nature des sommets. Notons cependant, que le processus de recalage réussit également pour ces deux images (voir figure 2.47).

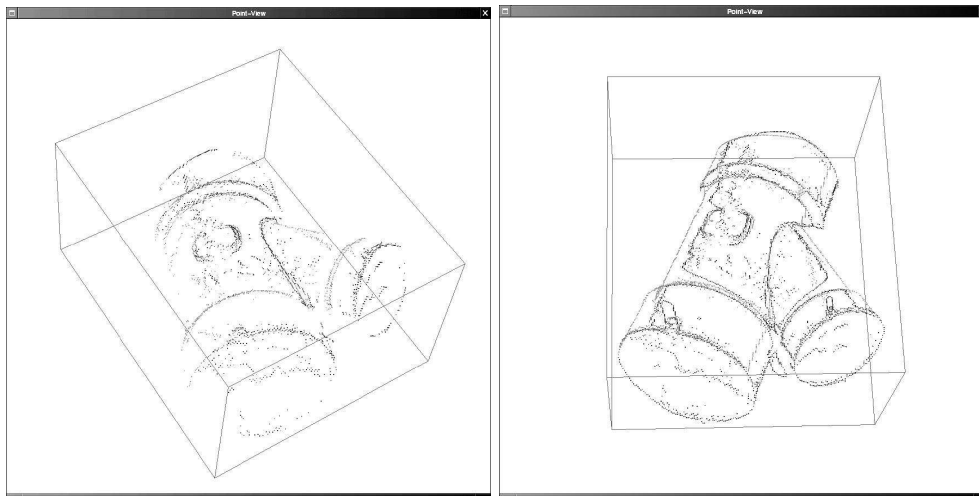


FIG. 2.46 – Evaluation de la méthode d’ICP sur l’objet cylindrique WYE, en utilisant seulement les contours de d’orientation à gauche, par contre, à droite tous les points de contour ont été utilisés.

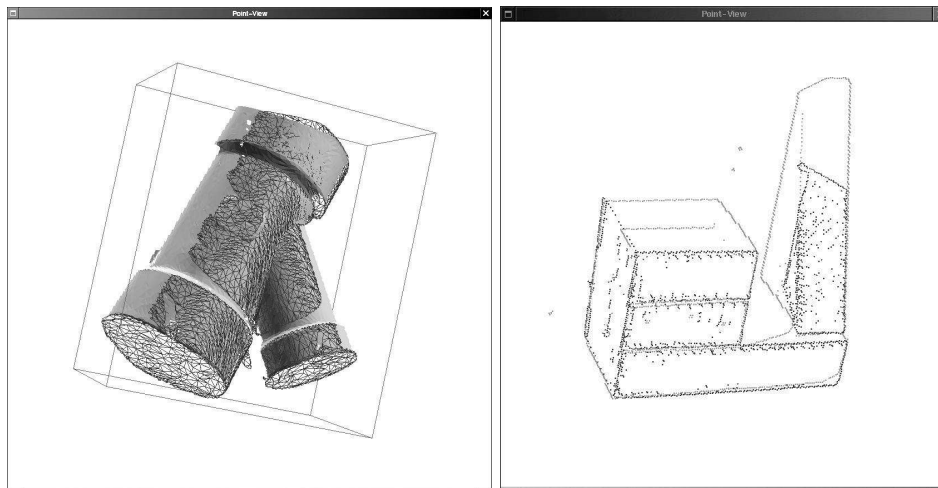


FIG. 2.47 – Evaluation de la méthode d’ICP : (gauche) deux vues de WYE recalées avec l’information du maillage. A droite, nous montrons deux vues de BLOCK2 recalées avec les contours.

- pour les objets polyédriques, les discontinuités de profondeur sont conservées pour le recalage, parce qu’elles peuvent s’apparier avec des discontinuités de normale dans d’autres vues. Sur la deuxième figure 2.47, deux images BLOCK2 ont été recalées en utilisant tous les points de contour, même les points qui correspondent aux frontières de l’image.

## 2.10 Conclusion.

Dans ce chapitre, nous avons présenté la méthode utilisée dans nos travaux, pour recalier des ensembles de points 3D. Cette méthode s’appuie sur l’algorithme ICP, *Iterative Closest Points*.

Notre version est très classique : l’appariement entre points 3D utilise exclusivement, le critère de distance euclidienne entre les points candidat. Nous limitons la complexité de cette approche, en sélectionnant des points de contrôle, soit des points de contour, soit des sommets de maillage. Nous gérons la possibilité d’avoir plusieurs appariements potentiels pour un point.

Mais, notre méthode n'exploite pas, deux types d'information qui permettraient certainement, de rendre ICP plus performant :

- des attributs locaux 3D, liés à la courbure ou au relief local : ce sont les *spin images* proposées par M.Hébert en particulier.
- des attributs photométriques, pour les images 3D acquises par stéréovision.

Une fois que les images 3D acquises sur un objet à modéliser, sont recalées, nous pouvons générer une représentation de cet objet. Le chapitre suivant traite donc de la génération d'un maillage triangulaire.





# Chapitre 3

## Fusion d'images 3D

### 3.1 Introduction.

Le recalage des images 3D, analysé au chapitre précédent, n'est que la première étape du processus qui permet de construire le modèle d'un objet 3D à partir de plusieurs images acquises depuis un capteur qui se déplace autour de l'objet, en des positions non connues a priori. Dans ce chapitre, nous nous proposons d'analyser l'étape suivante qui permet de fusionner toutes les données dans une représentation de cet objet.

Ce traitement de *fusion* et de *construction* du modèle, s'exécute soit à la fin du processus de recalage, soit de manière incrémentale après chaque recalage d'une image avec le modèle courant :

- la *fusion incrémentale* des données acquises depuis les différents points de vue, permet de mettre à jour un modèle après chaque acquisition.
- la *Construction* d'une représentation finale de l'objet, exécute une extraction de caractéristiques par segmentation ou une autre méthode, à partir du nuage de points 3D obtenu après recalage et simple concaténation des données (fusion des images 3D dans un nuage de points 3D).
- une méthode intermédiaire consiste à focaliser le traitement incrémental afin d'extraire des points représentatifs de chaque image et de limiter la quantité de données dans le nuage final.

Dans ce chapitre, nous supposons résolu le problème du recalage et nous nous intéressons à la construction d'une représentation finale d'un objet, par un maillage ou par un ensemble de primitives surfaciques, à partir d'un nuage de points 3D. Pour cela, nous proposons d'adapter le type d'algorithme proposé par Bernardini, Mittleman, Rushmeier, Silva et Taubin, le *Ball Pivoting Algorithm* présenté dans [Bernardini 99] (désigné ci-après par BPA, plutôt que sous la dénomination de l'algorithme de la boule pivotante).

La méthode initiale traite de la construction de maillages triangulaires : notre adaptation de BPA vise à rendre cet algorithme robuste aux erreurs de recalage ou aux bruits sur les mesures, à traiter la segmentation de la surface en même temps que la création du maillage, en profitant du comportement de type "croissance de régions" de cet algorithme.

La figure 3.1 illustre cette étape sur des vues de synthèse : elle présente quatre vues d'une surface (a), qui sont recalées sur (c). (b) montre la superposition des maillages existants pour chaque vue et (d) est le résultat de la reconstruction incrémentale du maillage complet à partir du maillage existant de la vue 1 et les points des autres vues 2, 3 et 4.

Ce chapitre comporte 4 sections : la section 2 présente l'état de l'art des méthodes de construction de maillage. La section 3 concerne la description de la méthode, et la présentation de résultats sur différents types d'images 3D. Nous validons nos méthodes sur les images classiques téléchargées depuis le Web (objets *Bunny*, *curvblock*...), mais surtout sur les images acquises par stéréovision depuis nos robots. Rappelons que notre objectif est la modélisation en ligne, d'objets 3D inconnus a priori, qu'un robot pourrait trouver lors de l'exploration d'un environnement. Les sections 4 et 5 présentent une comparaison des résultats obtenues par BPA et par deux autres méthodes de construction de maillage ; comparaison qualitative en 4, quantitative en 5. Enfin la section 6 nous permettra de conclure et d'évoquer nos travaux actuels.

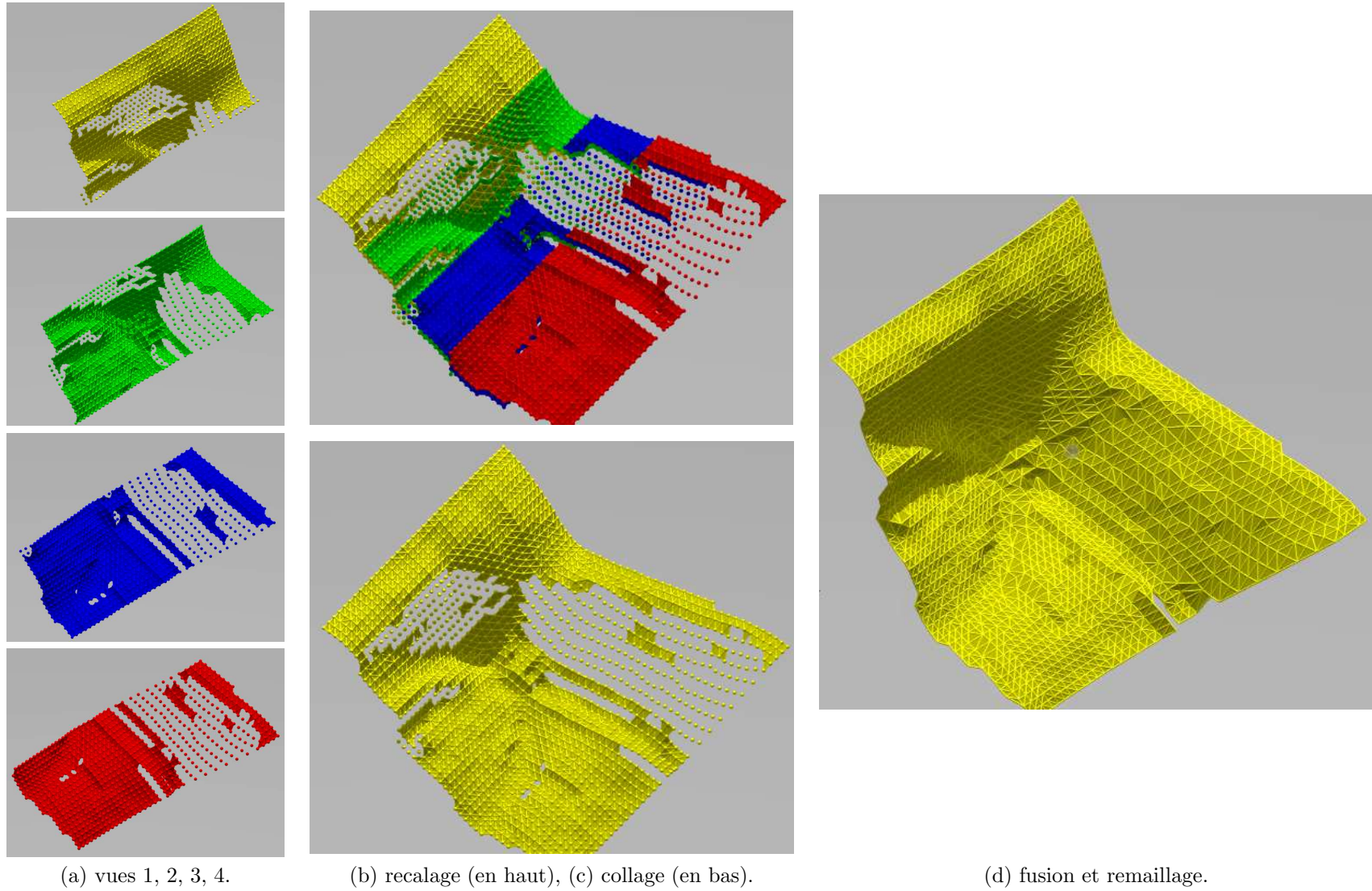


FIG. 3.1 – Exemple de recalage et construction d'une représentation finale, à partir de 4 vues de synthèse.

## 3.2 Etat de l'art.

Le développement récent de systèmes performants d'acquisition de données 3D a motivé la conception de différentes méthodes pour reconstruire un objet. Ces méthodes sont adaptées aux spécifications des applications (type d'objets, résolution, précision... du modèle 3D) et aux propriétés des systèmes d'acquisition (résolution, précision... des données 3D).

Un objet ou un environnement peuvent être représentés de nombreuses manières : les maillages triangulaires, initialement introduits par la communauté CAO ou *Computer Graphics*, se sont imposés du fait de leur généricité (tous les objets sont représentables par des maillages). De très nombreux logiciels sont disponibles, en particulier pour *décimer* des maillages (suppression itérative de sommets ou d'arêtes jusqu'à obtenir la résolution nécessaire pour une application).

Nous décrivons d'abord ci-dessous les techniques de construction incrémentale d'un maillage triangulaire, puis, les nombreuses méthodes de construction de maillage qui traitent globalement un nuage de points, résultat du recalage de nombreuses images :

- la méthode dite des *Marching Cubes* et ses variations fondées sur une discrétisation de la zone de perception [Lorenson 87, McCormick 02, Lee 02].
- les surfaces déformables fondées sur l'application de forces sur un maillage initial, souvent créé sur une sphère par subdivision d'un icosaèdre [Lagarde 04, Kobbelt 99]; ces forces plaquent itérativement le maillage sur les points de mesure.
- enfin les méthodes qui se rapprochent de celle que nous avons développés, l'algorithme BPA.

### 3.2.1 Construction incrémentale de maillage

Les techniques de construction incrémentale d'un maillage triangulaire à partir de points 3D peuvent être classées sommairement en deux grandes catégories :

**méthodes fondées sur un découpage itératif** : elles utilisent surtout la triangulation de Delaunay, ce qui donne une garantie théorique pour la qualité de la maille finale. Les inconvénients sont le besoin de mémoire et le temps de traitement, ainsi que la possibilité d'instabilité numérique.

**méthodes fondées sur une croissance de régions** : l'avantage principal est la robustesse aux erreurs d'acquisition; par contre ces méthodes ont besoin d'un triangle-souche pour commencer à intégrer des nouveaux points perçus à chaque pas du processus.

Plus explicitement, Hoppe et DeRose [Hoppe 92] ont proposé une méthode de découpage, en considérant la fonction de distance entre chaque point et une surface tangente, tandis que Bernardini et Bajaj [Bajaj 95, Bajaj 97] utilisent la fonction de distance à la forme alpha du nuage de points. Attali [Attali 97] fait la reconstruction de la surface en utilisant un sous-ensemble du diagramme de Delaunay. Boissonnat et Cazals [Boissonnat 00] ont appliqué une technique mixte, avec les points et les normales à la surface, en calculant la surface comme un ensemble de zéros d'une fonction, ce qui donne une relation de voisinage naturelle entre points.

[Bozier 00b] traite de la construction incrémentale d'un maillage par fusion des maillages locaux construits à chaque image 3D par les techniques de découpage itératif (*Split*) présentées en 2.4.2. Une des difficultés majeures concernait la détection des parties communes entre le modèle global courant construit avec les images précédentes, et le maillage local construit avec l'image courante, en tenant compte des possibles occultations; une fois isolées, ces parties peuvent être fusionnées (ce qui peut consister à conserver la meilleure des deux représentations), puis reliées au reste par des techniques de *couture* (rajout d'arêtes pour relier deux maillages) [Turk 94]. Cette méthode est trop complexe, et conduit souvent à des erreurs, surtout si les données acquises ont des résolutions variables ou sont très bruitées.

D'autres auteurs (Sappa, Sequeira...) ont proposé aussi des techniques de "couture", mais sans traiter ni le cas général des occultations, ni la fusion des maillages générés dans des zones perçues depuis plusieurs points de vue.

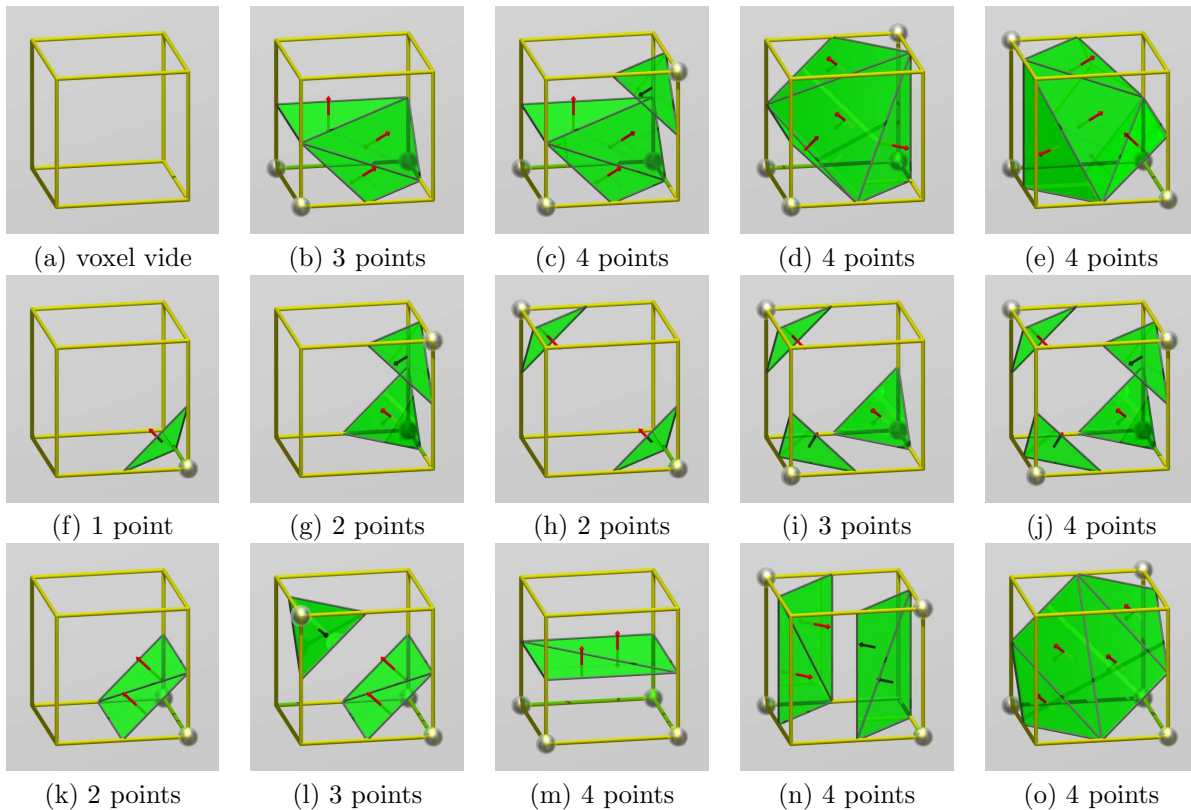


FIG. 3.2 – Combinaisons de base des surfaces pour un voxel de l’algorithme des “Marching Cubes”.

### 3.2.2 Algorithme des “Marching Cubes”.

La méthode de “Marching Cubes” est un algorithme breveté par Cline, Lorensen et Crawford [Lorensen 87], qui génère un maillage à partir de données volumétriques. L’élément de base est le *voxel*, qui est une unité de discrétisation de l’espace. Nous avons déjà utilisé de telles cellules dans les *clusters*, comme structure de données pour optimiser l’accès aux points 3D pour ICP et nous les utiliserons aussi pour BPA.

Dans une première étape, l’algorithme exploite une fonction scalaire, qui assigne en tout point de l’espace, une isovaleur, nulle si le point est sur la surface de l’objet à modéliser, négative si le point est intérieur, positive s’il est extérieur : cette isovaleur est déterminée à partir du nuage de points 3D. Un voxel est caractérisé par ses huit sommets : une isovaleur sera donc associée à chaque sommet, nulle, négative ou positive en fonction de sa position vis-à-vis de la surface.

Dans une deuxième étape, chaque voxel sera analysé : si un ou plusieurs sommets ont des isovaleurs positives et négatives, alors ce voxel doit contribuer à la formation de la surface. En analysant les arêtes intersectées par l’isosurface, sachant quel sommet est à l’intérieur ou à l’extérieur, nous trouvons la combinaison de triangles assignée au voxel. Après avoir connecté chaque combinaison de triangles, voxel par voxel, un maillage triangulaire est généré.

Cet algorithme se fonde donc sur deux idées assez simples :

- la première est la classification des voxels en fonction de la répartition des isovaleurs sur ses sommets : la classe correspond directement à une répartition des triangles sur un voxel ;
- la deuxième est l’union entre voxels : du fait que les voxels adjacents partagent des sommets, la fusion des triangles générés pour chaque voxel, vont former une surface continue.

En théorie, les configurations possibles des sommets d’un voxel, sont au nombre de 256 (puisque un

sommet ne peut avoir que deux étiquettes : - ou + ... le cas 0, donc du sommet exactement sur la surface, n'est pas considéré). Une configuration correspond à une classe de voxels, et donc à un découpage en triangles. Les deux configurations où tous les sommets sont à l'extérieur ou à l'intérieur du volume, ne sont pas intéressantes, car ces voxels ne coupent pas la surface de l'objet. Pour toutes les autres 254 combinaisons, on doit savoir quelle arête du voxel a été intersectée afin de former un triangle ou un groupe de triangles. En tenant compte des symétries, les configurations se réduisent à 14 : elles sont représentées sur la figure 3.2.

Plus exactement,

- si un seul sommet du voxel est négatif (point intérieur à l'objet) (figure 3.2(f)), un triangle avec une normale sortante, est formé avec les intersections de la surface et de chaque arête qui part de ce sommet. Il y a 8 configurations possibles dans ce cas, une par position possible du sommet négatif.
- de manière duale, si un seul sommet est positif, nous obtenons aussi un seul triangle avec la normale de signe inverse, les 7 autres sommets du voxel étant à l'intérieur du volume ; on a aussi 8 configurations possibles selon la position du sommet positif.
- dans le cas de deux sommets qui sont négatifs, il existe 3 configurations possibles et uniques (figures 3.2(g),(h) et (k)).
- pour trois sommets négatifs, il existe aussi 3 configurations uniques (figures 3.2(b),(i) et (l))
- et finalement pour quatre sommets nous en obtenons 7.

Chaque voxel qui a une configuration non triviale, donc qui coupe la surface, contribue au modèle de cette surface, avec de 1 à 4 triangles. Les sommets des triangles sont calculés par interpolation le long des arêtes des voxels.

Avec ces contributions pour chaque voxel, la formation de la surface complète est obtenue par propagation en parcourant l'espace de voxel en voxel, deux voxels voisins partageant les arêtes créées sur leur face commune.

Cet algorithme est très utilisé afin de construire un maillage depuis un nuage de points ; il s'applique aussi à des données surfaciques, comme par exemple dans le cas des données acquises par profilométrie dans le système BODY SCAN, conçu et réalisé dans notre équipe [Lagarde 04]. Il a les avantages et inconvénients suivants :

- il est très performant, puisque un nuage de 20000 points 3D par exemple (ce que nous avons typiquement avec BODY SCAN), peut être maillé en 2 ou 3s.
- la résolution du maillage final, est ajustable, par la résolution de la discrétisation de l'espace. Plus les voxels sont petits (typiquement,  $1\text{cm}^3$  dans le projet BODY SCAN), plus le maillage colle sur la surface de l'objet.
- un inconvénient qui peut s'avérer important selon l'application, est la possibilité d'avoir des trous dans le maillage final. En particulier, si la résolution en voxel est trop fine, des voxels "vides" peuvent s'intercaler entre les points de mesure. De tels problèmes peuvent se résoudre par une approche multi-résolution (en augmentant la taille des voxels connexes des trous), mais avec le risque de perdre la cohérence du maillage final.

### 3.2.3 Surfaces déformables.

Une autre méthode classique de génération d'un maillage, utilisé aussi dans le projet BODY SCAN [Lagarde 04], exploite le concept de surfaces déformables, aussi appelées surfaces actives. Nous avons implémenté cette méthode, en coopération avec P.Le Coutaller [Coutaller 03], lors de son stage de fin d'étude au laboratoire.

L'introduction de ces méthodes par Kass, Witkin et Terzopoulos [Kass 88], concernait initialement la recherche d'un contour dans une image 2D ou 3D, à partir de la minimisation d'un critère d'énergie ; un tel contour "actif" est connue sous le nom de *Snakes*. La méthode a été ensuite reprise et développée dans le domaine tridimensionnel à l'aide du principe des surfaces actives, déformées sous l'action conjuguée de forces internes, résultant de la pression par exemple, et de forces externes (déplacement) dépendantes du

nuage de points.

Le principe consiste à faire converger un maillage sur la surface, grâce à différentes caractéristiques de position, d'élasticité, d'éloignement... entre sommets du maillage et nuage de points acquis sur cette surface. La principale contrainte d'évolution sera la distance euclidienne des sommets aux points de mesure.

En partant d'un maillage initial plus ou moins résolu, la méthode consiste donc à déformer par itérations successives, ce maillage à l'aide d'un ensemble de forces afin de le faire converger vers le nuage de points, jusqu'à un état d'équilibre stable. En général, le maillage initial est défini sur une sphère discrétisée placée soit à l'intérieur de l'objet à modéliser comme un ballon gonflable, soit au contraire, englobant le nuage de points 3D (comme un sac dans lequel en fait le vide)

Pour déformer ce maillage, nous utilisons un système dynamique masse-ressort inspiré des travaux de J-O. Lachaud et de H. Delingette [Lachaud 98] [Delingette 94, Delingette 97] : une équation de mouvement newtonien est assignée à chaque sommet du maillage, qui de ce fait, va évoluer sous l'action de forces internes et externes. Les forces internes dépendent de la géométrie du maillage et garantissent la cohésion et la régularité du maillage final. Les forces externes dépendent du nuage et attirent le maillage vers celui-ci.

Les critères d'arrêt ou de convergence, peuvent être complexes. De manière simpliste, le calcul itératif des positions des sommets doit s'arrêter quand la stabilisation du maillage est obtenue entre deux instants  $t$  et  $t + 1$ . En fait, comme le maillage peut aussi être soumis à des oscillations dues à des configurations particulières de la surface, un autre critère d'arrêt généralement utilisé est le nombre d'itérations.

Les figures 3.16(b), 3.17(b), 3.18(b), 3.20(b), 3.21(b)(c) présentent des maillages obtenus après convergence d'une surface déformable sphérique. Pour les surfaces sans repliements comme illustré par exemple sur les figures 3.16 et 3.17, le maillage déformable épouse correctement la forme du nuage des points étudiés.

Par contre, sur les régions accidentées de la surface, par exemple un motif replié ou une ouverture circulaire comme sur les figures 3.18 et 3.21, les résultats sont médiocres ; sans rajouter des traitements spécifiques, la méthode générale de convergence du maillage, ne permet pas de représenter les zones fortement concaves. Un maillage de départ plus proche du nuage de points nous permettrait d'avoir une convergence plus précise dans ces zones.

Rappelons quelques avantages et inconvénients de cette méthode :

- par définition, le maillage final est fermé ; sur des objets simples, c'est un avantage, mais si la scène à représenter comporte plusieurs objets disjoints, des techniques spécifiques doivent être rajoutées pour scinder le maillage initial en plusieurs composantes : il faudra supprimer les arêtes dont les sommets appartiennent à des composantes différentes.
- généralement, les temps de convergence sont importants : initialisation du maillage initial par discrétisation d'une sphère, déformation du maillage, détection de la convergence ...
- la méthode possède de très nombreux paramètres qu'il est parfois difficile de régler, selon la topologie de la surface à modéliser.
- la méthode peut être multi-résolution, si les sommets sont rajoutés en certain endroits (fortes variations de courbure) en cours de convergence.

### 3.2.4 Méthodes mixtes : maillage et primitives surfaciques

Le travail de Wheeler [Wheeler 98] est une solution basée sur des représentations surfaciques avec un volume implicite en utilisant un "octree" pour trouver une surface à partir de plusieurs vues. Ce type d'approximation est très robuste aux erreurs de recalage.

Les travaux suivants ont fait des choix similaires à la méthode présentée dans ce chapitre. Fisher, Fitzgibbon et Eggert [Fisher 97] construisent une représentation maillée ("one tessellation") à partir du nuage de points recalés selon la méthode de Hoppe et DeRose, puis effectuent une classification de la forme géométrique locale de l'objet à modéliser afin d'adapter la construction du modèle par croissance

de régions.

Enfin, Bernardini, Mittleman, Rushmeier, Silva et Taubin [Bernardini 99] ont développé la méthode de triangulation BPA, pour traiter aussi la problématique de la reconstruction et de la fusion des images 3D acquises depuis plusieurs points de vue. C'est une méthode itérative fondée sur les formes alpha, qui permet de fusionner dans un maillage des données préalablement recalées. C'est cette méthode que nous avons adaptée, car elle peut manipuler beaucoup de points 3D avec peu de coût de calcul.

Nous avons introduit quelques adaptations en complément des autres caractéristiques :

- traiter les images 3D de manière incrémentale, donc pouvoir fusionner avec le maillage en cours de construction, chaque image dès qu'elle est acquise et recalée. Cette capacité permet une stratégie flexible de modélisation, puisque BPA est applicable soit de manière globale sur un nuage de points, soit de manière incrémentale.

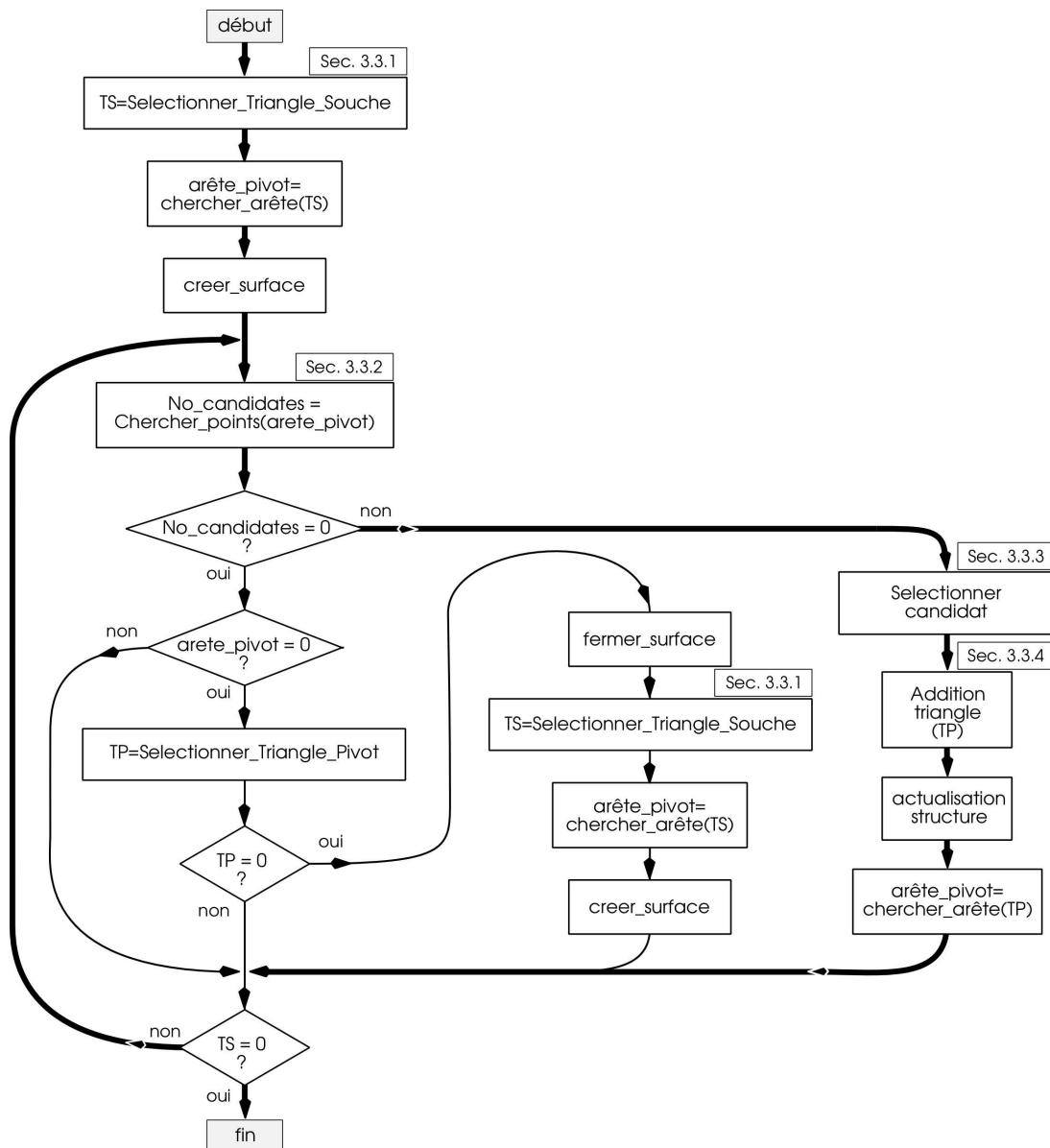


FIG. 3.3 – Diagramme de l'algorithme de la boule pivotante. La ligne épaisse montre le chemin de croissance d'une surface.



- segmenter simultanément le modèle en un ensemble de primitives surfaciques (ici des facettes planes).
- affinement du maillage courant par exécution de l'algorithme BPA avec de nouveaux paramètres (différents rayons de la boule), ce qui permet de fermer les trous ou de mieux décrire une partie du modèle si besoin.

### 3.3 Description de l'algorithme BPA.

Le principe de l'algorithme BPA est très simple et à notre sens, élégant : le processus commence à partir d'un triangle-*souche*, dont les trois points appartiennent à une sphère de rayon  $\rho$  choisie par l'utilisateur (figure 3.4). Cette boule pivote autour d'une des arêtes du triangle, en gardant toujours le contact avec les deux sommets de cette arête pivot, jusqu'à ce que :

- la boule rencontre un autre point de mesure, utilisé pour former un nouveau triangle (figure 3.5).
- la boule ne rencontre pas de point de mesure ; alors, cette arête restera libre, et fera partie de la frontière de la surface.

Le mécanisme s'exécute depuis une arête du nouveau triangle ou depuis une arête non exploitée d'un autre triangle créé précédemment. Si toutes ces arêtes ont été exploitées, un nouveau triangle-*souche* doit être sélectionné parmi les points de mesure non utilisés et le processus peut recommencer. Cette procédure s'arrête quand toutes les arêtes ont été exploitées et quand tous les points ont été utilisés.

La version originale de la méthode BPA proposée dans [Bernardini 99], est très sensible aux bruits sur les points 3D, à la résolution variable des images et aux erreurs de recalage. Or notre capteur privilégié en robotique mobile, est la stéréovision ; nous avons dit combien les données 3D stéréo étaient imprécises et de résolution variable. Nous devons donc adapter l'algorithme BPA pour le rendre plus robuste. Dans les sous-sections suivantes, nous allons décrire chaque étape, et présenter nos adaptations et améliorations.

Nous donnons quelques définitions au préalable :

**surface** : notre méthode génère un ensemble de maillages disjoints, appelés surface ou portion de surface. Si l'option *segmentation* est activée, une surface correspond à une primitive surfacique. Dans la version actuelle, les seules surfaces détectées sont des plans, mais des extensions sont possibles pour extraire des primitives surfaciques spécifiques [quadriques] telles que cylindres et sphères.

**triangle-souche** : c'est le triangle à partir duquel le maillage d'une surface est généré.

**triangle-pivot** : la construction du maillage pour une surface, commence par le triangle-*souche*. Une fois créé un nouveau triangle, le BPA se poursuit en sélectionnant le triangle à partir duquel le maillage va se développer. C'est le triangle-pivot.

**arête-pivot** : une fois le triangle-pivot choisi, il faut encore sélectionner une arête de ce triangle autour de laquelle la boule va pivoter. C'est l'arête-pivot.

**sommets-pivot** : sont les deux sommets de l'arête-pivot. Le nouveau triangle est formé avec ces deux sommets et le point à trouver.

**arête frontière** : c'est une arête qui n'a qu'un triangle adjacent, donc qui est à la frontière de la surface. Les autres arêtes, internes à la surface, sont dite "occupées".

**triangle frontière** : c'est une triangle qui contient une arête frontière, qui est donc en bordure d'une composante connexe du maillage.

Dans sa version de base, sans l'option *segmentation*, le processus de triangulation à partir d'un nuage de points 3D, démarre à partir du triangle-*souche* et va continuer jusqu'à ce que l'ensemble des points intégrables dans cette portion de surface soit épuisé : cela surviendra quand aucun point de mesure non encore exploité, n'est atteint par la boule depuis toutes les arêtes pivot non exploitées. Ces arêtes deviennent de ce fait des arêtes frontières de la surface.

L'algorithme est conçu pour croître autour du triangle-*souche* ; s'il reste des points libres, un nouveau triangle-*souche* devra donc être sélectionné et une deuxième portion de surface sera créée, et ainsi de suite jusqu'à exploiter tous les points libres. La figure 3.9 illustre la création d'une série de triangles de la surface.

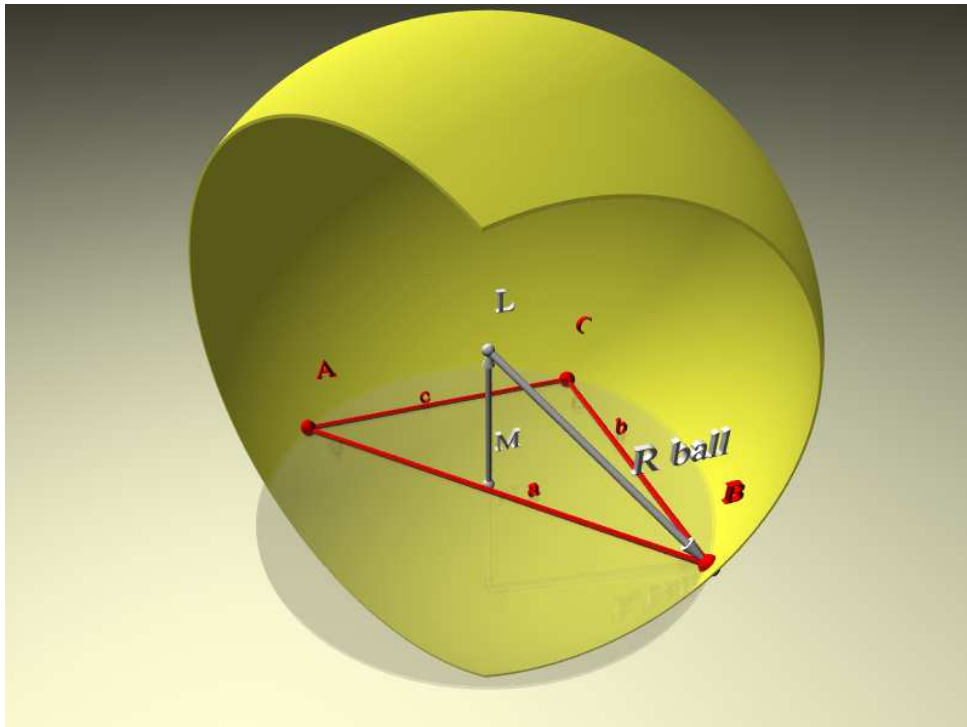


FIG. 3.4 – La boule pivotante de rayon  $\rho$  et de centre  $L$ , repose sur un triangle  $ABC$ .  $M$  (sur le plan  $ABC$ ) est le centre du cercle circonscrit à  $ABC$ .

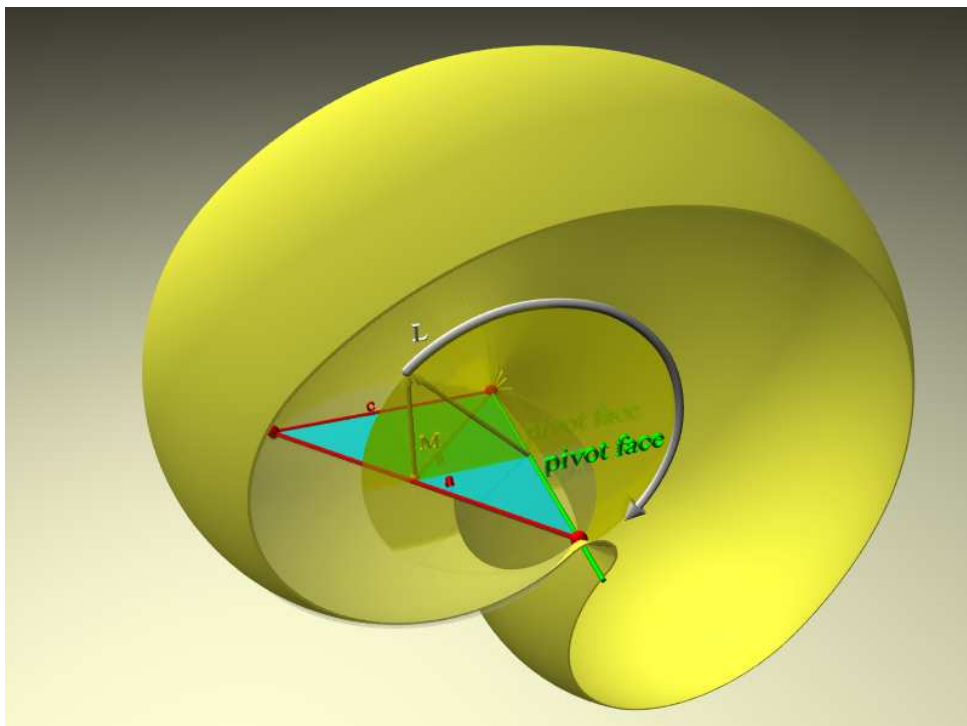


FIG. 3.5 – La boule pivote autour de l'axe  $BC$ , qui est désigné comme l'arête pivot.

Si l'option *segmentation* est activée, le fonctionnement est similaire, sauf qu'un critère d'homogénéité est vérifié à chaque intégration d'un triangle dans une surface, ici un critère de planéité. La construction d'une surface peut donc s'arrêter, soit comme dans le cas général, parce qu'aucun point n'est atteint par la boule, soit parce que les points atteints ne permettent pas de construire des triangles compatibles avec la surface en formation.

Nous détaillons les opérations essentielles de notre méthode BPA : choix du triangle-souche, sélection des points candidats, choix du meilleur point, choix de l'arête-pivot...

L'algorithme complet est montré sur la figure 3.3.

### 3.3.1 Initialisation : choix du triangle-souche

Une étape importante du processus est de trouver un bon *triangle-souche* pour commencer la méthode, une fois que le rayon de la boule pivotante est donné. Le choix de ce rayon, doit être adapté à la résolution du nuage de points à mailler, par ailleurs, ce rayon prédéterminera la qualité du modèle construit :

- si le rayon est petit, la boule ne pourra pas inclure le triangle-souche, ou ne rencontrera pas de points de mesure en pivotant sur de nombreuses arêtes, qui de ce fait, deviendront des frontières de la surface ; donc, le modèle final sera non continu, et comportera de nombreux trous.
- si le rayon est trop grand, la boule rencontrera de nombreux points ; un seul sera choisi pour construire un nouveau triangle, et les autres seront invalidés (ne seront plus libres) ; le modèle final sera continu, mais n'épousera pas complètement la forme du nuage de points.

La condition majeure à remplir par le triangle-souche, est que la boule doit contenir le triangle ; de plus il doit satisfaire quelques contraintes de surface et d'angle prises en considération pour assurer la bonne géométrie du triangle (similaire à un triangle équilatéral). Selon les modalités d'utilisation de l'algorithme, il existe plusieurs options :

**Construction globale à partir d'un nuage de points :** le premier triplet de points qui respecte les conditions de rayon et de forme est pris pour former un triangle-souche.

**Construction de maillage à partir des maillages locaux :** si pour chaque vue, un maillage local est conservé , éventuellement simplifié par décimation, le triangle-souche est cherché seulement parmi les triangles de ces maillages locaux. Cette alternative est rapide parce que nous n'avons pas besoin de parcourir les triplets de points, mais uniquement les triangles existants. Le maillage ancien va être remplacé, sauf la nouvelle figure à former.

**Affinement de maillage ou construction incrémentale :** dans ce cas, le triangle-souche sera cherché dans les triangles frontière du maillage existant. Pour être sûr de trouver un triangle-souche, la valeur minimale du rayon est automatiquement sélectionnée en fonction des dimensions des triangles frontière.

Normalement, un algorithme de construction de maillage, tel que *Marching Cubes* ne passe qu'une fois sur un nuage de points. Avec nos adaptations, le processus de maillage avec la boule pivotante peut effectuer plusieurs passages sur le maillage à construire, par exemple :

- un premier passage permet de créer le maillage ou de le compléter à partir des points d'un nuage ou d'une nouvelle image 3D,
- les passages suivants prennent en entrée le maillage construit et le nuage de points afin d'affiner l'ensemble, ou seulement une composante connexe.

### 3.3.2 Recherche des points candidats.

A partir d'une arête-pivot, une fonction de base consiste à trouver dans le nuage de points, les points candidats pour être sommet du prochain triangle. Comme dans la section 2.5, la routine de balayage de points exploite une organisation de l'espace par *clusters* pour réduire le temps de calcul ; seuls les *clusters* localisés autour du triangle-pivot sont analysés. Cette méthode présente la complexité arithmétique indépendante du nombre croissant de triangles et du nombre de points dans le nuage. Une comparaison entre un processus avec l'utilisation de *clusters* et un autre sans *clusters* apparaît sur la figure 3.6. La

résolution de cette structure doit être choisie pour optimiser la vitesse de l'algorithme, sans pour autant oublier des points candidats probables ; cette résolution est liée au rayon  $\rho$ . L'exploitation des *clusters* divise le temps nécessaire pour la recherche des points candidats, d'un facteur 10 environ.

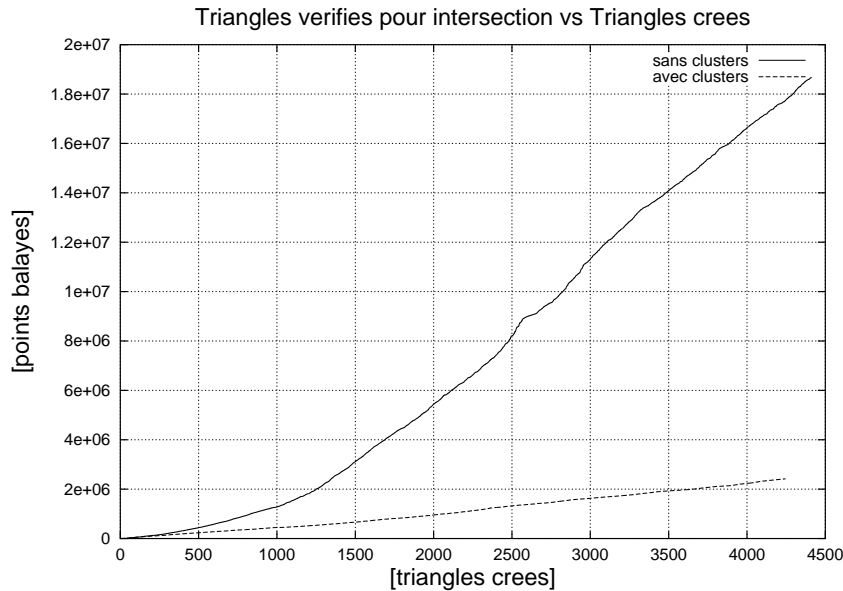


FIG. 3.6 – Impact de l'utilisation de clusters, en analysant le nombre de points analysés en fonction du nombre de triangles créés.

La sélection des points candidats est faite étape par étape, en exploitant divers critères ; des résultats intermédiaires de cette sélection sont conservés, ce qui est utile pour la recherche du triangle-pivot suivant. Les étapes de cette sélection sont :

1. d'abord sont sélectionnés les points qui sont à l'intérieur de la trajectoire de la boule autour de l'arête-pivot ;
2. ensuite, pour chaque point candidat, sont calculés la normale, la surface et les angles du triangle candidat, construit entre ce point et l'arête-pivot. Seuls sont conservés les points pour lesquels ces paramètres satisfont certains tests. En particulier, si la segmentation est activée, la normale moyenne de la surface actuelle est employée pour choisir un point qui reste sur la surface.
3. enfin, pour chaque point candidat qui satisfait ses critères, le triangle candidat ne doit pas intersecter les autres triangles existants. Ce test d'intersection est vital pour la qualité du maillage résultat, pour éviter des problèmes de topologie créés par les triangles qui passent au-dessus ou au-dessous du triangle candidat. C'est la procédure de calcul géométrique la plus complexe de l'algorithme.

A la fin de cette sélection pour une arête-pivot, nous avons un ensemble de points candidats, qui sont soit points libres (non encore exploités), soit points frontières (sommets d'arêtes frontières). La fonction sélectionne finalement deux points candidats : le meilleur point libre, et s'il y en a, le meilleur point frontière.

La sélection de ces meilleurs points candidats dépend d'un test de qualité sur le triangle qui résulterait de ce choix : un triangle est meilleur s'il se rapproche d'un triangle équilatéral, s'il a une surface proche de celle du triangle-pivot courant, et si sa normale est proche de la normale moyenne de la surface. L'algorithme de cette partie est montré sur la figure 3.7.

### 3.3.3 Sélection du point candidat.

Dans la plupart des cas, le triangle-pivot a deux arêtes frontières et une occupée, cette dernière est l'arête-pivot de l'itération précédente, qui fait le contact avec la surface qui est en train de se former ;

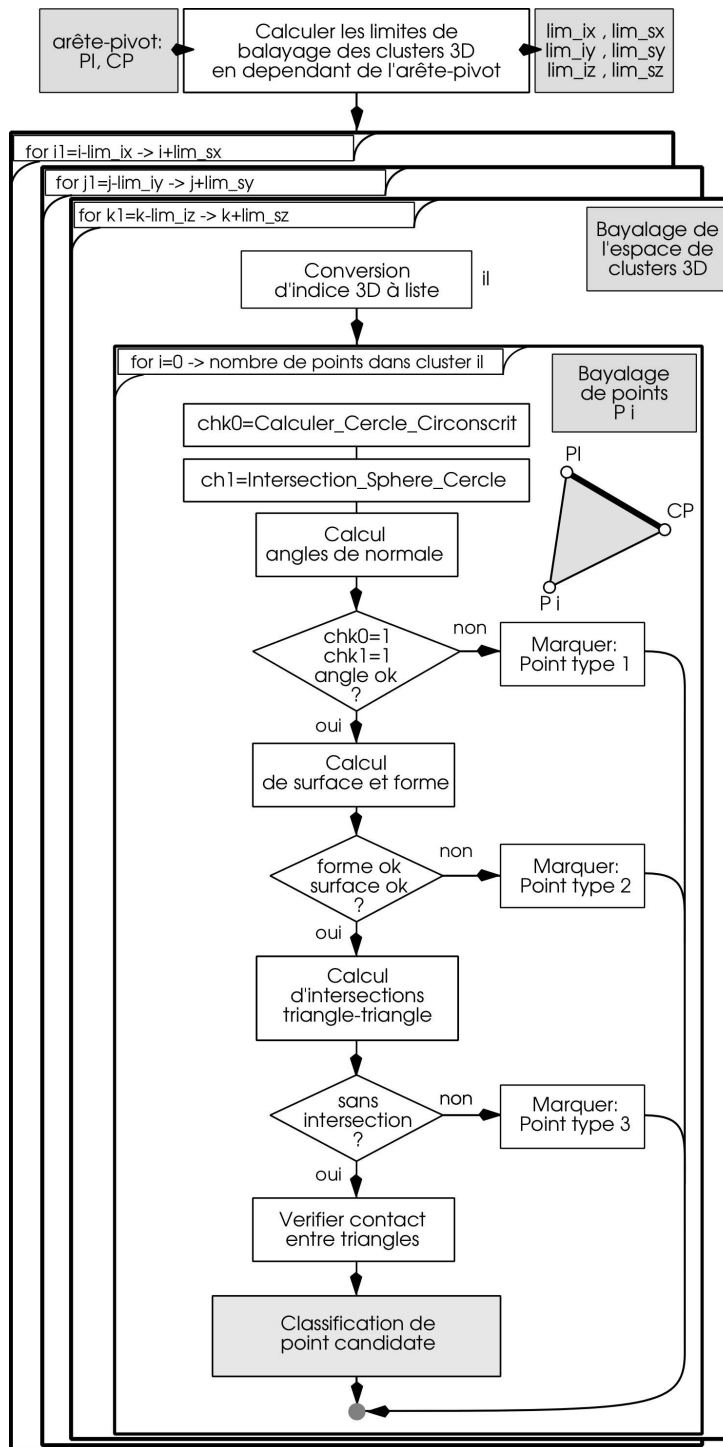


FIG. 3.7 – Diagramme de l'algorithme de recherche de points pour la création d'un triangle décrit à la section 3.3.2.

quand il existe deux arêtes frontières, les points candidats pour la création d'un nouveau triangle sont scrutés sur ces deux arêtes.

Comme vu sur la figure 3.8, l'algorithme commence à scruter l'arête orientée de la même manière que lors de l'itération précédente, à droite ou à gauche de l'arête-pivot précédente. Après examen des points

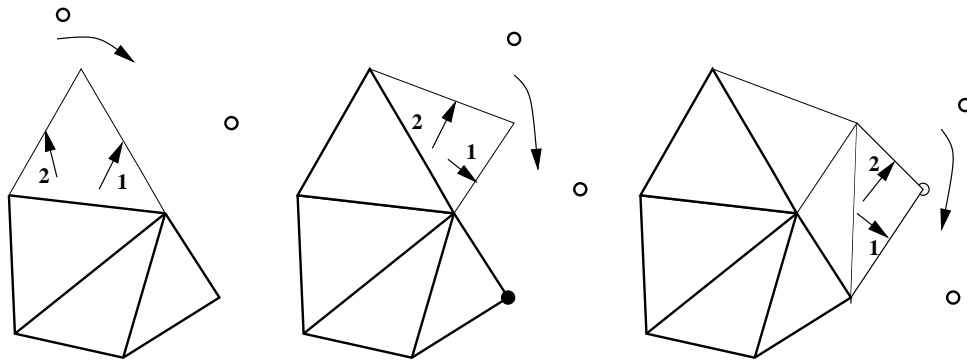


FIG. 3.8 – Progression du maillage suivant une évolution en spirale.

candidats pour l'arête 1, s'il y a une deuxième arête disponible, elle sera scrutée seulement s'il n'y a pas de candidats trouvés pour l'arête 1, ou s'il n'y a pas de points frontière parmi ces candidats. Lorsque la deuxième arête doit être analysée, la sélection entre les deux arêtes dépend de la détection des points frontière sur chaque arête.

Dans le pire des cas, pour chaque arête, la sélection des points représentatifs donnera deux points-candidats : un point libre et un point frontière. Donc, pour un triangle-pivot "normal" avec deux arêtes frontières, nous aurons une paire de candidats judicieux, 4 points au total : 2 libres et 2 points frontière, qui sont ensuite comparés, pour choisir le meilleur point-candidat de tous et par conséquent révèle finalement quelle arête a servi de pivot.

Ce traitement accroît la complexité, mais il provoque une croissance de la partie maillée de la surface, en forme de spirale autour du triangle-souche. Ce comportement permet de minimiser la création de "trous". Comme vu sur la figure 3.8, le point frontière a généralement la préférence dans la sélection, même si le triangle construit est moins représentatif à partir du critère de qualité.

### 3.3.4 Création d'un nouveau triangle.

Après la sélection du meilleur point pour créer le triangle suivant, il faut faire quelques procédures avant d'itérer. D'abord, un nouveau triangle liant l'arête pivot et le point candidat sélectionné, doit être ajouté à la surface en cours de construction ; la structure doit être mise à jour, notamment la liste des arêtes frontières de la surface.

Ce nouveau triangle deviendra en général le triangle-pivot de calcul, mais dans certains cas, un triangle-pivot supplémentaire devra être choisi. Ceci dépend des tests suivants :

**changement d'arête-pivot** : il est possible que dans ce nouveau triangle, on ne puisse pas utiliser la même arête-pivot que lors de la dernière étape. Si c'est le cas, il faut en changer ; ce cas survient surtout au démarrage du maillage d'une surface, dans le choix des arêtes-pivot sur le triangle-souche.

**changement de triangle-pivot** : il est possible que le nouveau triangle trouvé n'ait aucune arête frontière. Un nouveau triangle-pivot dans la même surface sera sélectionné, dans la liste des triangles déjà créés pour cette surface, liste qui commence par le triangle-souche. Le résultat sera le premier triangle qui possède encore une arête frontière non analysée.

**changement de surface** : si sur la surface courante, il n'y a plus d'arête frontière non analysée, alors le processus est fini pour cette surface. Un autre triangle-souche devra être trouvé parmi les points encore libres.

Finalement, la condition d'arrêt de la construction du maillage survient quand il n'y a plus de points libres pour être considérés comme candidats de triangle ou pour former des triangles-souche.

### 3.3.5 Exemple d'exécution de l'algorithme.

La construction étape par étape d'une surface triangulaire est montré sur la figure 3.9 :

- (A-1) : Le triangle-souche et points candidats trouvés par rotation de la boule sur une arête-pivot, le point sombre est le point choisi.
- (A-2) à (B-2) : La boule bouge de triangle en triangle toujours en utilisant la même référence d'arête; un sommet pivot est fixe.
- (B-3) : Le triangle trouvé a fermé le cycle et maintenant une arête-pivot libre sur le triangle présent doit être cherché, si il n'y a pas, un autre triangle de pivot doit être choisi.
- (B-4) à (C-2), (C-3) à (D-1) et (D-2) à (D-3) Cycles suivants avec un autre sommet pivot fixe.

### 3.3.6 Evaluation de l'algorithme BPA.

La méthode de BPA a quelques particularités utiles pour notre but ; nous allons les mettre en évidence à l'aide des surfaces synthétiques polygonalisées avec points régulièrement distribués pour évaluer le comportement classique de l'algorithme BPA. Les erreurs de mesure et de recalage ont été simulées en faisant chevaucher plusieurs vues. Chaque fois le BPA utilise dans ces exemples les maillages locaux des différentes vues pour la recherche du triangle-souche.

#### Robustesse aux erreurs de mesure.

Sur la figure 3.10, nous présentons trois vues sur une même surface synthétique régulière ; ces vues se chevauchent légèrement une sur l'autre pour simuler l'erreur de mesure ou d'ICP. Le BPA a produit un maillage régulier en prenant les meilleurs points de chaque groupe local, points qui remplissent les conditions sur la surface des triangles et sur leur qualité géométrique : une seule surface est générée en sortie.

#### Segmentation des surfaces planes.

La figure 3.11 présente un cas plus complexe ; le nuage de points contient six vues sur deux plans orthogonaux. Les vues se chevauchent aussi légèrement deux à deux, et une paire intersecte les quatre autres surfaces pour évaluer les propriétés de segmentation du BPA. Le résultat est encore très satisfaisant, puisque seules deux surfaces sont détectées. L'intersection, soulignée sur la figure 3.12, n'a pas obstrué la construction du maillage.

### 3.3.7 Modalités d'exécution de BPA : construction incrémentale, affinement de maillage.

L'algorithme a été modifiée d'une manière telle qu'il soit possible d'utiliser un maillage existant pour continuer le processus de triangulation. C'est le cas lorsque les vues sont intégrées de manière incrémentale.

Ceci peut aussi être utile pour créer un maillage avec des résolutions variables, c'est-à-dire un maillage fait de morceaux créés à partir de boules ayant des rayons différents. Ceci permet de résoudre quelques problèmes intrinsèques à la philosophie de l'algorithme, notamment sa tendance à créer des ouvertures. En particulier, nous appliquons l'algorithme BPA de manière itérative à partir d'un maillage existant, en sélectionnant une arête frontière comme axe de pivot et en utilisant un rayon de boule plus grand pour "attraper" un sommet d'un triangle déjà créé ou un point encore libre, afin de fermer une ouverture. Un exemple de ce cas est montré à la section 3.5.

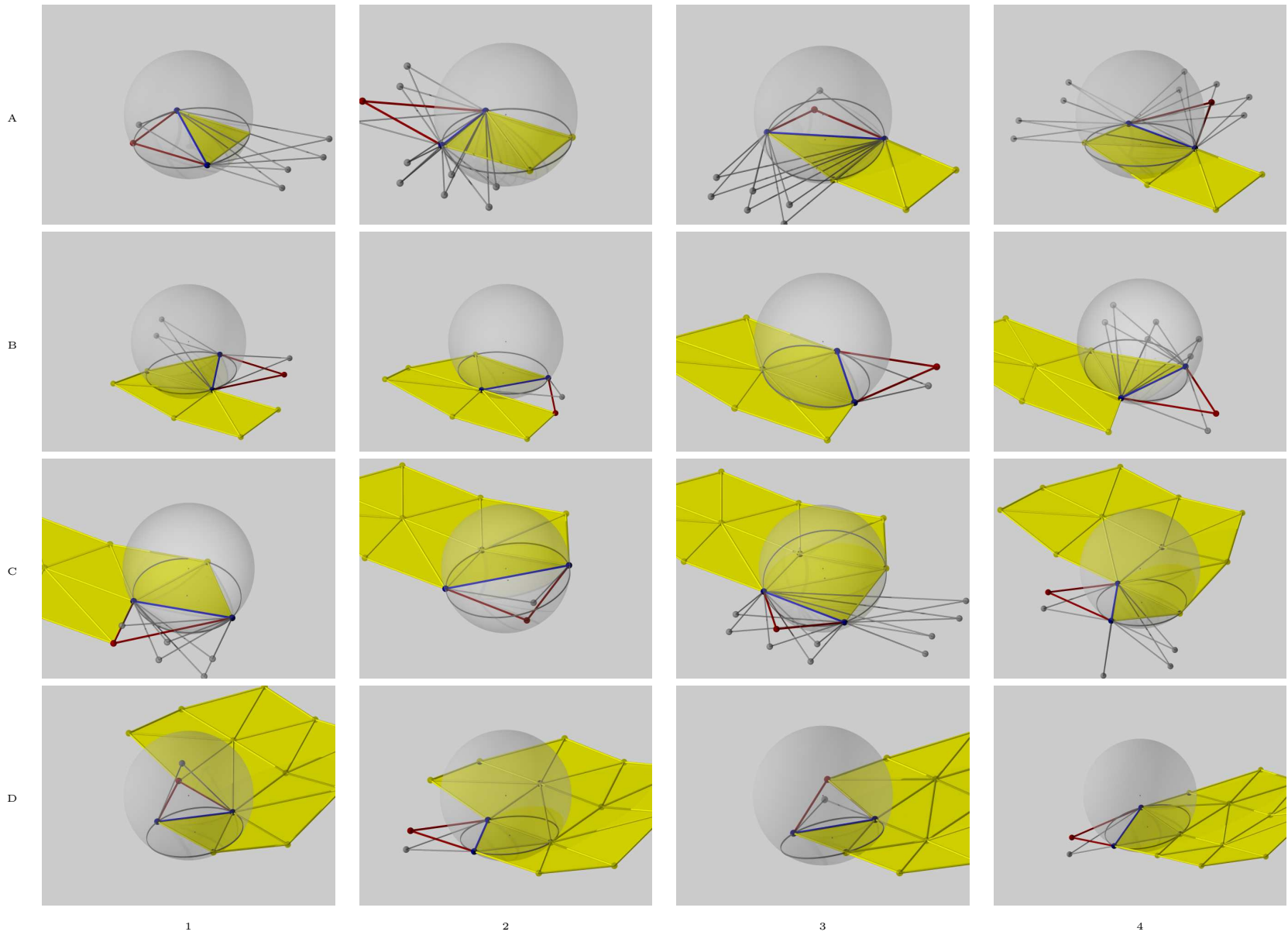


FIG. 3.9 – La construction de la surface triangulaire étape par étape décrite dans la section 3.3.5.



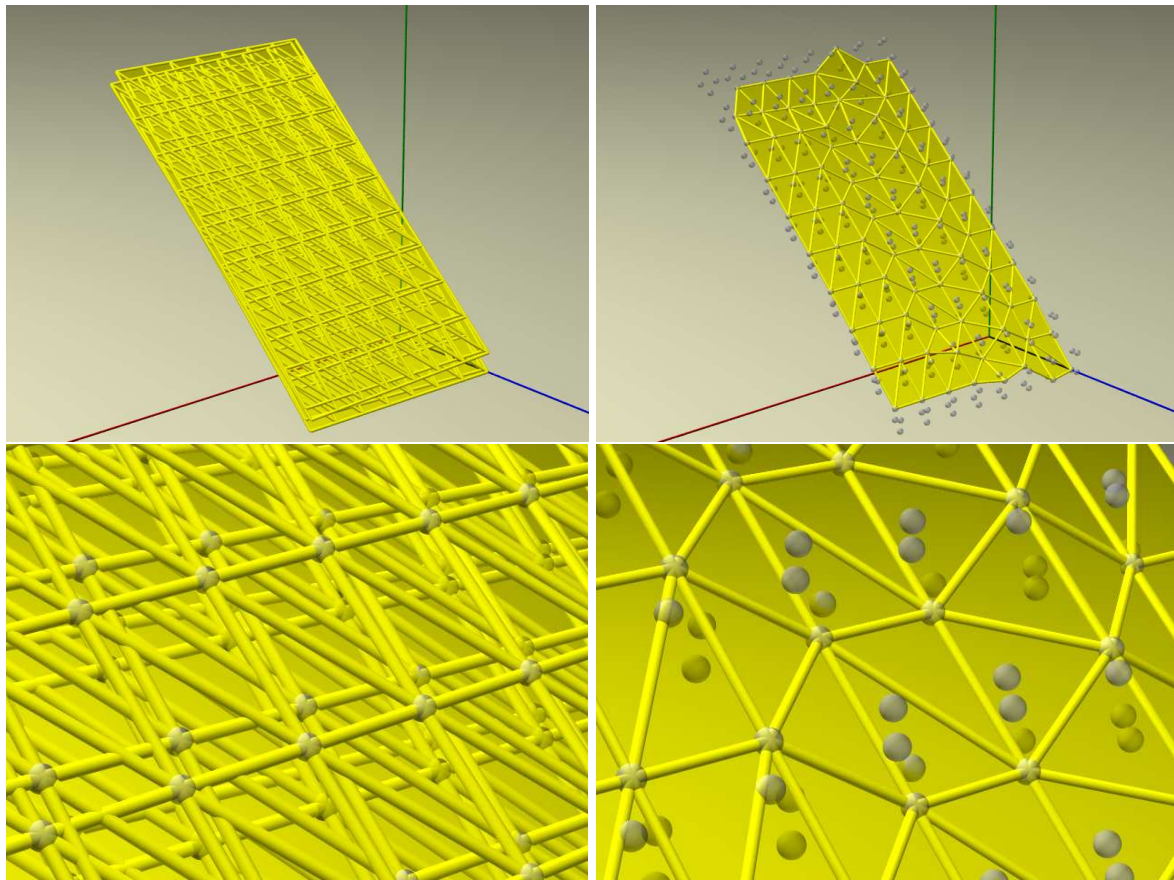


FIG. 3.10 – Résultat de BPA avec segmentation : (gauche en haut) nuage synthétique constitué de deux vues (droite en haut) un seul plan . En bas, des détails sont montrés.

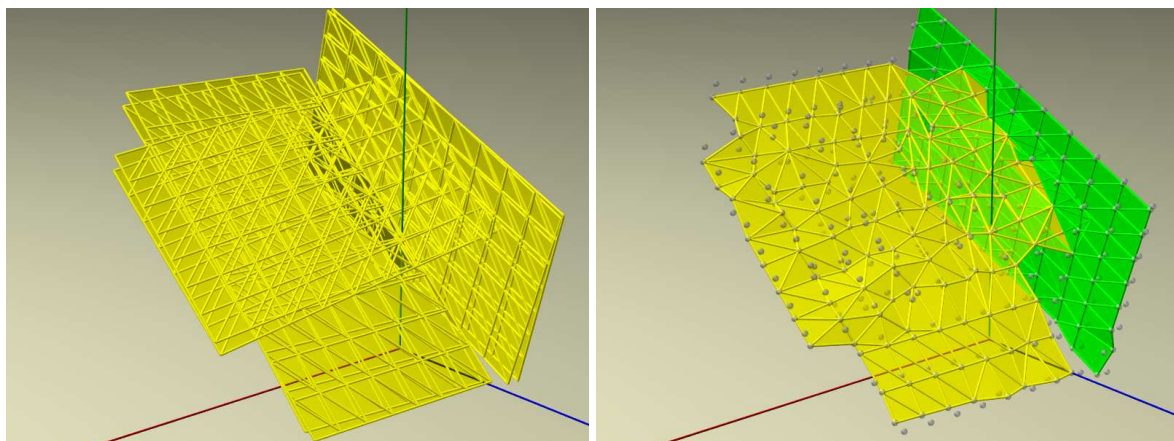


FIG. 3.11 – Résultat de BPA avec segmentation : (gauche), nuage synthétique constitué de six vues, (droite) deux plans orthogonaux

En conclusion les modalités d'opération de l'algorithme, applicable pour différentes situations en fonction du contexte, sont :

- Une construction globale du maillage avec un rayon de boule constante.
- Une construction globale du maillage avec un rayon de boule variable et plusieurs passages d'affi-

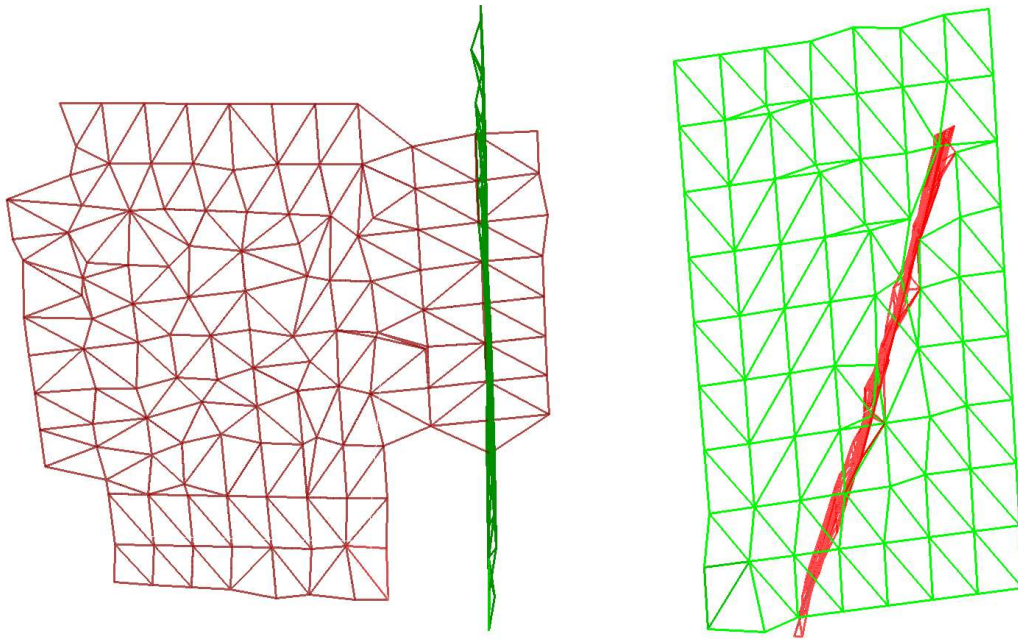


FIG. 3.12 – Vues détaillées du résultat de la figure 3.11 : seuls deux plans ont été détectés malgré les erreurs de recalage.

nement pour suivre des nuages de points à densité irrégulière, ou pour décrire une zone avec plus de détail, ou pour boucher des trous.

- Une construction incrémentale du maillage, en intégrant une nouvelle image au maillage courant.
- Une extraction (incrémentale ou globale) de primitives surfaciques dans le même processus que la construction du maillage.

### 3.4 Comparaison globale entre les méthodes de maillage.

Vous pouvez vous demander pourquoi utiliser la méthode BPA *Ball Pivoting*. En particulier, pourquoi ne pas se contenter d'une méthode traditionnelle et plus connue comme les "Marching cubes" ou un processus basée sur la méthode de "Delaunay" ?

Les points forts de l'algorithme de la boule pivotante sont la possibilité de sélectionner les points à connecter selon quelques règles grâce à la similitude avec les formes "alpha". De façon différente, ce processus n'est pas possible en utilisant la méthode de Delaunay, car une telle règle peut violer les règles de formation des triangles. Par ailleurs, le caractère de construction progressive d'un maillage à partir d'une souche, s'adapte à notre idée de faire la segmentation en même temps que le maillage d'une zone désirée. De plus, la possibilité de croissance autour des bords d'une surface déjà établie, nous donne l'opportunité de réutiliser des surfaces triangulaires déjà générées ou itérées plusieurs fois avec différents paramètres jusqu'à décrire toute l'image. Dans ce cas, si nous utilisons la méthode de Delaunay ou bien la technique de "Marching cubes" nous serions obligés de mailler entièrement l'image.

Au delà de ces avantages dans la stratégie du processus de maillage, nous présentons dans la table 3.1, quelques points de comparaison qualitatif entre les algorithmes BPA, *Marching Cubes* et Surfaces Déformables. Les concepts de ces méthodes sont différents ; les résultats dépendent de l'application pour lesquels elles ont été conçues.

Caractéristique	Boule pivotante	Surface déformable	<i>Marching Cubes</i>
Orientation de la méthode	Surface	Volume	Volume
Fonctionnement basique	Croissance à partir d'un triangle souche avec ou sans comparaison de la normale	Itératif à partir d'un maillage initial ; déformations contrôlées par des forces internes et externes	Formation d'un maillage sur un <i>voxel map</i> : pour chaque voxel, configuration systématique après classification
Structure des données	Voxels	Kdtrees (dans notre implémentation)	Voxels
Sélection des sommets	points du nuage	points de la surface déformable	points interpolés sur les arêtes des voxels
Données d'entrée	Très flexible : Nuage de points ou Maillage ou Fusion de plusieurs maillages ...	Nuage de points et maillage initial englobant ou englobé	Nuage de points avec normales, pour savoir où est l'intérieur et l'extérieur
Résolution du maillage	Dépend du rayon de la sphère	Fixé sommets dans le maillage initial	Dépend de la taille des voxels
Complexité	Dépend du rayon de la boule : plus de calculs avec de grands triangles	Dépend de la résolution du maillage initial	Rapide ; dépend du nombre de voxels.
Affinement	Possible avec croissance par les bords, fermeture des trous...	Possible, mais seulement avec un maillage fermé	Non possible
Création de trous	Susceptible	Maillage fermé garanti	Susceptible
Concavités	Aucun problème : rentre dans les concavités	Concavités non traitées	Concavités traitées, selon la résolution des voxels
Surfaces irrégulières	Susceptible	Maillage souple, selon le nombre de triangles initiaux, les paramètres...	Susceptible
Détails	Capture des détails et contours par densité des points contrôlable	Contours et détails non contrôlables	Densité uniforme quelquesoit le nuage de points
Type de surfaces	Obtention de surfaces ouvertes ou fermés en fonction des données, du rayon de la sphère...	Obtention d'un maillage fermé uniquement	Obtention de surfaces ouvertes ou fermés, mais l'information d'intérieur et extérieur est nécessaire

TAB. 3.1 – Comparaison des techniques de maillage.

## 3.5 Comparaison quantitative des méthodes *Ball Pivoting* et *Surfaces Déformables*.

Dans cette section nous comparons quantitativement, les résultats des maillages obtenus avec notre algorithme de la boule pivotante et la méthode des surfaces déformables développée initialement pour l'application à la profilométrie, avec P.LeCoutaller. Nous présentons d'abord les critères d'évaluation, avant de donner les résultats sur un ensemble significatif d'images.

### 3.5.1 Test de qualité de maillage.

Afin de comparer les deux méthodes de construction d'un maillage nous avons utilisé deux critères de qualité pour la répartition spatiale des sommets du maillage résultat :

- l'idéal serait d'avoir des triangles équilatéraux; un critère évalue la différence moyenne entre les triangles du maillage et un triangle équilatéral.
- généralement, on souhaite obtenir une surface lisse, ou présentant de faibles variations locales de courbure; un second critère évalue donc la moyenne, pour tous les triangles, de la différence d'orientation entre la normale et les normales des triangles voisins. C'est une valeur de planarité locale.

Nous proposons le critère de qualité pour les triangles souvent utilisé dans le calcul des éléments finis [Frey 97].

$$\mathbf{Q}_{equ} = \alpha \frac{\rho}{h} = \alpha \frac{A}{s \cdot h} \quad (3.1)$$

où  $\alpha$  est le coefficient de normalisation,  $\rho$  est le rayon intérieur,  $s$  le demi-périmètre, avec  $A$  et  $h$  égaux à la surface et à l'arête maximale du triangle respectivement :

$$\alpha = \frac{6}{\sqrt{3}} \quad \rho = \frac{A}{s} \quad s = \frac{(a+b+c)}{2} \quad A = \sqrt{s(s-a)(s-b)(s-c)} \quad h = \max(a,b,c) \quad (3.2)$$

$a, b, c$  sont les arêtes du triangle. Le coefficient de normalisation  $\alpha$  assure que  $\mathbf{Q}_{equ} = 1$  pour un triangle équilatéral. Dans le pire des cas où le triangle est déformé (effondré) avec une surface égale à zéro,  $\mathbf{Q}_{equ} = 0$ .

Le critère de planarité est le suivant :

$$\mathbf{Q}_{nor} = \frac{\vec{n} \cdot \vec{n}_a + \vec{n} \cdot \vec{n}_b + \vec{n} \cdot \vec{n}_c}{3} \quad (3.3)$$

ou  $\vec{n}_a, \vec{n}_b$  et  $\vec{n}_c$  sont les normales des triangles voisins et  $\vec{n}$  est la normale du triangle.

La valeur idéale pour  $\mathbf{Q}_{nor}$  est 1 si les figures voisines sont coplanaires au triangle de test. De façon différente si les voisins sont perpendiculaires ou plus au triangle de test,  $\mathbf{Q}_{nor}$  donnera une valeur  $\leq 0$ . Pour ce comportement nous devons effectuer un filtrage préalable. Ce type de cas doit comporter un filtrage antérieur .

Nous pouvons observer sur la figure 3.13 l'évaluation du premier critère appliqué à un jeu des triangles sélectionnés : (a) est un triangle équilatéral, les vues (b) à (h) montrent des triangles en déformation progressive. A noter que, pour un triangle rectangle isocèle (figure 3.13(d)), le critère  $\mathbf{Q}_{equ}$  donne une valeur constante : avec  $a = b, c = \sqrt{2}a, A = a^2/2, s = a(1 + 1/\sqrt{2})$  donc  $\mathbf{Q}_{equ} = \sqrt{3}/(1 + \sqrt{2}) = 0.7174$ .

Le deuxième critère a été testé sur un maillage 3D synthétique, généré sur une sphère fermée, discrétisée par des nombres différents de triangles équilatéraux, ce qui affecte la variation angulaire des normales. En figure 3.14, toutes les sphères sont discrétisées par des triangles équilatéraux parfaits, mais nous obtenons une différence significative entre les normales voisines qui dénotent une surface anguleuse, par contre la

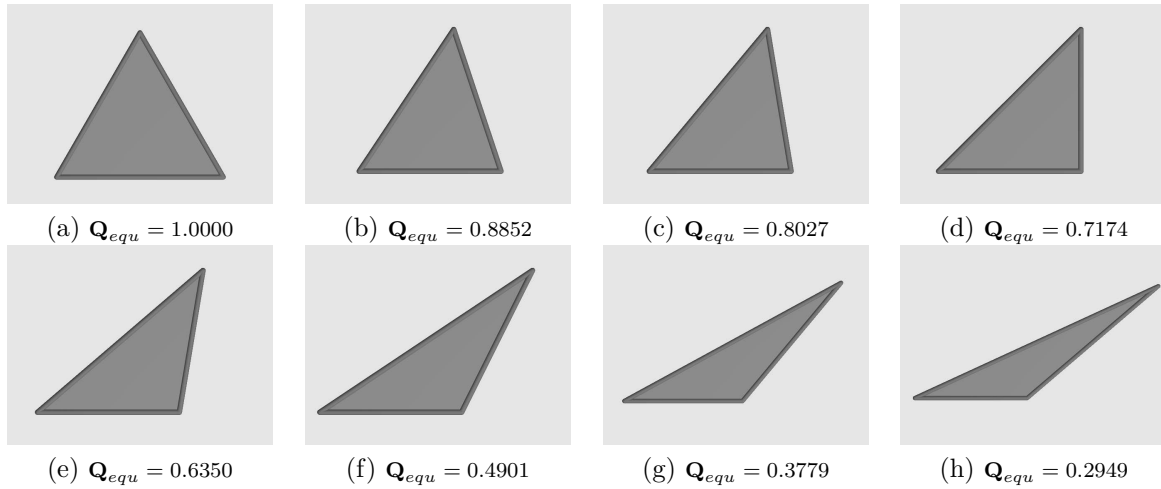

 FIG. 3.13 – Test du critère de la qualité  $Q_{equ}$  sur la forme du triangle, appliqué à divers triangles.

figure (e) contient les triangles moins équilatéraux que (a) à cause des erreurs numériques mais la surface reste encore plane localement..

Ensuite les critères  $Q_{equ}$  et  $Q_{nor}$  ont été testés pour vérifier leurs pertinences, sur une portion d'image de synthèse (voir figure 3.15) modélisée par l'algorithme de la boule pivotante et la surface déformable. On peut observer la qualité supérieure des triangles et de la planarité dans le maillage construit par la surface déformable dans les histogrammes de  $Q_{equ}$  et de  $Q_{nor}$ . Nous donnons pour chaque image, le nombre de sommets ( $\circ$ ) et de triangles ( $\Delta$ ), la valeur moyenne des deux critères et leurs histogrammes. En ce cas, le maillage *Surface déformable* est meilleur que le maillage BPA.

### 3.5.2 Résultats globaux.

Nous avons exploité les deux algorithmes *Surface déformable* et BPA avec différents types d'images pour obtenir leurs avantages et leurs limitations dans le but de comparer les deux méthodes. Les résultats à analyser ont été le nombre de points et de triangles, le temps de calcul et les critères de qualité.

La table 3.2 montre les résultats pour les différents paramètres de comparaison. Les valeurs idéales pour les qualités sont  $Q_{equ} = 1$  et  $Q_{nor} = 1$ . Les  $Pts_{init}$  sont le nombre de points du nuage au début et  $Pts_{fin}$  sont le nombre de sommets du maillage à la fin de chaque méthode.

Image	Fig.	$Pts_{init}$	$Pts_{fin}$		$\Delta$		$Q_{equ}$		$Q_{nor}$		Temps CPU [s]	
			BPA	SD	BPA	SD	BPA	SD	BPA	SD	BPA	SD
cat	3.16	10000	2231	2562	4379	5120	0.6970	0.6465	0.9657	0.9282	54.8	54.0
club	3.17	4215	2987	2562	5946	5120	0.6780	0.3957	0.9554	0.8099	29.21	71.4
mech	3.18	4102	2290	2562	4544	5120	0.6858	0.6132	0.8890	0.9159	31.01	50.8
scurv	3.19	8057	5302	2562	10296	5120	0.6905	0.5195	0.8595	0.8087	256.37	187.02
halfhead	3.20	8164	5737	2562	11255	5120	0.7340	0.6739	0.9658	0.8857	109.85	48.5
teapot	3.21	26103	6349	2562	12675	5120	0.8087	0.7438	0.9846	0.9166	241.73	84.2
teapot	3.21	"	"	10242	"	20480	"	0.7073	"	0.9002	"	1101.4

TAB. 3.2 – Comparaison entre BPA et Surfaces déformables avec les critères de qualité et le temps de calcul (Pentium 4, 3.06MHz).

Les différents exemples que nous avons testés, sont décrits dans les paragraphes suivants.

Les images "cat", "club", "mech", "teapot" et "halfhead", proviennent de la base de donnée de H. Hoppe [Hoppe 94]. Ce sont des nuages de points acquis sur des objets fermés avec une distribution aléatoire répartie de points, à l'exception de "teapot" qui a une distribution organisée. Ce type d'images est idéal pour tester les méthodes de maillage surfaciques et volumétriques que nous traitons dans ce

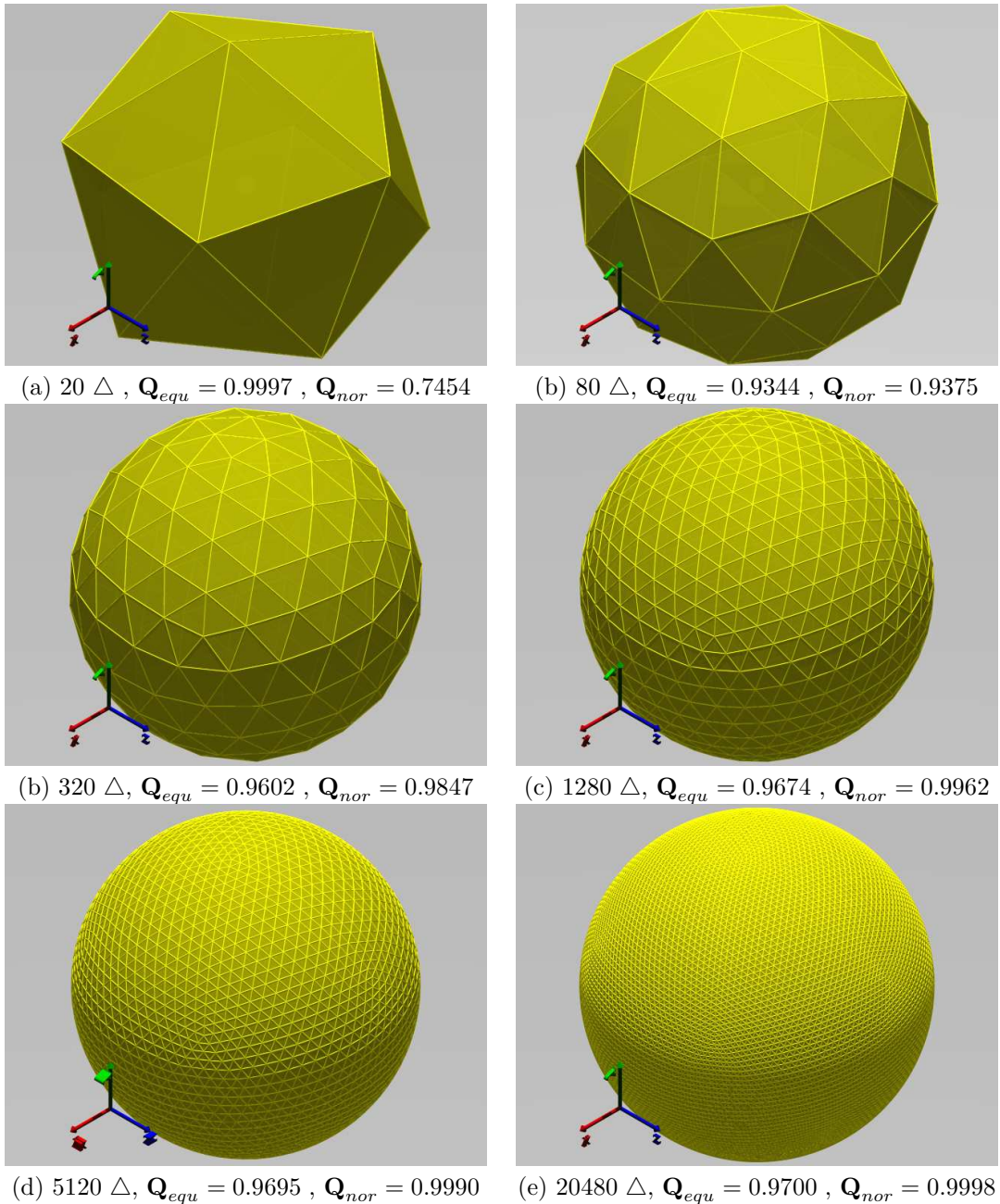
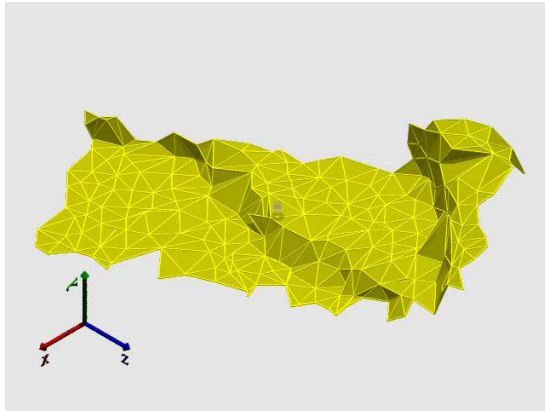


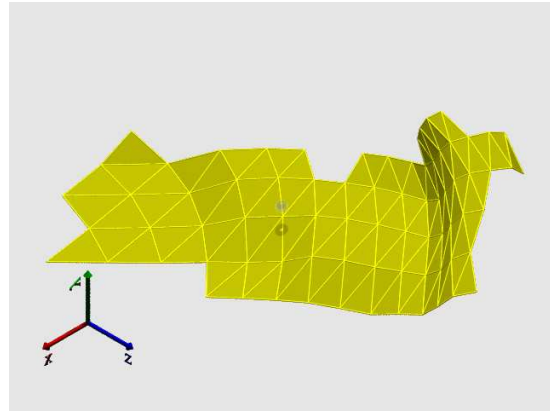
FIG. 3.14 – Test du critère de qualité  $Q_{nor}$  sur la courbure d'un maillage.

paragraphe. Par contre l'image "scurv" est une reconstruction de la méthode d'ICP. Chaque image a des caractéristiques spéciales pour évaluer le comportement des deux méthodes sur le maillage à former :

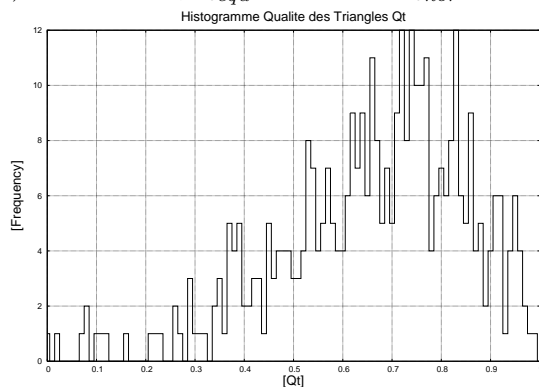
- "cat" : est un objet de type sculpture convexe, sans surface plane, sans ouverture.
- "club" : est un mélange de surfaces de forme libre et de forme géométrique avec une légère concavité et une surface plane.
- "mech" : est une forme géométrique avec des trous, des surfaces planes et des cylindres. Une difficulté supplémentaire est la faible densité des points, surtout dans la zone centrale de l'image.
- "teapot" : objet convexe, synthétique pur, mais avec quelques composantes difficiles à manipuler, qui sont l'avancée antérieure, la poignée et le couvercle.



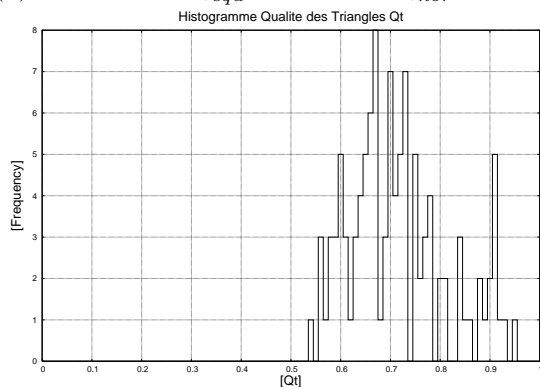
(a)  $226 \times 380 \triangle$ ;  $Q_{equ} = 0.6640$  et  $Q_{nor} = 0.9160$ .



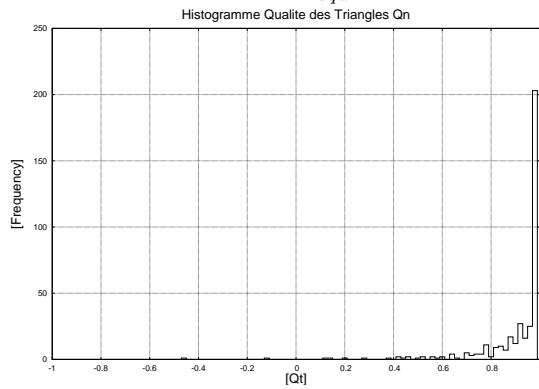
(b)  $78 \times 109 \triangle$ ;  $Q_{equ} = 0.7194$  et  $Q_{nor} = 0.9729$ .



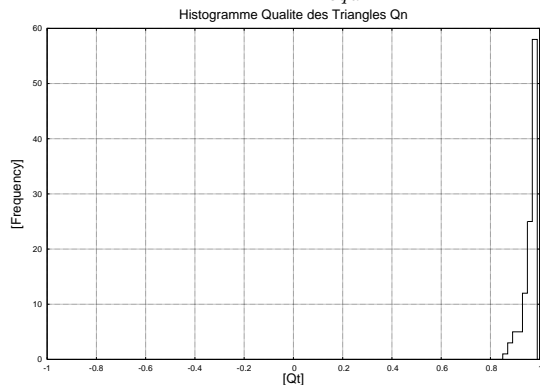
(c) Histogramme du  $Q_{equ}$  pour le BPA.



(d) Histogramme du  $Q_{equ}$  pour le SD.



(e) Histogramme du  $Q_{nor}$  pour le BPA.



(f) Histogramme du  $Q_{nor}$  pour le SD.

FIG. 3.15 – Critères  $Q_{equ}$  et  $Q_{nor}$  testés sur une portion d'image de synthèse.

- "halfhead" : est un détail d'une tête complète qui a des points à distribution plus dense dans les zones plus détaillées, et moins dense ailleurs. L'idée est de regarder le comportement des méthodes dans ce cas.
- "scurv" : est un détail de la reconstruction par ICP de 5 images synthétiques décimées. La distribution est très irrégulière avec un niveau d'erreur typique d'ICP. L'image a été coupée pour évaluer le comportement des méthodes avec des volumes ouverts, des surfaces planes coupantes et des formes concaves.

L'algorithme BPA a été exécuté pour toutes les images en exploitant au début la variante de recherche

des triangles souches à partir du nuage de points. Par la suite les surfaces ont été affinées plusieurs fois avec des boules de rayons plus grands jusqu'à obtenir des surfaces fermées (sauf pour "mech").

Le maillage initial pour la surface déformable a toujours été une sphère de 5120 triangles équilatéraux : ordre 5 de discrétisation d'une sphère à partir d'un icosaèdre, 2562 sommets, maillage représenté sur la figure 3.13(d). *Seulement* pour la figure 3.21(c) nous avons utilisé une sphère d'ordre 6 de 20480 triangles ; excepté cette dernière, le temps d'initialisation de la méthode *Surface déformable* est similaire pour toutes les images.

### 3.5.3 Résultats pour les différentes images de test.

Nous donnons quelques commentaires pour chaque image :

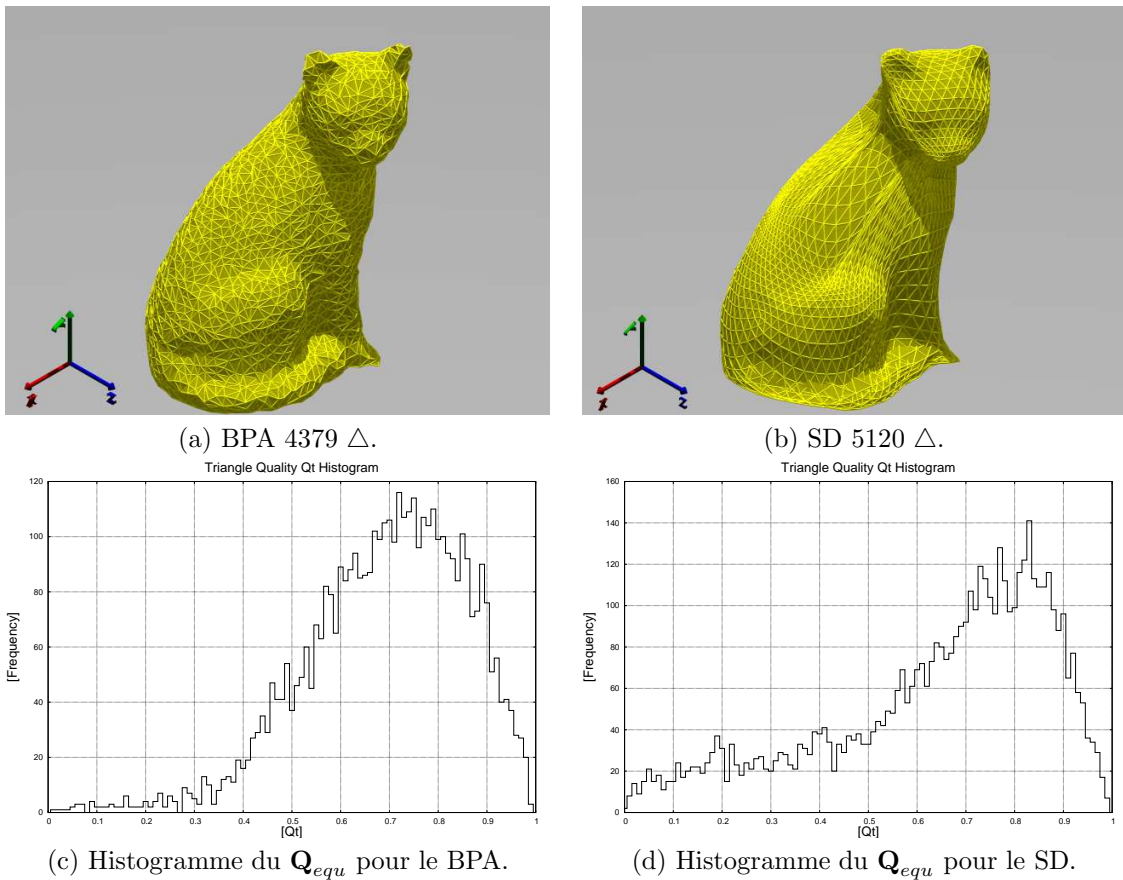


FIG. 3.16 – Exemple d'objet "cat" sur (a) avec BPA (b) surface déformable.

#### Images sur l'objet *cat* .

Voir la figure 3.16.

La méthode des surfaces déformables a pris 19,95s pour initialiser les polygones, 24,67s pour les premières 50 itérations et 4,39s pour les 50 itérations suivantes, ce qui donnent un total de 150 itérations. Le BPA a utilisé une seule séquence avec une boule de rayon 2,0 et a nécessité 54,82s.

#### Images sur l'objet *club* .

Voir la figure 3.17.



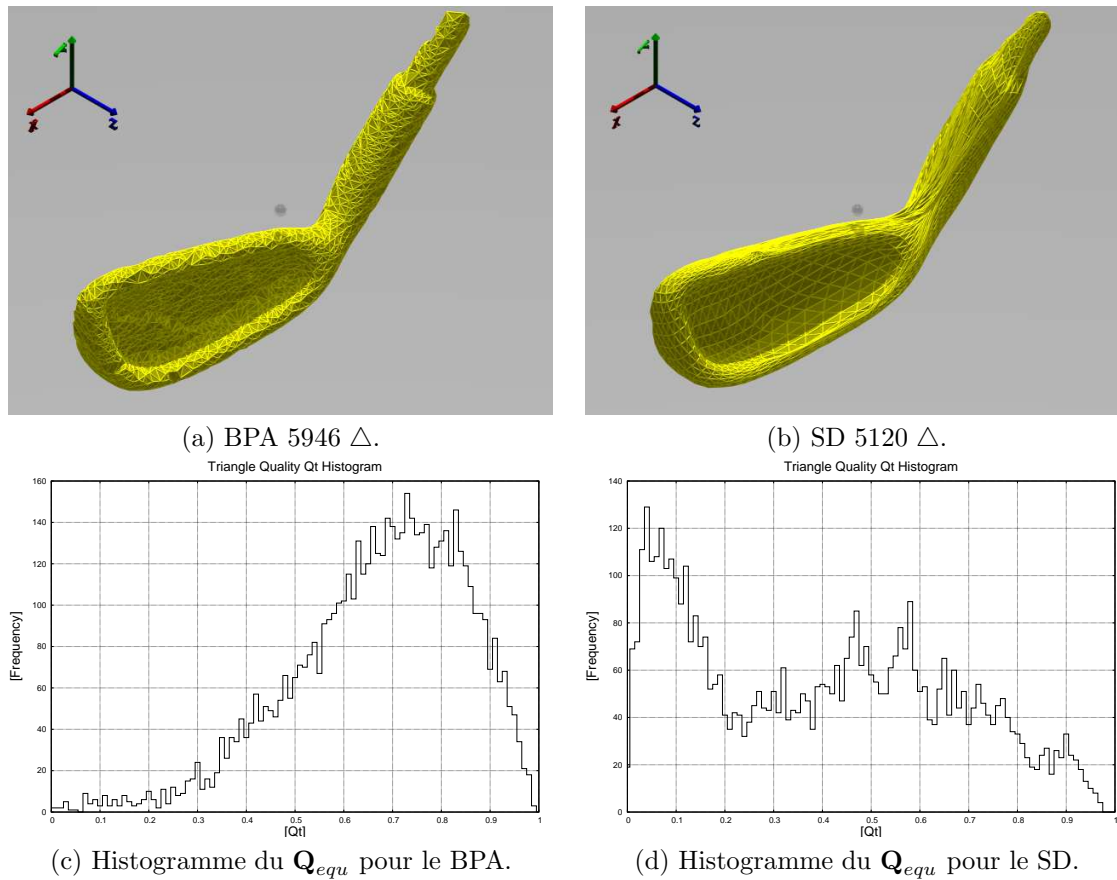


FIG. 3.17 – Exemple d'objet "club" sur (a) avec BPA (b) surface déformable.

L'histogramme (d) montre une accumulation de triangles de basse qualité en raison de la déformation supérieure de la surface.

La surface déformable est initialisée à partir de 19,34s pour continuer les premières 50 itérations pendant 20,34s. Les 300 itérations suivantes ont duré 4,5s pour des groupes de 50. La zone concave supérieure montre un étirement fort de la surface qui a formé des triangles dégénérés. Dans le cas étudié la méthode de BPA a commencé une première séquence de 11,88s avec un rayon de boule égal à 0,8 et les deux séquences suivantes 2,76s et 14,57s jointes à des rayons de 1,0 et 2,0 respectivement.

### Images sur l'objet *mech*.

Voir la figure 3.18.

Egalement avec une initialisation de 19,39s, la méthode des surfaces déformables a duré 20,9s les 50 itérations initiales et 3,5s pour le reste d'un total de 150 itérations. Nous pouvons constater le problème connu de la méthode avec les concavités des objets (le maillage ne rentre pas dans les trous). Le BPA a utilisé deux séquences avec une boule de 1,5 et une de 2,0 avec les temps de calcul de 17,49s et 13,52s respectivement. La distribution des points trop spacieuse dans la partie centrale de l'image a montré des limitations sur la méthode, qui a généré des régions ouvertes et donné des solutions difficiles.

### Images sur l'objet *scurv*.

Voir la figure 3.19.

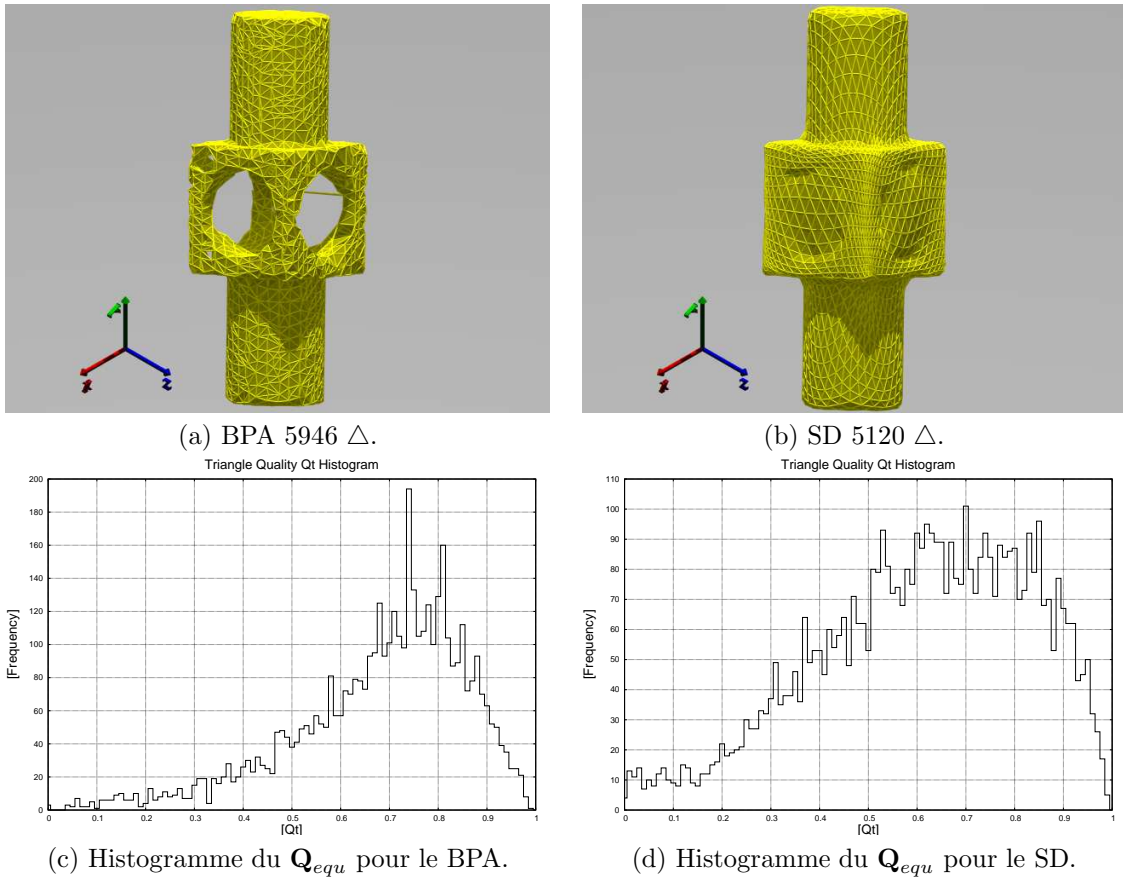


FIG. 3.18 – Exemple d’objet ”mech” sur (a) avec BPA (b) surface déformable.

L’histogramme (d) montre que la surface est fortement déformée dans les contours et a généré des triangles qui présentent des qualités moindres.

la méthode de la surface déformable présente des temps d’initialisation de 17.49s, pour les premières 100 itérations la durée est égal à 21.46s et pour les 100 suivantes donnent 6.9s avec un total de 300 itérations. L’algorithme de la boule pivotante a été appliqué à quatre scénarios de la façon suivante :

Séquence	rayon	temps [s]	triangles trouvés	itérations
Recherche par points	0.5	47.15	5601	7084
Recyclage de polygones	0.7	40.99	+3287	5466
Recyclage de polygones	0.9	97.90	+911	2037
Recyclage de polygones	2.0	70.33	+639	1644

### Images sur l’objet *halfhead*.

Voir la figure 3.20.

l’image a été traitée par la surface déformable avec 150 itérations et un temps de calcul similaire utilisé par l’image ”cat”. Le BPA a utilisé 7 séquences pour accomplir la surface en raison des distributions de points plus denses dans les zones détaillées de l’image.

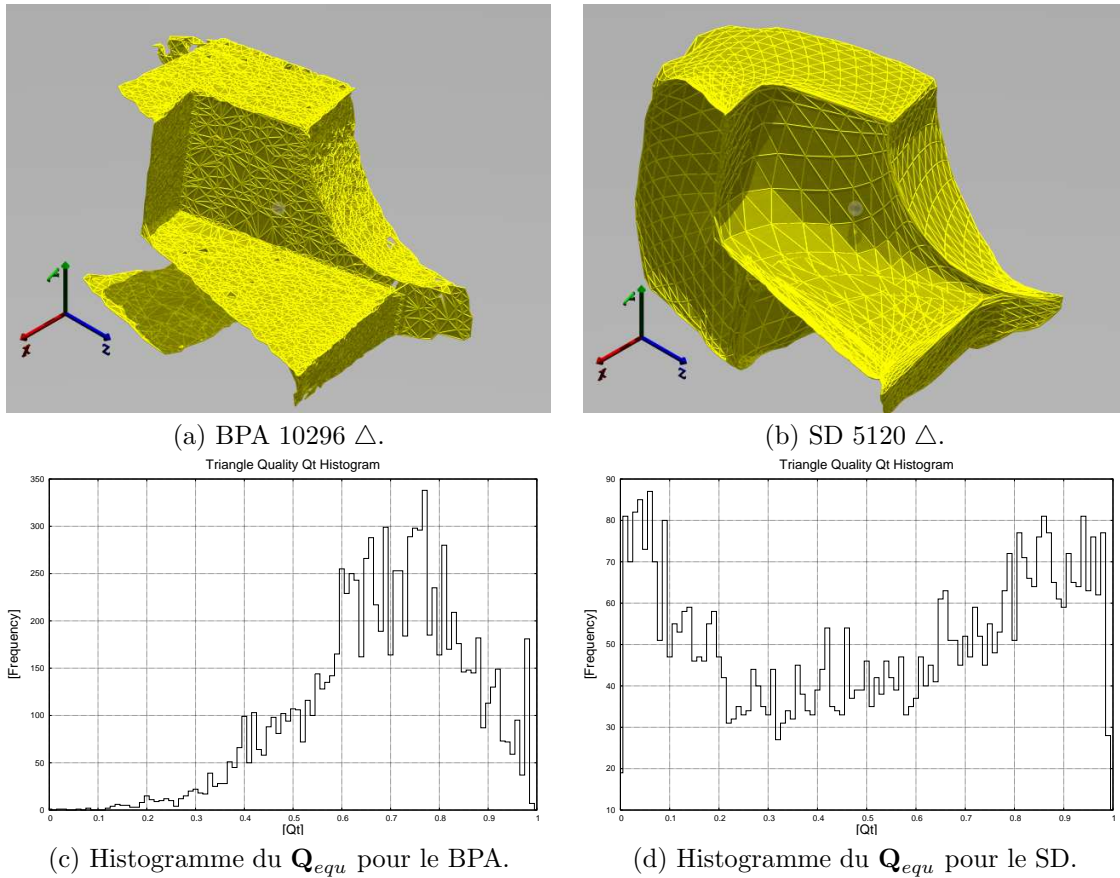


FIG. 3.19 – Exemple d'objet "scuro" sur (a) avec BPA (b) surface déformable.

Séquence	rayon	temps [s]	triangles trouvés	itérations
Recherche par points	1,8	51,79	7199	8019
Recyclage de polygones	2,0	4,20	+247	1132
Recyclage de polygones	3,0	15,32	+978	1298
Recyclage de polygones	3,5	13,49	+1747	2249
Recyclage de polygones	4,0	10,75	+1361	2131
Recyclage de polygones	5,5	14,30	+774	1248

### Images sur l'objet *teapot*

Voir la figure 3.21.

Nous avons exécuté deux maillages initiaux, une sphère d'ordre 5 et d'ordre 6, sur cette image pour regarder le résultat et la performance de la méthode et aussi arriver à un nombre de triangles plus comparable avec la méthode de BPA.

Pour l'exécution avec le maillage initial d'ordre 5 les résultats ont été similaires aux autres images analysées : initialisation=18,71s, les premières 100 itérations=54,02s, les 50 suivantes=5.88s, total=200 itérations.

En utilisant la sphère d'ordre 6, les temps ont augmenté d'une manière importante. L'initialisation est montée à 513,78s, les premières 100 itérations ont pris 321,0s et les 100 suivantes se sont élevées à environ de 50s à 70s.

Sur la figure 3.21(d), le détail montre l'étirement de la surface déformable qui s'adapte au versoir de la théière et a produit des triangles aigus. Au contraire, l'historgramme (e) met en évidence un maximum

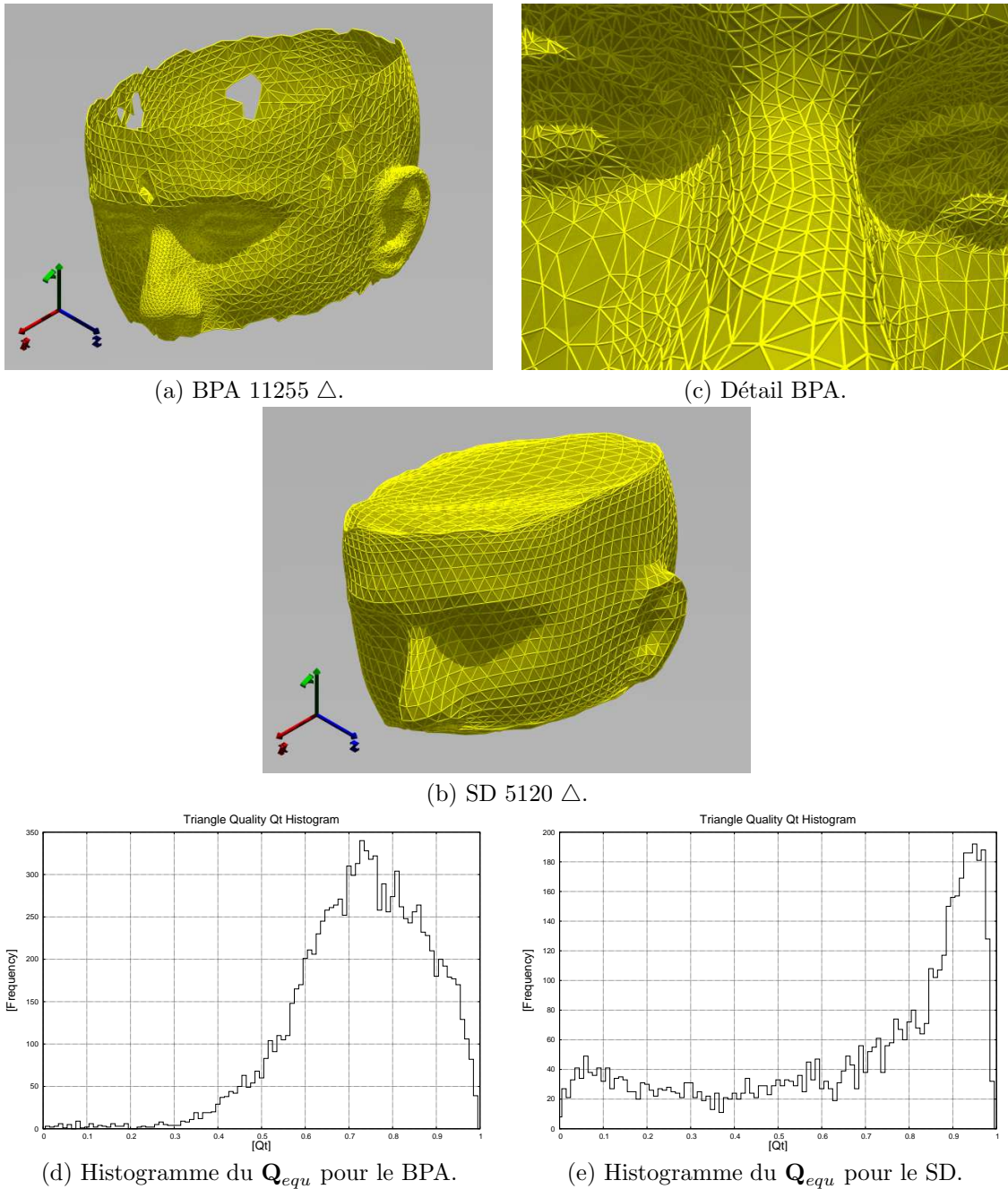
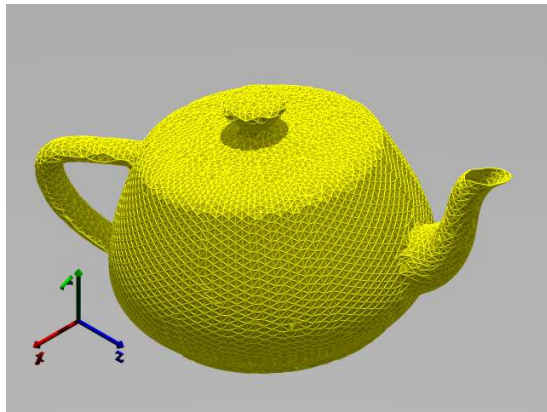


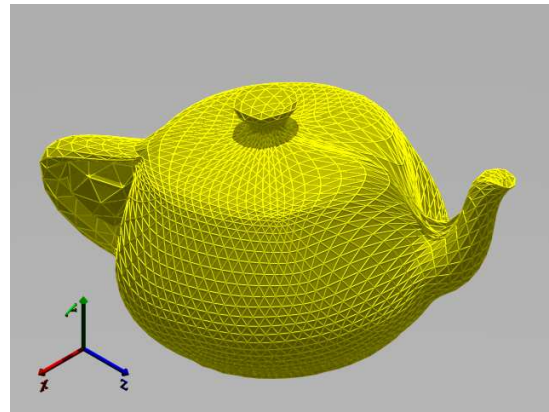
FIG. 3.20 – Exemple d’objet “halfhead” sur (a) avec BPA (b) surface déformable. Dans (c), le détail montre la densité variable des points qui a été traitée par BPA par plusieurs exécutions en utilisant un rayon initial de la boule plus petit.

dans les triangles rectangles ( $Q_{equ} = 0.71$ ), grâce à la distribution synthétique régulière des points au cours de la mise en oeuvre de la méthode de BPA.

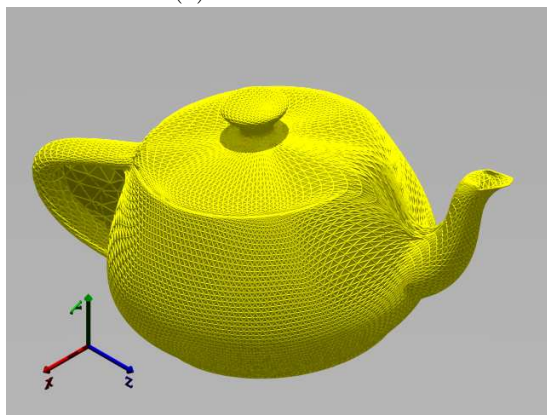
Pour réaliser un résultat acceptable comme ceux vus sur la figure 3.21(c), 500 itérations ont été nécessaires. Par contre, grâce à la distribution régulière de points de cette image, le BPA a utilisé une seule séquence avec une boule de rayon 1,0 et a pris un temps de CPU de 241,73s. Le BPA peut traiter les formes convexes et tubulaires correctement, mais il manque la finition douce des surfaces déformables.



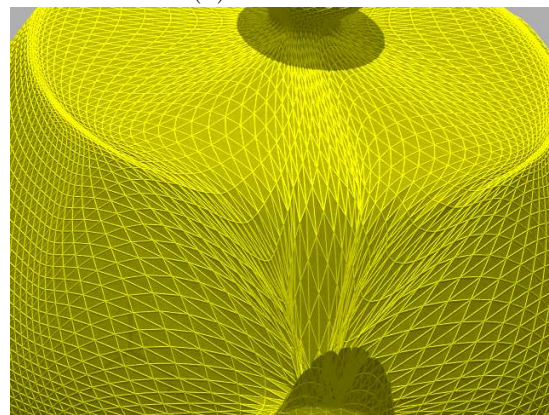
(a) BPA 12675  $\Delta$ .



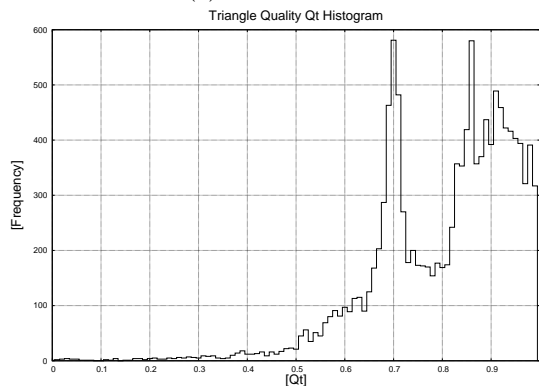
(b) SD 5120  $\Delta$ .



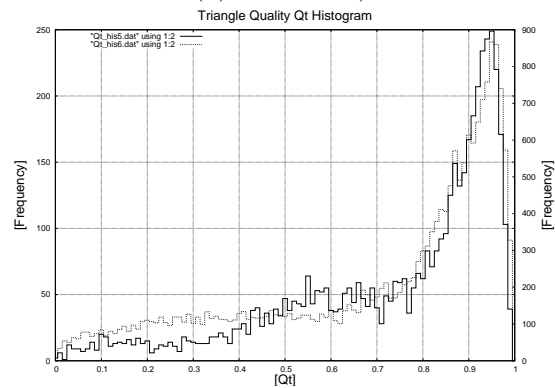
(c) SD 20480  $\Delta$ .



(d) détail de c).



(e) Histogramme du  $Q_{equ}$  pour le BPA.



(f) Histogramme du  $Q_{equ}$  pour le SD.

FIG. 3.21 – Exemple d'objet "teapot" : (a) avec BPA (b), (c) et (d) surface déformable.

### 3.6 Conclusion.

Dans ce chapitre, nous avons présenté nos travaux sur la construction de maillage triangulaire à partir de nuages de points 3D.

Cette méthode devait répondre au cahier des charges indiqué dans le chapitre 1. C'est pour y arriver que nous avons choisi de développer une version adaptée de l'algorithme *Ball Pivoting*. Nous avons montré que cet algorithme répond à plusieurs de nos besoins :

- flexibilité : capacité de l'exploiter en mode incrémental ou en mode global. De ce fait, il permet d'intégrer des nouveaux points dans un maillage existant ; il permet aussi, moyennant un choix adapté du rayon de la boule, de corriger certains défauts dans ces maillages existants (possibilité de boucher des trous par exemple).
- segmentation d'un maillage en primitives surfaciques : nous n'avons intégré que l'extraction de surfaces planes ; la même méthode serait exploitable pour des primitives quadriques.
- capacité de fusionner des données bruitées ou ayant des résolutions variables ; c'est ce besoin qui a justifié quelques adaptations vis-à-vis de l'algorithme de base. Les résultats à ce niveau, ne sont pas encore satisfaisants, puisque la construction de modèles à partir de données stéréoscopiques est toujours complexe.

En fin de chapitre, nous avons comparé notre version de la méthode BPA, avec l'approche *Surfaces Déformables* très utilisée dans la communauté, particulièrement pour des objets de forme libre (par exemple, des organes en imagerie médicale). L'une et l'autre méthodes ont leurs avantages et défauts selon le type d'objet ou le type d'images. Concernant les performances, elles sont assez semblables : elles ne permettent pas dans un avenir proche, d'envisager une modélisation en ligne d'objets 3D.

Nous savons acquérir des images 3D, les recaler, les fusionner dans une représentation de type maillage ; dans le chapitre suivant, nous allons nous intéresser à la planification de la tâche de modélisation.



## Chapitre 4

# Planification perceptuelle : sélection de la vue optimale

### 4.1 Introduction.

Dans ce chapitre nous allons traiter du problème connu dans la littérature sous le nom de *Next Best View* ; il s'agit de la planification perceptuelle appliquée à la reconstruction de l'environnement à partir d'images tridimensionnelles.

Comme vu dans les chapitres précédents, la modélisation 3D est un processus incrémental. A chaque itération, des données sont acquises à l'aide d'un capteur 3D placé au voisinage d'une bonne position, ces données sont recalées avec les perceptions précédentes puis fusionnées dans le modèle courant. Puis une représentation de l'environnement est produite ou mise à jour, typiquement un simple nuage de points 3D, un maillage triangulaire pour des applications graphiques, ou un ensemble de primitives surfaciques pour des applications de métrologie.

Une méthode de planification de perception pour une tâche de modélisation, doit être exécutée entre chaque acquisition. Le problème à résoudre consiste à déterminer la position optimale du capteur en fonction de l'information déjà acquise lors des acquisitions précédentes. Cette position optimale est trouvée généralement, par maximisation d'une fonction de qualité. Plusieurs critères doivent être pris en compte dans cette fonction ; ils dépendent de la géométrie du capteur, de l'équipement utilisé pour déplacer le capteur (un robot mobile, un opérateur...) et de contraintes liées à l'application, en particulier liées à la résolution du modèle final. C'est aussi cette fonction de planification, qui doit détecter la fin de la tâche de modélisation, lorsque tout l'espace occupé par l'entité à modéliser a été scruté avec la résolution requise.

Lorsque la planification a produit une position optimale, le capteur devra être déplacé en cette position avec le plus de précision possible. En effet, la convergence de la méthode ICP traitée au chapitre 2, dépend de l'estimée initiale de la position du point de vue par rapport à l'ensemble des autres points de vue atteints jusqu'à l'itération courante.

Dans la section suivante, nous évoquons l'état de l'art sur la planification perceptuelle pour la modélisation 3D. Puis, dans les sections 4.3 et 4.4, nous décrivons respectivement la méthode et la fonction qui l'applique ; ces travaux ont été réalisés initialement par M.Lozano dans le cadre du projet CAMERA ; nous les avons poursuivis pour essayer d'améliorer les résultats, et en particulier, pour résoudre des problèmes de convergence. Enfin, en section 4.5, une version sous-optimale, très simplifiée vis-à-vis de l'approche originale, est proposée pour le contexte "Modélisation d'objets".

### 4.2 État de l'art.

Ce problème de planification de la perception, quelque soit la tâche visée, n'est pas très étudié dans la littérature, sans doute parce qu'il nécessite à la fois des connaissances sur la perception, la géométrie algorithmique et les techniques de planification. Les méthodes diffèrent d'abord selon la tâche perceptuelle



(modélisation, reconnaissance, navigation...), et de ce fait, selon les connaissances dont dispose le système pour effectuer sa planification : modèle en cours de construction, modèle d'objets ou d'environnements déjà appris... Ensuite, la complexité des méthodes proposées dépend largement des modalités sensorielles : type de capteur (3D ou 2D) et des actionneurs qui vont le déplacer (opérateur, platine, robot mobile...).

L'état de l'art pour la thématique de *Next Best View* concerne plutôt la planification perceptuelle pour l'exploration d'un environnement et pour la navigation des robots mobiles ; lorsque la fonction *Simultaneous Localization and Mapping* (SLAM) est exécutée, le robot planifie sa prochaine position (dans le cas d'une fusion d'images acquises en des points précis) ou sa prochaine trajectoire (dans le cas de la fusion de toutes les images acquises durant un mouvement).

Citons quelques travaux caractéristiques. Dans [Tarabanis 96], Tarabanis&Tsai présentent une approche purement géométrique ; à partir du modèle courant de la scène, un algorithme calcule les volumes de visibilité depuis lesquels secteurs occultés pourraient être vus ; dans ces volumes, d'autres critères doivent être exploités pour sélectionner la prochaine position d'acquisition, par exemple position atteignable la plus proche, position qui minimise les manoeuvres dans le cas d'un robot mobile non holonome. Dans [Pito 96], R.Pito traite d'un problème plus adapté à la modélisation d'objet : la complexité est réduite, en imposant au capteur de se déplacer uniquement sur une surface cylindrique dont l'axe passe dans l'objet à modéliser. Comme le capteur reste avec une orientation selon le rayon du cylindre, sa position est seulement déterminée par deux paramètres ; nous nous sommes inspirés de ces travaux pour adapter la méthode générale décrite dans les sections suivantes, au cas spécifique de la modélisation d'un objet.

Dans [González-Baños 00], González-Baños&Latombe traitent le problème classique du gardien de musée ; en supposant qu'une carte polygonale 2D de l'environnement est connue, il faut trouver le nombre minimum de capteurs et leurs positions respectives dans cet environnement, afin d'assurer que les champs de vue de tous les capteurs couvrent tout l'environnement. L'approche proposée dans [González-Baños 00], est fondée sur l'application de méthodes probabilistes pour le placement des capteurs ; la solution finale est sous-optimale, mais elle est atteinte en un temps de calcul réduit. Cet article ne traite donc pas de la tâche de modélisation. De même, il existe plusieurs auteurs qui prennent en compte explicitement les problèmes de localisation d'un robot mobile lors de la planification des trajectoires ; connaissant la position des amers dans la scène, la trajectoire du robot et les modalités du capteur embarqué (typiquement, une caméra active avec azimuth, site et zoom commandables) sont planifiées afin que le robot ait toujours dans son champ de vue un ou plusieurs amers présents.

D'autres travaux traitent du choix des positions d'acquisition durant la construction d'un modèle 2D par un robot équipé d'un capteur laser à balayage horizontal. Dans [Colly 89], une approche purement géométrique est proposée : à partir du modèle courant, des contraintes successives (orienter le capteur vers une frontière courante de visibilité, percevoir des segments déjà perçus, éviter les occultations...) sont prises en compte en réduisant progressivement la zone d'intérêt par des intersections de polygones. Dans [Tovar 01], le même problème est considéré par une approche d'optimisation d'une fonction d'utilité, non linéaire par rapport à la configuration du capteur (position, orientation).

Citons d'autres recherches faites sur *Next Best View*, centrées sur la construction du modèle, soit d'un objet, soit d'un environnement :

- [García 98] : reconstruction des objets, algorithme à deux étapes afin de détecter les occultations ; choix d'une position sur une sphère discrétisée centrée sur l'objet.
- [Banta 95] : reconstruction des objets ; exploitation d'une grille d'occupation pour représenter les obstacles qui provoquent des occultations ; recherche d'une position pour explorer une zone cachée (concavités).
- [Wong 99] : reconstruction des objets, méthode également fondée sur une sphère sur laquelle se déplace le capteur ; possibilité de mesurer la performance de l'algorithme.
- [M. K. Reed 99] : reconstruction des objets, méthode basée sur les surfaces visibles et les volumes accessibles par le capteur.
- [Sequeira 96] : reconstruction des scènes 3D ; algorithme fondé sur la détection des occultations ;
- [Klein 00] : reprise des travaux de V.Sequeira, mais la détection des occultations est sous-traitée à une carte graphique, donc est faite en *hardware* ; calcul du point de vue optimal en exploitant un

critère de qualité.

- [Nüchter 03] : reconstruction des environnements pour la navigation 2D (en intérieur) ; méthode inspirée de l’algorithme de la galerie d’art [González-Baños 00].

Notre recherche est basée sur la méthode générale présentée dans [Massios 98], [Sanchiz 99] et [Lozano 02] : le problème du *Next Best View* est traité pour planifier une tâche autonome de construction du modèle 3D d’un environnement, avec une plate-forme mobile disposant de 5 degrés de liberté. Notre méthode emploie une stratégie d’optimisation similaire et la même représentation de la scène, une carte discrète de *voxels* pour mémoriser les secteurs occupés, vides ou non encore perçus le long du processus de modélisation.

Notre contribution concerne deux extensions de cette méthode : d’abord la qualité du modèle est prise en considération, dans de nouvelles expressions des critères d’utilité pour un point de vue donné. Ensuite la méthode d’optimisation a été modifiée de sorte que le capteur puisse atteindre des positions qui vont nous fournir une plus grande quantité d’informations.

## 4.3 Description de la méthode.

### 4.3.1 Une vue d’ensemble de notre système.

Nous proposons ici de donner une vue d’ensemble de notre méthode. Un *voxel map*, similaire à la structure de *clusters* déjà exploitée dans les chapitre 2 et 3, est employé par notre algorithme comme une représentation d’environnement. Ce *voxel map* est utilisé dans l’algorithme de recalage ICP au chapitre 2 et dans la méthode BPA au chapitre 3, pour avoir un accès efficace aux données 3D déjà acquises ; à partir des coordonnées xyz d’une position désirée, une clef est calculée pour trouver dans le *voxel map* les points 3D du modèle courant proche de cette position.

Comme déjà décrit au chapitre 2, un *voxel map* est une matrice 3D de *voxels* ; chaque *voxel* représente un petit cube dans la zone de travail. Le choix de la résolution du *voxel map* est un paramètre important de notre méthode. Pour l’utilisation courante la taille du *voxel* est fixe, mais dans de futures extensions, ce volume élémentaire devrait être variable. Il conviendrait de commencer une première tâche de modélisation avec une résolution grossière (typiquement, des cubes de 20x20x20cm) pour modéliser un environnement puis d’améliorer lors d’une deuxième passe, la qualité du modèle en augmentant la résolution du *voxel map* (typiquement, des cubes de 5x5x5cm). La résolution de cette représentation, doit être compatible avec la résolution du capteur (pour un laser, quelques cms) et avec la précision de la méthode prévisible de recalage.

Cette stratégie de modélisation en plusieurs étapes avec des résolutions de plus en plus fines des données acquises sur les scènes, reproduit un comportement humain classique. L’observateur fera une première visite rapide de la zone de travail pour établir un modèle brut initial ; puis, selon les besoins, il peut planifier d’autres visites plus détaillées jusqu’à ce que le modèle final ait la qualité exigée.

Pour chaque *voxel*, nous connaissons la position dans la *voxel map* et une étiquette ; si cette étiquette est *occupé*, alors, nous mémorisons aussi le nombre de points acquis en ce *voxel*, la distance la plus proche du capteur lorsque ces points ont été acquis et la normale moyenne à la surface. Les étiquettes possibles sont :

- *inconnu* : un *voxel* dans un secteur non encore exploré. Au début dans un environnement inconnu tous les *voxels* sont étiquetés *inconnu* ; ils perdent cette étiquette dès qu’ils sont atteints par le champ de vue du capteur.
- *vide* : un *voxel* dans un secteur perçu, mais aucun point n’a été acquis en cette position. Des rayons optiques ont traversé ce *voxel* sans rencontrer une surface.
- *occupé* : un *voxel*, sur lequel des points ont été acquis. Un tel *voxel* appartient à une surface qui est un obstacle pour le mouvement du capteur, et qui occulte les *voxels* situés derrière lui vis-à-vis de la position du capteur, sur le même rayon optique.
- *occulté* : le *voxel* dans un secteur vu par le capteur, mais inaccessible parce que le secteur se trouve derrière un *voxel* marqué *occupé*.

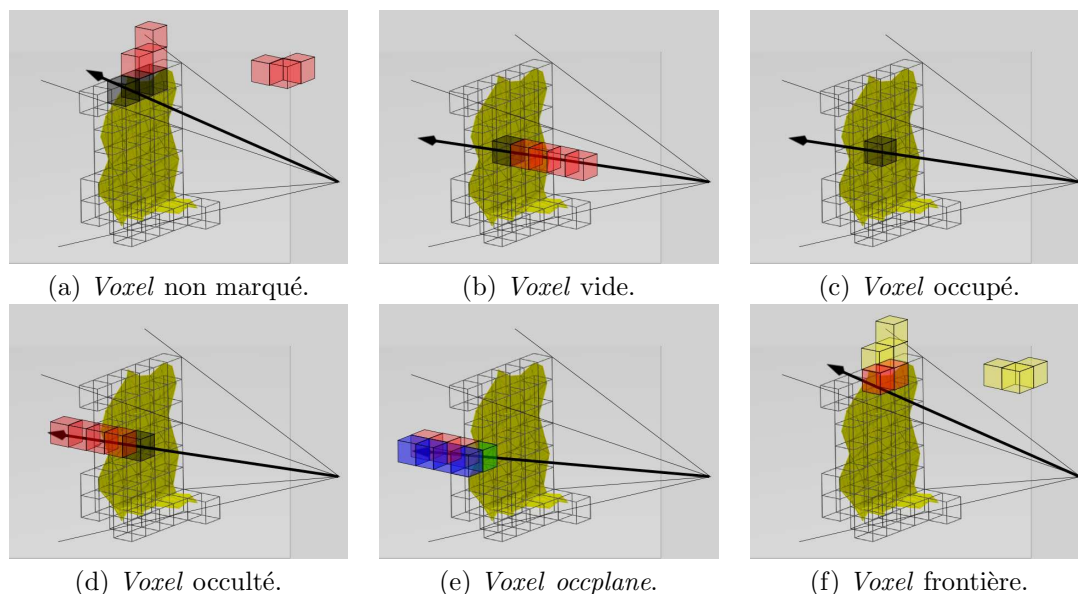


FIG. 4.1 – Types de voxels.

- *occlplane* : un *voxel* d'abord marqué *occulté*, mais qui est situé à côté d'un *voxel* marqué *vide*, sur une de ses six facettes. Ce sont les limites de visibilité créées par les objets qui apparaissent au premier plan dans la scène perçue pendant une acquisition précédente.
- *frontière* : un *voxel* d'abord marqué *occupé*, mais adjacent avec n'importe laquelle de ses six facettes à un *voxel* marqué *inconnu*. Un tel *voxel* était à la limite du champ de vue pendant une acquisition précédente.

Pour les *voxels* marqués *occupés* et *frontière*, la normale moyenne de la surface formée avec les points actuellement acquis en ce *voxel*, est calculée et mémorisée.

Les *voxels* marqués *frontière* ont été introduits pour deux raisons : (1) leur existence facilite la détection de la frontière entre les secteurs connus et les secteurs inconnus de l'environnement, et (2) la prochaine acquisition à planifier, doit faire diminuer le nombre des *voxels* marqués *inconnu*, mais doit également percevoir encore quelques *voxels* marqués *occupé*, de sorte que le recalage puisse être fait entre le modèle courant et la prochaine image.

Par conséquent la prochaine vue doit couvrir un certain nombre de *voxels* marqués *frontière*. Par ailleurs, afin d'améliorer la qualité du modèle acquis, il est préférable de percevoir une surface avec l'axe du capteur, ou l'axe optique de la caméra, orthogonal à cette surface. Bien que l'orientation des surfaces non encore perçues (*occulté* ou *inconnu*) soit inconnue, elle peut être estimée à partir des surfaces les plus proches déjà détectées.

Les *voxels* marqués *frontière* sont donc utiles pour estimer les normales en ces surfaces encore non perçues voisines.

De même, pour des *voxels* marqués *occlplane*, la normale est calculée et gardée en mémoire ; il s'agit en ce cas, de la normale au rayon optique qui a créé cette limite de visibilité. Cette normale est utilisée pour choisir la vue optimale afin de percevoir un secteur occulté.

### 4.3.2 La fonction d'utilité pour un point de vue.

Notre méthode est fondée sur l'optimisation d'une fonction d'utilité, afin de rechercher la meilleure position et orientation du capteur pour acquérir une vue optimale en ce qui concerne le modèle courant.

Nous avons tout d'abord validé notre méthode NBV sur un environnement de simulation utilisé par

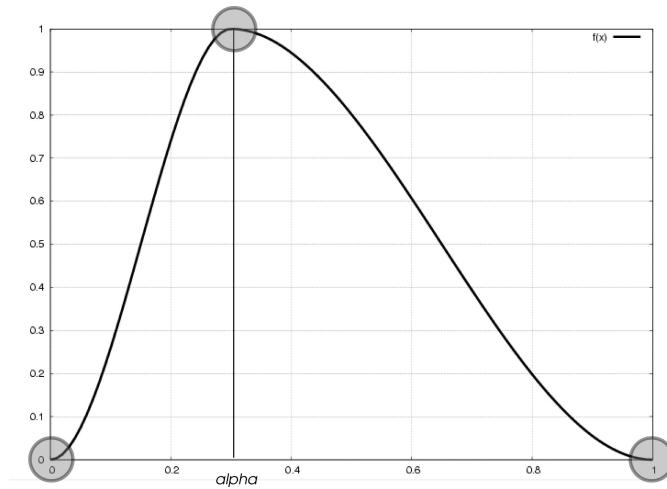


FIG. 4.2 – Variation de la fonction d'utilité pour chaque type de voxel pour un  $\alpha$  donné et les restrictions à accomplir sur les points indiqués.

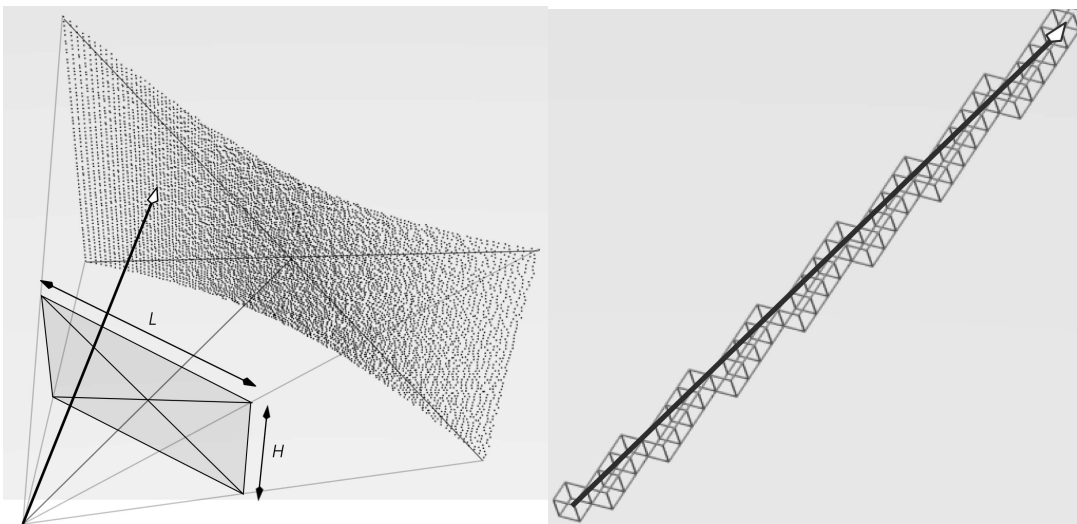


FIG. 4.3 – Matrice d'acquisition d'une résolution  $L \times H$  à gauche. Parcours des rayons à droite.

V.Sequeira [Klein 00] et M.Lozano [Lozano 02] dans le cadre du projet CAMERA. Il s'agit de l'exploration d'un environnement d'intérieur par un robot se déplaçant en  $(x, y)$  sur un sol plat ; un capteur 3D est monté sur le robot sur un mécanisme qui donne trois degrés supplémentaires, l'élévation  $z$  et l'orientation de l'axe du capteur  $(\theta, \phi)$ . Un point de vue ou un placement est donc défini par cinq paramètres ; dans cette configuration, la méthode d'optimisation devra estimer les coordonnées  $(x, y, z)$  de la position en laquelle l'origine du repère capteur doit être emmenée, et les angles d'inclinaison et d'orientation <sup>1</sup>,  $\theta$  et  $\phi$ , de l'axe de ce capteur. Les autres paramètres liés au capteur sont constants : les distances minimales et maximales d'acquisition de données et l'angle du champ de vue, entre autres.

Le critère à optimiser doit mesurer l'intérêt d'un placement possible du capteur par rapport à son placement courant et à l'état du modèle de l'environnement. La prochaine vue devrait fournir les caractéristiques

<sup>1</sup>pan and tilt

téristiques suivantes :

- le champ de vue doit avoir un certain taux de recouvrement avec les champs de vue des positions précédentes, pour rendre plus facile le recalage.
- elle doit résoudre les occultations, en d'autres termes, le placement choisi doit permettre de percevoir des surfaces cachées derrière les secteurs déjà occupés dans le modèle courant.
- elle doit percevoir des nouveaux secteurs inconnus pour augmenter la zone de travail connue.
- et elle doit améliorer si possible, la qualité des *voxels* déjà marqués comme *occupé* : pour chaque *voxel*, la qualité dépend de
  - la résolution, donc le nombre de points acquis, ce qui dépend de la distance entre les positions du capteur et le *voxel*.
  - et de la précision des mesures, qui dépend de cette distance, mais aussi de l'angle moyen entre les rayons optiques issus du capteur et la normale estimée à la surface en ce *voxel*.

Pour un placement donné, ce critère à optimiser est fondé

1. sur les pourcentages des *voxels* marqués dans le modèle courant, *occupé*, *inconnu*, *ocplane* et *frontière*, qui appartiendraient au champ de vue atteint depuis cette position
2. et sur la qualité d'acquisition qui peut être exprimée en utilisant les prévisions de résolution et d'angle d'incidence pour chaque *voxel* marqué *occupé* du champ de vue atteint.

La figure 4.2 montre la forme générale de la fonction d'utilité d'un placement par rapport au pourcentage de *voxels* de chaque type, atteint dans le champ de vue de ce placement.

Comment calculer ces pourcentages ? Le planificateur doit les estimer par simulation, en fonction des caractéristiques du capteur, du placement à évaluer et de l'état du modèle courant, mémorisé dans le *voxel map*. A chaque acquisition, le capteur (laser ou stéréovision) effectue un balayage du champ de vue pour mesurer la profondeur en chaque pixel de l'image 3D, une matrice de taille  $LxH$  (voir la figure 4.3 à gauche). Chaque pixel de la matrice correspond à un rayon (faisceau laser ou rayon optique) qui est lancé depuis l'origine du repère capteur : ce rayon est analysé voxel par voxel en utilisant l'algorithme de Bresenham pour calculer la fonction d'optimisation.

- Si ce rayon rencontre une surface située entre les distances minimales et maximales de perception, alors un point 3D est instancié en ce pixel de l'image 3D ;
- sinon ce pixel ne porte pas d'information (pas de retour laser ou pixel non apparié).

Comme tout l'espace est discrétisé par *voxels*, la méthode d'analyse du rayon devient un problème de création d'une ligne droite dans un milieu numérique. La solution la plus utilisée est fournie l'algorithme de Bresenham, illustré en figure 4.3 à droite. Pour un placement à évaluer et pour chaque pixel de l'image 3D qui serait acquise en ce placement, un rayon est donc lancé : il traverse les *voxels* marqués *vide* dans le modèle courant et s'arrête dès qu'il rencontre

- un *voxel* marqué *inconnu* ou *ocplane* car en ce cas, on accroît potentiellement la portion connue de l'environnement à explorer, notons que par construction, il ne peut pas atteindre directement un *voxel* marqué *occulté* ;
- un *voxel* marqué *occupé* ou *frontière*, car en ce cas, on perçoit de nouveau une zone déjà perçue, ce qui rend possible un recalage et peut améliorer la qualité du modèle.

A la fin du balayage de l'image 3D qui serait acquise depuis le placement à évaluer, nous avons les pourcentages de *voxels* de chaque type qui seraient atteints dans ce champ de vue. Ceci est la valeur pour  $x$  dans la fonction générique représentée en figure 4.2.

Pour un type donnée de *voxels*, la fonction d'utilité  $f(x)$  doit être optimale,  $f(x) = 1$ , pour un pourcentage nominal,  $x = \alpha$ , et minimale,  $f(x) = 0$ , si et seulement si

- $x = 1$  : uniquement ce type de *voxels* est perçu,
- $x = 0$  : aucun pixel de ce type n'est perçu.

La fonction d'utilité globale sera la somme des fonctions d'utilité pour chaque type de *voxels*. Les valeurs de  $\alpha$  pour les différents types de *voxel*, sont des paramètres de l'algorithme. Dans la phase initiale, le cas d'une utilité nulle pour  $x = 1$  se justifie parfaitement ; par exemple, si seuls des *voxels* marqués *inconnu* sont perçus, alors le champ de vue est très intéressant pour avancer l'exploration, mais le taux de recouvrement avec les vues précédentes étant nul, l'image acquise depuis ce placement ne pourrait pas être recalée avec le modèle courant. Le cas d'une utilité nulle pour  $x = 0$  se justifie de manière duale. Par

contre en fin d'exploration, il se peut que tous les *voxels* aient été perçus au moins une fois ; un point de vue peut-être intéressant pour améliorer la qualité du modèle, même si  $x = 0$  pour le type *inconnu*.

Dans les travaux précédents présentés dans [Lozano 02] et [Sanchiz 99], JM.Sanchiz et M.Lozano avaient choisi une courbe cubique pour la forme de  $f(x)$ . Après évaluation de ces travaux, nous avons choisi une courbe cubique, mais segmentée en deux parties pour donner un nombre additionnel de restrictions, surtout pour les cas  $x = 0$  et pour la dérivabilité aux frontières :

$$f(x) = \begin{cases} f_1(x) = A_1x^3 + B_1x^2 + C_1x + D_1 & , \quad x < \alpha \\ f_2(x) = A_2x^3 + B_2x^2 + C_2x + D_2 & , \quad x \geq \alpha \end{cases} \quad (4.1)$$

avec les restrictions suivantes pour la valeur de la fonction et de sa dérivée :

$$\begin{aligned} f(0) &= 0 & \dot{f}(0) &= 0 \\ f(\alpha) &= 1 & \dot{f}(\alpha) &= 0 \\ f(1) &= 0 & \dot{f}(1) &= 0 \end{aligned} \quad (4.2)$$

et les restrictions de forme suivante :

$$\begin{aligned} f(x) &> 0 & \text{pour tous } x \in [0, 1] \\ \dot{f}(x) &> 0 & \text{pour tous } x \in [0, \alpha] \\ \dot{f}(x) &< 0 & \text{pour tous } x \in [\alpha, 1] \end{aligned} \quad (4.3)$$

Exprimons la dérivée de  $f(x)$  :

$$\dot{f}(x) = \begin{cases} 3A_1x^2 + 2B_1x + C_1 & , \quad x < \alpha \\ 3A_2x^2 + 2B_2x + C_2 & , \quad x \geq \alpha \end{cases} \quad (4.4)$$

Les restrictions appliquées sur la fonction  $f$  et sa dérivée donnent 8 équations pour 8 inconnues.

$$\begin{aligned} D_1 &= 0 & C_1 &= 0 \\ A_1\alpha^3 + B_1\alpha^2 + C_1\alpha + D_1 &= 1 & 3A_1\alpha^2 + 2B_1\alpha + C_1 &= 0 \\ A_2\alpha^3 + B_2\alpha^2 + C_2\alpha + D_2 &= 1 & 3A_2\alpha^2 + 2B_2\alpha + C_2 &= 0 \\ A_2 + B_2 + C_2 + D_2 &= 0 & 3A_2 + 2B_2 + C_2 &= 0 \end{aligned} \quad (4.5)$$

La solution qui donne ces paramètres en fonction de  $\alpha \in [0, 1]$  est :

$$\begin{aligned} A_1 &= -\frac{2}{\alpha^3} & A_2 &= -\frac{2}{(\alpha-1)^3} \\ B_1 &= \frac{3}{\alpha^2} & B_2 &= \frac{3(\alpha+1)}{(\alpha-1)^3} \\ C_1 &= 0 & C_2 &= -\frac{6\alpha}{(\alpha-1)^3} \\ D_1 &= 0 & D_2 &= \frac{3\alpha-1}{(\alpha-1)^3} \end{aligned} \quad (4.6)$$

Nous vérifions les conditions 4.3 qui sont détaillées comme suit :

$$f(x) = \begin{cases} \frac{2}{\alpha^3}(-x + \frac{3}{2}\alpha)x^2 & , \quad x < \alpha \\ \frac{1}{(1-\alpha)^3}(x-1)^2(x - (\frac{3}{2}\alpha - \frac{1}{2})) & , \quad x \geq \alpha \end{cases} \rightarrow f(x) > 0 \text{ pour tous } x \in [0, 1] \quad (4.7)$$

et

$$\dot{f}(x) = \begin{cases} \frac{6}{\alpha^3}(\alpha - x)x & , \quad x < \alpha \rightarrow \dot{f}(x) > 0 \text{ pour tous } x \in [0, \alpha] \\ \frac{6}{(1-\alpha)^3}(x-1)(x-\alpha) & , \quad x \geq \alpha \rightarrow \dot{f}(x) < 0 \text{ pour tous } x \in [\alpha, 1] \end{cases} \quad (4.8)$$

Nous utilisons ces résultats pour tous les types de *voxels*, chacun avec un  $\alpha$  choisi selon les besoins de la modélisation. Les valeurs choisies pour ces paramètres  $\alpha$  doivent être compatibles entre elles (leur somme vaut 1). Dans la version courante, ces valeurs sont fixes. Elles pourraient être adaptées en fonction de l'état

d'avancement de l'exploration. Par exemple, pour favoriser au début les placements qui permettent de voir les zones occultées plutôt que les zones inconnues, il suffirait de choisir un pourcentage optimal pour le type *ocplane* supérieur à celui du type *inconnu*, puis d'inverser ce choix pour modifier le comportement de notre robot explorateur.

Nous avons testé plusieurs valeurs possibles des paramètres  $\alpha$ . Nous donnons en figure 4.4 la forme de la fonction à minimiser, pour les valeurs suivantes de  $\alpha$  :

$$\alpha_{occupe} = 0.20, \alpha_{ocplane} = 0.375, \alpha_{inconnu} = 0.375 \text{ et } \alpha_{frontiere} = 0.05$$

Pour prendre en compte la qualité du modèle, nous rajoutons uniquement une contrainte sur l'orientation; cette contrainte ne peut être exprimée que pour les *voxels* pour lesquels il existe une estimée de normale à la surface. Dans la version courante, ces estimées sont prise en compte uniquement pour les *voxels* marqués *occupé*, *ocplane* ou *frontière*; nous ne propageons pas ces estimées sur les *voxels* marqués *inconnu* connexe aux frontières des zones connues.

Les fonctions d'utilité sur les types *occupé*, *ocplane* ou *frontière* sont donc pondérées par des coefficients de qualité  $K_{occup}$ ,  $K_{ocplane}$  et  $K_{frontiere}$ . Donc la fonction à minimiser est :

$$F(x) = -K_{occup}f_{occupe}(x) - K_{ocplane}f_{ocplane}(x) - f_{inconnu}(x) - K_{frontiere}f_{frontiere}(x) \quad (4.9)$$

Les coefficients de qualité dépendent de l'angle d'incidence entre le rayon (laser ou optique) et la surface. La qualité est maximale si le rayon est orthogonal à la surface. Donc pour chaque type considéré de *voxels*, ce coefficient est :

$$K = \frac{\sum_{i=1, np} \cos(\delta_i)}{np}$$

où  $np$  est le nombre de *voxels* de ce type qui sont perçus depuis le placement évalué, et  $\delta$  est l'angle entre le rayon et l'inverse de la normale (0 pour l'optimum).

Pour vérifier le fonctionnement de la méthode, nous présentons en figure 4.5, le résultat de la planification pour une simulation sur un environnement simple (un cube) perçu avec un capteur doté simplement de deux degrés de liberté (site et azimut).

A chaque prise de vue, la technique doit trouver le minimum de la fonction de qualité. Il faut utiliser une méthode d'optimisation sur la fonction globale d'utilité, en évitant les inconvénients connus de minima locaux qui empêchent la méthode de progresser jusqu'à trouver le minimum global. Dans la version initiale, JM.Sanchiz avait proposé d'exploiter une méthode de type *Hill Climbing* en évaluant la fonction d'utilité pour un nombre discret de placements (4 positions  $(x, y, z)$ , 36 orientations  $(\theta, \phi)$  par position) : les 4 positions sont d'une part la position courante du capteur et d'autre part, trois positions tirées de manière aléatoire dans l'espace atteignable; l'espace atteignable est donné par la zone du sol perçue sans obstacle.

Cette méthode est très lente et ne converge pas toujours. Nous avons choisi d'utiliser l'algorithme de Levenberg-Marquardt pour minimiser la fonction d'utilité. Dans l'exemple illustré en figure 4.5, nous appliquons une stratégie illustrée en figure 4.6 pour sélectionner les conditions initiales sur une spirale

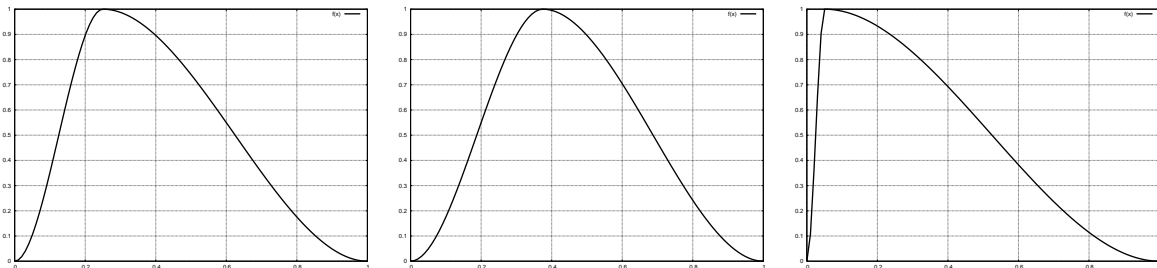


FIG. 4.4 – Fonctions des types de *voxels* marqués *occupé* (gauche), *inconnu* et *ocplane* (centre), *frontière* (droite).

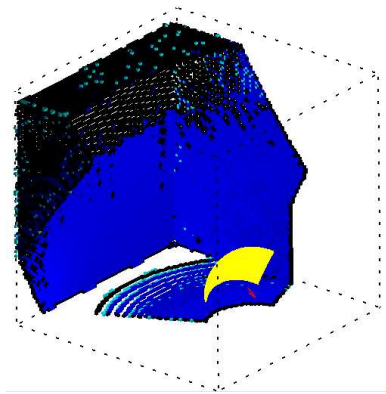


FIG. 4.5 – Environnement de test et position du capteur.

dans l'espace  $(\theta, \phi)$  des orientations, afin de trouver le meilleur minimum pour démarrer l'optimisation.

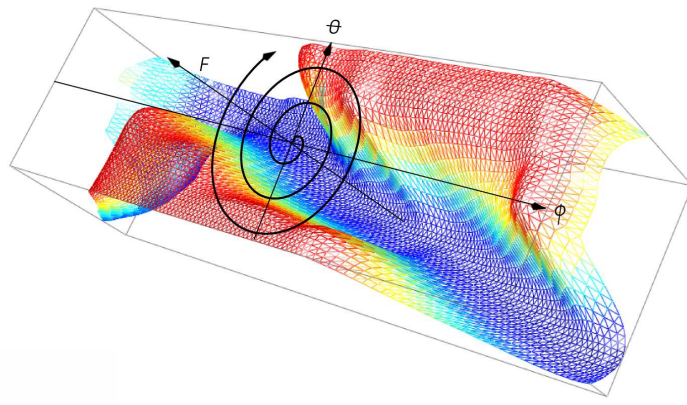


FIG. 4.6 – Stratégie d'optimisation.

La figure 4.7 montre la progression de la fonction d'optimisation des *voxels* pour l'environnement vu sur la figure 4.5 ; la couleur bleue dénote la valeur minimale de la fonction. . La méthode a pris 14 vues pour couvrir la plupart de l'environnement ; à la fin la fonction reste avec une quantité réduite de minima locaux acceptables, éloignés de la valeur attendue. Ceci pose une difficulté à la méthode d'optimisation et est une condition d'arrêt du processus ; de ce fait, le résultat est "sous-optimal".

En figure 4.8, nous présentons l'état de la répartition des données sensorielles dans la *voxel map* après six vues ; chaque acquisition a une couleur ; les régions communes entre vues, nécessaires pour le processus de recalage, sont visibles.

#### 4.4 Description de l'algorithme.

La figure 4.9 montre le fonctionnement de la méthode exécutée jusqu'à marquer tous les *voxels* de l'espace à explorer. Les conditions de fin de la modélisation sont toujours problématiques, car le but de n'avoir aucun *voxel* marqué *inconnu* ne peut pas toujours être assuré. La boucle interne minimise la fonction d'utilité des placements possibles du capteur en fonction de l'état courant des *voxels* et donne le placement idéal pour acquérir l'image suivante. C'est dans cette partie que nous utilisons la stratégie



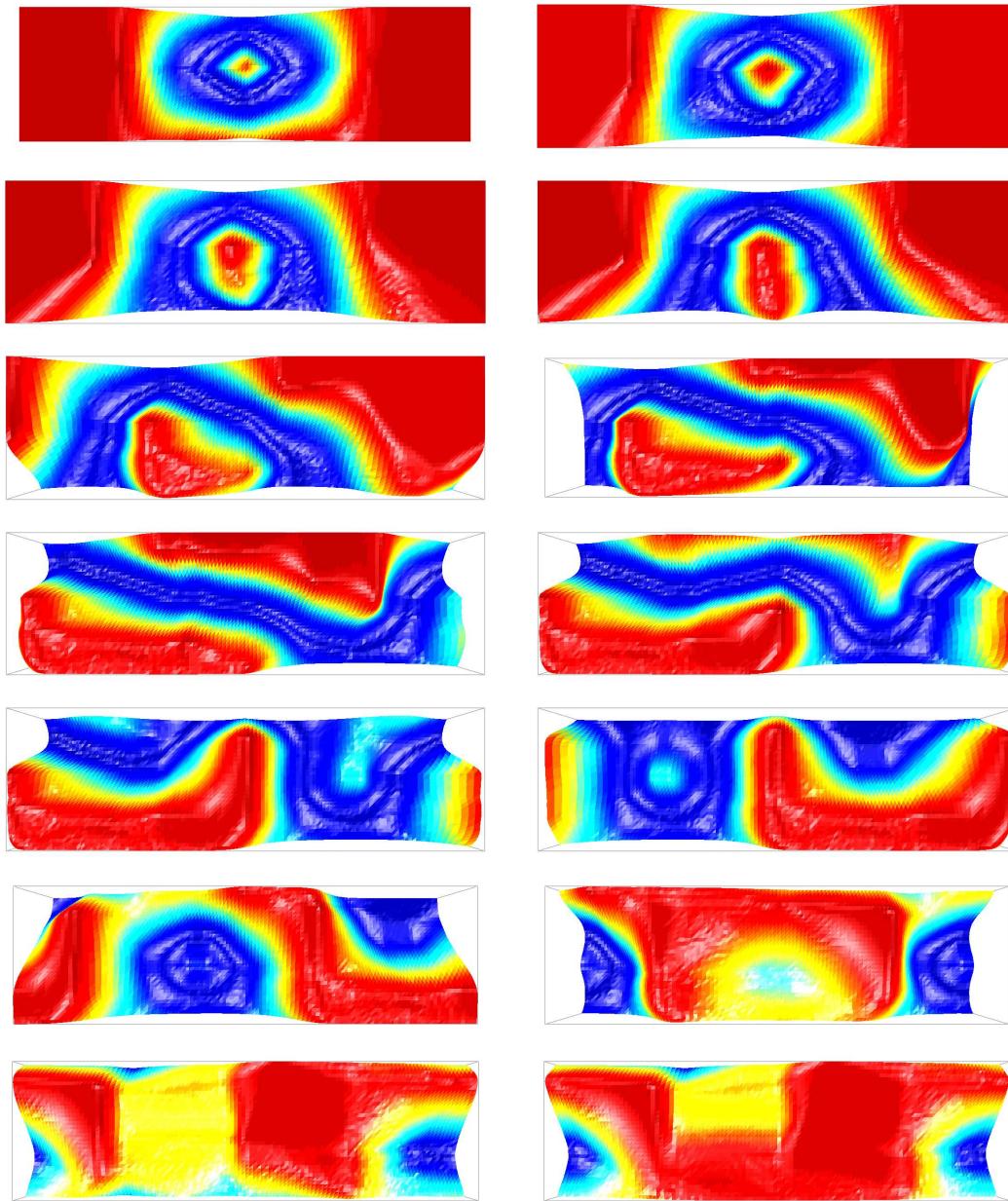


FIG. 4.7 – Progression de la fonction d'optimisation pour  $\phi$  et  $\theta$  à chaque prise de vue.

d'optimisation montrée sur la figure 4.6 pour sélectionner d'abord, une estimée initiale de l'orientation. Les boîtes vertes indiquent les manipulations sur le capteur.

L'algorithme de *Next Best View* a été évalué à l'aide d'une simulation complète de l'environnement et du capteur laser. Le plan de l'environnement à explorer est présenté en figure 4.10 : cet environnement est complexe car il est constitué de deux parties connectées par un passage central plus étroit ; dans la partie gauche, le grand cercle correspond à une calotte sphérique (plafond en forme de dôme), tandis que les petits cercles correspondent à des colonnes. La position et l'orientation initiale du capteur sont représentées par la flèche dans la partie centrale.

Deux étapes du processus d'exploration de cet environnement, sont montrées sur la figure 4.11. Dans

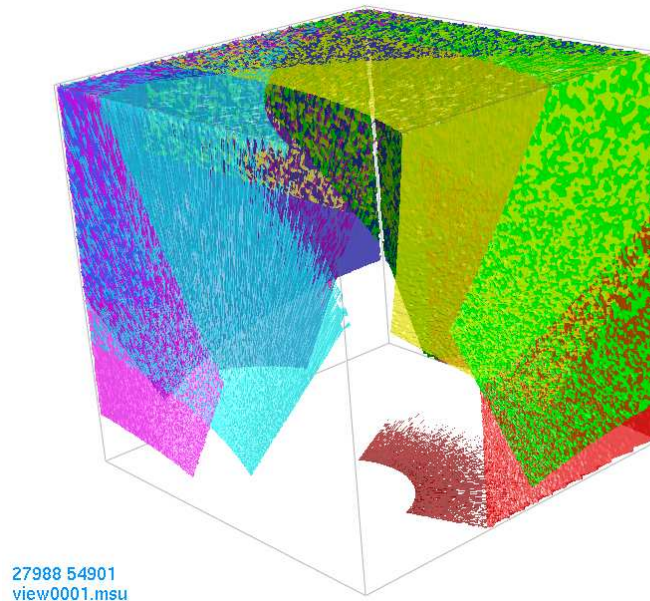


FIG. 4.8 – Données du capteur pour les six premières vues.

chaque figure, nous présentons à droite le plan de l'environnement avec une flèche pour le dernier placement sélectionné pour le capteur, et à gauche l'état du *voxel map*. En bleu sombre et bleu clair apparaissent les *voxels* marqués respectivement *occupé* et *ocplane*.

Globalement notre méthode parvient à trouver un ensemble de positions pour explorer tout cet environnement, mais les temps d'exécution sont très longs et la convergence vers un optimum n'est pas garantie. De plus, dans le cas de l'exploration d'un environnement par un robot mobile, des problématiques additionnelles s'ajoutent par rapport à la planification perceptuelle : par exemple

- la planification de la trajectoire du robot en évitant des manoeuvres complexes (notamment pour un robot non holonome),
- la minimisation de la distance de déplacement. Par exemple, sans prendre en compte cette contrainte, nous avons eu des cas où le robot faisait des acquisitions alternativement dans la partie droite, puis la partie gauche de l'environnement.
- la détection d'un échec de la méthode d'optimisation, qui survient en général sur une partie de l'environnement en fin d'exploration. Par exemple, dans notre environnement simulé, dans certains tests, le robot explore complètement la partie gauche, mais préfère affiner le modèle plutôt que d'aller explorer la partie droite.

Des solutions ad hoc ont été rajoutées pour traiter certaines situations ; par exemple, si l'échec de l'optimisation est détecté, le robot retourne sur ses pas en exécutant en sens contraire, la trajectoire déjà effectuée (*back tracking*) pour relancer l'optimisation, à partir des placements précédents.

## 4.5 Adaptation au cas de la modélisation des objets.

La méthode présentée de *Next Best View* est conçue pour l'exploration des environnements, mais elle peut s'adapter à notre application spécifique, qui est la modélisation d'un objet. Les problèmes spécifiques liés à la mobilité, ne sont pas si importants pour cette application, si nous faisons certaines hypothèses réalistes sur le scénario robotique :

- le capteur est porté sur un bras manipulateur, monté lui-même sur une plateforme mobile.
- la tâche de modélisation commence quand le robot a détecté un objet inconnu sur une table ; la

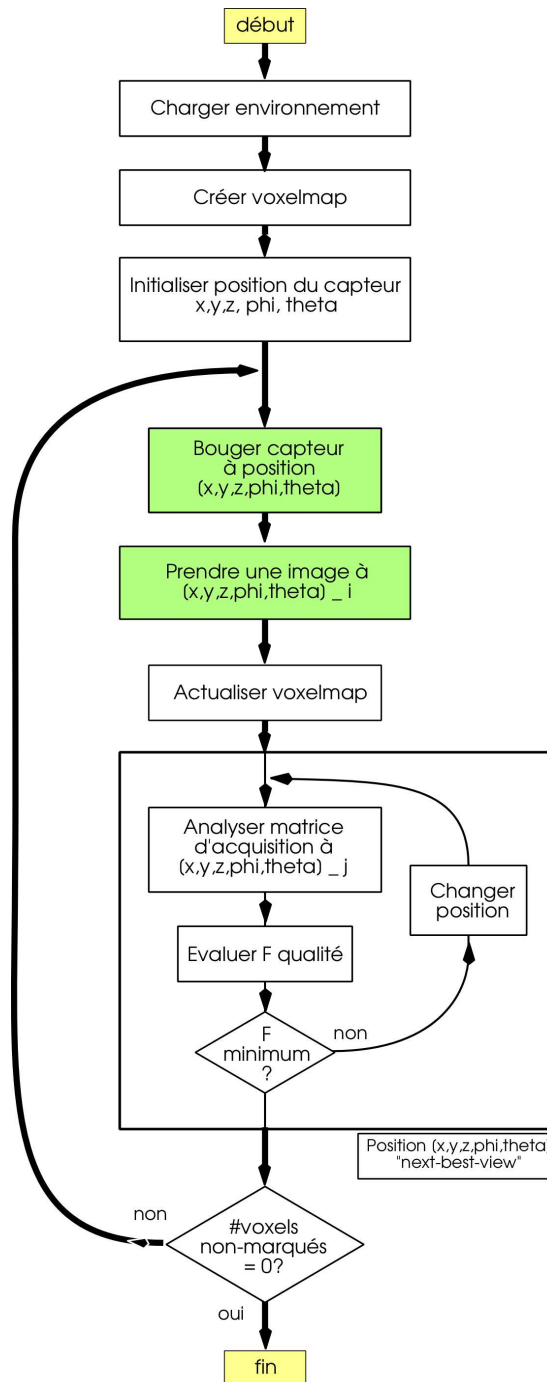


FIG. 4.9 – Le diagramme de l’algorithme de Next Best View.

table est déjà localisée, et nous supposons qu’il existe des fonctions pour que la plateforme accoste le long de cette table.

- le robot doit construire un modèle de l’objet, en déplaçant le capteur par le bras manipulateur, et si besoin, en accostant la table en plusieurs positions.

On suppose dans ce cas que la *voxel map* à explorer est centrée sur la table, autour de la position dans laquelle l’objet inconnu a été détecté. Le trajet d’exploration reste dans les voxels marqués *vide*, sur la

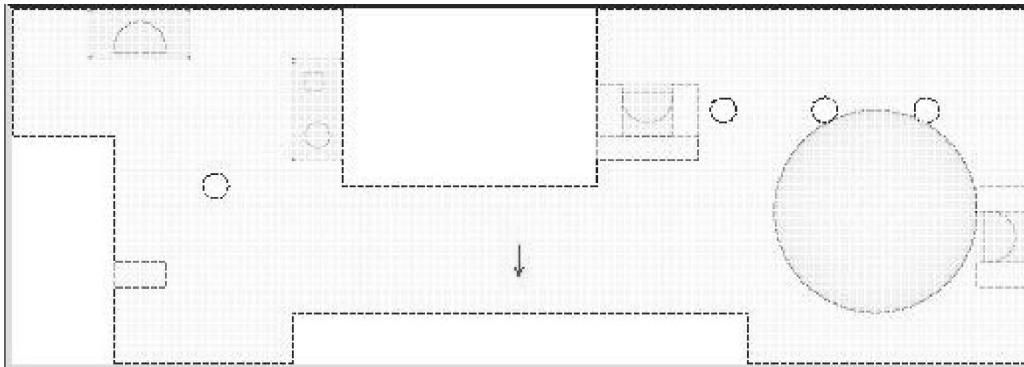


FIG. 4.10 – Plan de l'environnement à explorer.

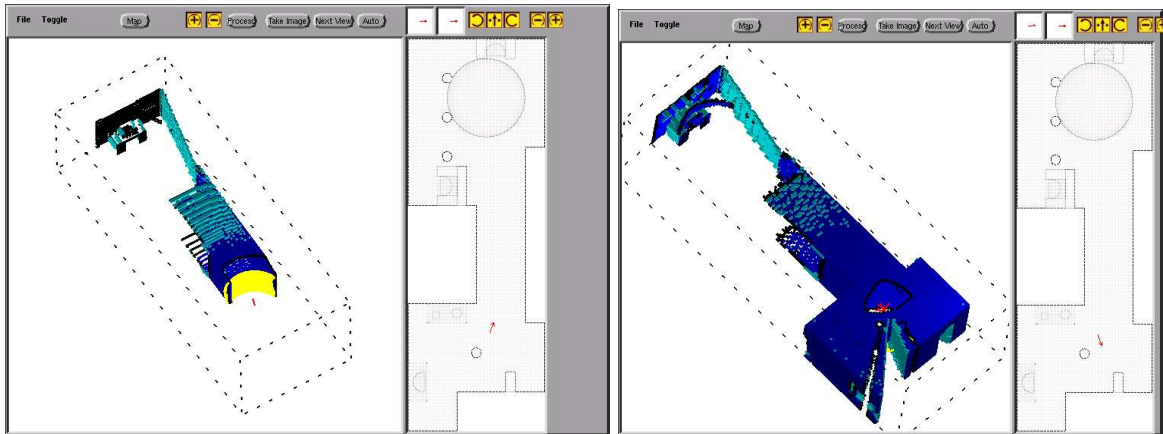


FIG. 4.11 – Les étapes du processus de Next Best View dans un environnement navigable.

périphérie du *voxel map*. Les *voxels* marqués *inconnu* initialement sont seulement ceux autour de l'objet à modéliser.

Afin d'accélérer le processus de sélection des vues traitées, qui peut prendre beaucoup d'itérations surtout au cours des dernières itérations, nous avons décidé de simplifier le trajet du capteur : comme dans les travaux de R.Pito ou MA.Garcia, notre capteur se déplacera sur une ligne portée sur une sphère centrée sur la position de l'objet à modéliser, la sphère d'observation. Le rayon de la sphère sera choisie en fonction des contraintes du capteur (distance minimale de perception) et du volume estimatif de l'objet à modéliser. Les courbes décrites par le capteur sur la sphère d'observation seront des cercles.

Dans la pratique les trajectoires ainsi planifiées seront exécutées uniquement par le bras manipulateur, la plateforme restant en position statique, accostée le long de la table. De ce fait la fonction d'utilité a moins de degrés de liberté et est optimisée plus vite.

Une sphère de points d'acquisition pour un bras manipulateur est montré sur les figures 4.13 a,b,c,d. La figure 4.13a montre les possibilités d'orientation du capteur sur le bras à différentes distances de l'objet. La figure 4.13b illustre la position idéale pour placer l'objet à acquérir, jusqu'au-dessus du robot, car le bras peut atteindre presque tous les angles de vue. Les figures 4.13c) et d) présentent des positions d'acquisition proches de la position "home" du bras, au-dessous de la base.

Généralement, le bras seul ne suffit pas pour atteindre toutes les positions autour de l'objet, et ainsi en construire un modèle complet (sauf la partie en contact avec la table), car le volume de travail d'un

bras embarqué sur une plateforme, est assez réduit. Afin d'acquérir des vues tout autour de l'objet, le robot mobile qui porte le bras devra se déplacer vers une autre position autour de la table (figure 4.12) ; pour réduire la complexité du problème, ces déplacements ne sont pas pris en compte dans la version dédiée *Modélisation d'Objet* de l'algorithme de *Next Best View* qui calcule seulement les positions locales du bras sur une sphère centrée sur l'objet.

Notons que, pour vérifier que les positions sur la sphère d'observation sont atteignables, nous avons dû intégrer dans notre planificateur, le modèle géométrique inverse du bras manipulateur et prendre en compte explicitement la position du capteur sur le bras, obtenu par la calibration main-oeil (ou *Hand-Eye calibration*). La trajectoire du bras pour déplacer le capteur sur une ligne sur la sphère d'observation, est donc directement donnée en coordonnées articulaires. Le bras est commandé par une commande en position.

## 4.6 Conclusion sur la méthode NBV

Ce chapitre 4 a décrit une méthode de planification perceptuelle, adaptée à une tâche de modélisation. Tout d'abord, nous avons poursuivi des travaux entamés par ailleurs, sur une méthode qui recherche le prochain meilleur choix de vue pour la construction autonome et incrémentale du modèle 3D d'un environnement. Nous avons évalué nos contributions sur un environnement et un robot simulé ; la tâche est accomplie à l'aide d'un capteur 3D monté sur un robot mobile avec 5 degrés de liberté. Nos contributions ont porté sur la fonction d'utilité d'un placement potentiel du capteur, et sur la méthode utilisée pour optimiser cette fonction : le problème est très complexe, et nous ne sommes pas parvenus à une solution optimale et robuste, même dans cet environnement de simulation.

Puis, nous avons simplifié la problématique pour traiter uniquement de la modélisation d'un objet, avec un scénario assez contraint. Dans ce cas, et en reprenant le concept de sphère d'observation proposé par plusieurs auteurs, nous sommes parvenus à une solution pratique efficace pour déplacer un capteur porté par un bras manipulateur, le long d'une courbe sélectionnée sur la sphère d'observation.

Notre ambition étant de proposer un système automatique de modélisation d'objets, il convient donc

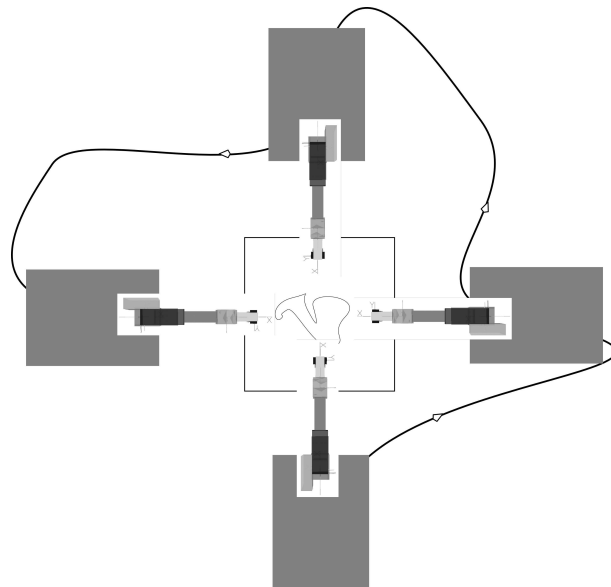


FIG. 4.12 – Positions du robot mobile pour l'acquisition de l'objet.

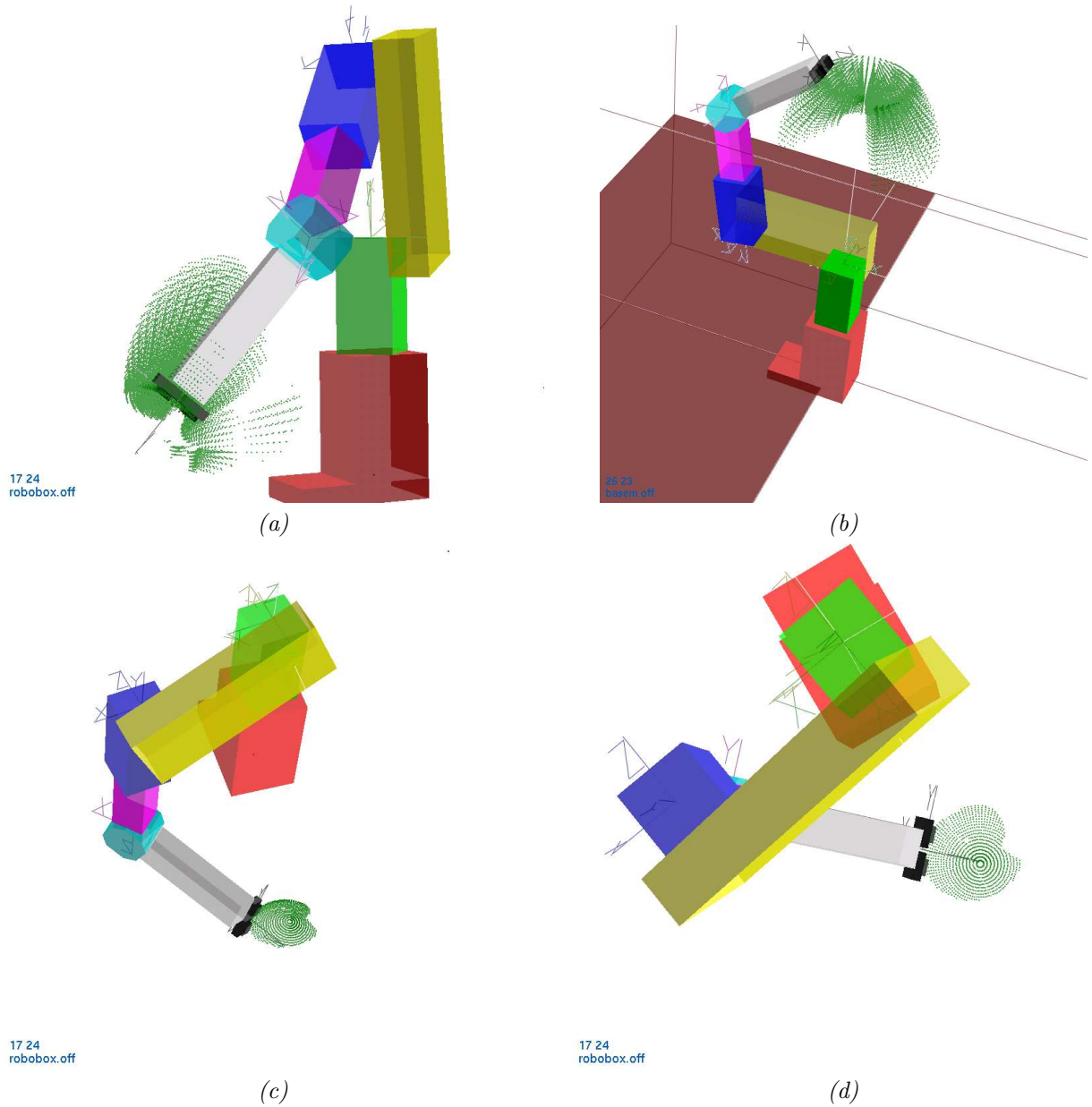


FIG. 4.13 – Exemple des positions d'un bras manipulateur GT6a autour de l'objet à modéliser.

d'intégrer cette méthode de planification, ainsi que les fonctions de recalage ICP et de construction de maillage BPA, sur un système robotique.



# Chapitre 5

## Résultats expérimentaux

### 5.1 Introduction.

Nous avons montré dans les chapitres précédents de nombreux résultats, obtenus surtout à des fins d'évaluation des algorithmes que nous avons développés (ICP à partir de points de contour ou de points de maillage) , ou de comparaison de leurs résultats avec ceux obtenus par d'autres méthodes (BPA ou Surfaces Déformables).

Nous allons présenter dans cette section, des résultats obtenus dans le contexte des applications robotiques de ces méthodes :

- pour ICP : application à la navigation des robots mobiles.
- pour BPA : application à la modélisation d'objets 3D et pour différents types d'images.
- Intégration : utilisation des méthodes sur des images acquises depuis un robot démonstrateur (bras mobile) disponible dans notre laboratoire.

### 5.2 Application de la méthode d'ICP pour la navigation d'un robot mobile.

Nous avons utilisé notre méthode d'ICP sur des données acquises avec un capteur laser 2D, par un robot mobile qui se déplace dans un environnement d'intérieur. Le problème de la navigation est très classique : en principe, un robot exécute des trajectoires; il contrôle cette exécution en se localisant périodiquement. Cette localisation peut être obtenue de plusieurs manières, mais on peut différencier plusieurs classes de méthodes :

- **localisation par intégration des estimations de déplacement élémentaire.**

Ces méthodes exploitent généralement, des capteurs proprioceptifs présents sur le robot, typiquement des odomètres pour un robot mobile. La position du robot peut donc être donnée seulement par intégration des mesures odométriques des roues : cette information des capteurs se dégrade progressivement le long de la trajectoire du fait des dérapages des roues ou autres raisons.

Ces techniques ne sont exploitables que pour prédire la position du robot entre deux localisations ou recalages, fondées sur les méthodes décrites ci-après. En exploitant uniquement l'odométrie, un robot mobile n'est plus localisé correctement après quelques mètres.

- **localisation par intégration des estimations de déplacement.**

Ces méthodes exploitent les données sensorielles acquises sur l'environnement, afin d'estimer les déplacements. Il n'existe pas de modèle : la fonction de localisation intègre les déplacements mesurés entre les vues. Nous avons déjà évoqué les travaux de notre groupe sur l'odométrie optique [Mallet 00] qui exploite la stéréovision pour mesurer les déplacements d'un robot en milieu naturel. Nous proposons ci-dessous d'exploiter l'algorithme ICP pour ce même traitement en milieu intérieur.



Ces techniques procèdent toujours par intégration de mesures sensorielles. Même si elles ne souffrent pas des mêmes dérives que les techniques fondées sur l’odométrie, toute erreur entre deux vues sera répercutée sur la position du robot ; après quelques dizaines de mètres, le cumul des erreurs rend la localisation très incertaine, si ce n’est imprécise.

– **localisation sur amers.**

Ces méthodes exploitent un modèle qui a du être appris dans une phase préalable d’exploration par une méthode de type SLAM évoquée dans le chapitre 1. Si le modèle est correct (et c’est là le problème du SLAM), alors, la localisation sur amers permet de maintenir bornée l’erreur de localisation.

Nous avons souhaité tester notre méthode ICP sur des données acquises par un capteur laser 2D, pour obtenir un profil ou une coupe de l’environnement. Une telle méthode est exploitée par l’équipe de J. Hertzberg (Fraunhofer Institut de Sankt Augustin). A partir de la position estimée approximativement par les odomètres et des positions antérieures, la méthode d’ICP est appliquée pour corriger la position réelle du robot en recalant deux profils successifs acquis par le robot.

Le traitement traditionnel pour calculer la position du robot à partir de ces profils laser, consiste en trois étapes ; (1) segmentation des profils pour obtenir des segments, (2) recherche d’appariements entre les segments extraits des profils successifs et (3) estimation de la transformation entre les segments appariés, donc entre les profils successifs. Cette transformation, moyennant quelques corrections (liées à la position du capteur laser sur le robot), correspond au déplacement du robot.

Le transformation est estimée par minimisation d’une erreur d’alignement entre les segments appariés ; une technique d’optimisation peut être requise à ce niveau (filtrage de Kalman étendu généralement) ; l’estimée initiale est fournie par l’odométrie.

Avec la méthode ICP, nous évitons la partie de segmentation, apparions directement les points des profils successifs et recalons directement ces ensembles de points. Dans ce cas spécifique, les calculs sont plus simples, car la transformation entre les deux vues est 2D, donc se compose seulement d’un angle et d’une translation 2D, trois inconnues contre six dans le cas général évoqué en section 2.6.

Nous avons essayé deux approches :

– **recalage avec fusion incrémentale :**

chaque nouvelle acquisition est recalée avec l’ensemble des données acquises et fusionnées par les recalages précédents. Cette approche serait utile pour générer une carte de navigation complète, donc en phase d’acquisition du modèle de l’environnement. Elle pourrait se substituer au SLAM classique, mais avec l’inconvénient d’avoir à chaque pas plus de données et par conséquent un temps de calcul plus grand. La carte de l’environnement est globale, et le problème classique de la boucle (robot qui fait le tour d’un bâtiment) peut être traité.

– **recalage avec une fenêtre d’acquisition de taille  $n$  :**

c’est la même situation que l’approche antérieure, mais à chaque acquisition, les données les plus anciennes sont éliminées du modèle fusionné, donc la quantité de points à recalcr est constant et la carte de l’environnement devient locale.

Profils recalés	Points		Erreur d’ICP	Itér.	Temps CPU [ms]	Correction à la position	
	1	2				$\vec{x}$	$\theta$
1 + 2	345	349	0.00614317	005	0.00	[-0.0983, +0.0232]	0.57296
1, 2 + 3	694	350	0.00560500	006	0.01	[-0.1857, +0.0237]	0.95875
1, 2, 3 + 4	1044	349	0.00379583	017	0.04	[-0.2677, +0.0588]	1.53745
1, 2, 3, 4 + 5	1047	352	0.00313381	009	0.02	[-0.2922, +0.0683]	1.82996
2, 3, 4 + 5	1393	325	0.00398105	009	0.02	[-0.2980, +0.0698]	1.84781
1, 2, 3, 4, 5 + 6	1745	350	0.00325502	021	0.10	[-0.3779, +0.1032]	2.20435
3, 4, 5 + 6	1049	350	0.00519497	026	0.11	[-0.4212, +0.1243]	2.36253
1, 2, 3, 4, 5, 6 + 7	2095	355	0.00477590	041	0.14	[-0.8378, +0.5051]	3.67829
4, 5, 6 + 7	1048	355	0.00748515	038	0.13	[-1.3137, +0.8327]	5.83432

TAB. 5.1 – Résultats du recalage des profils successifs avec les deux approches.

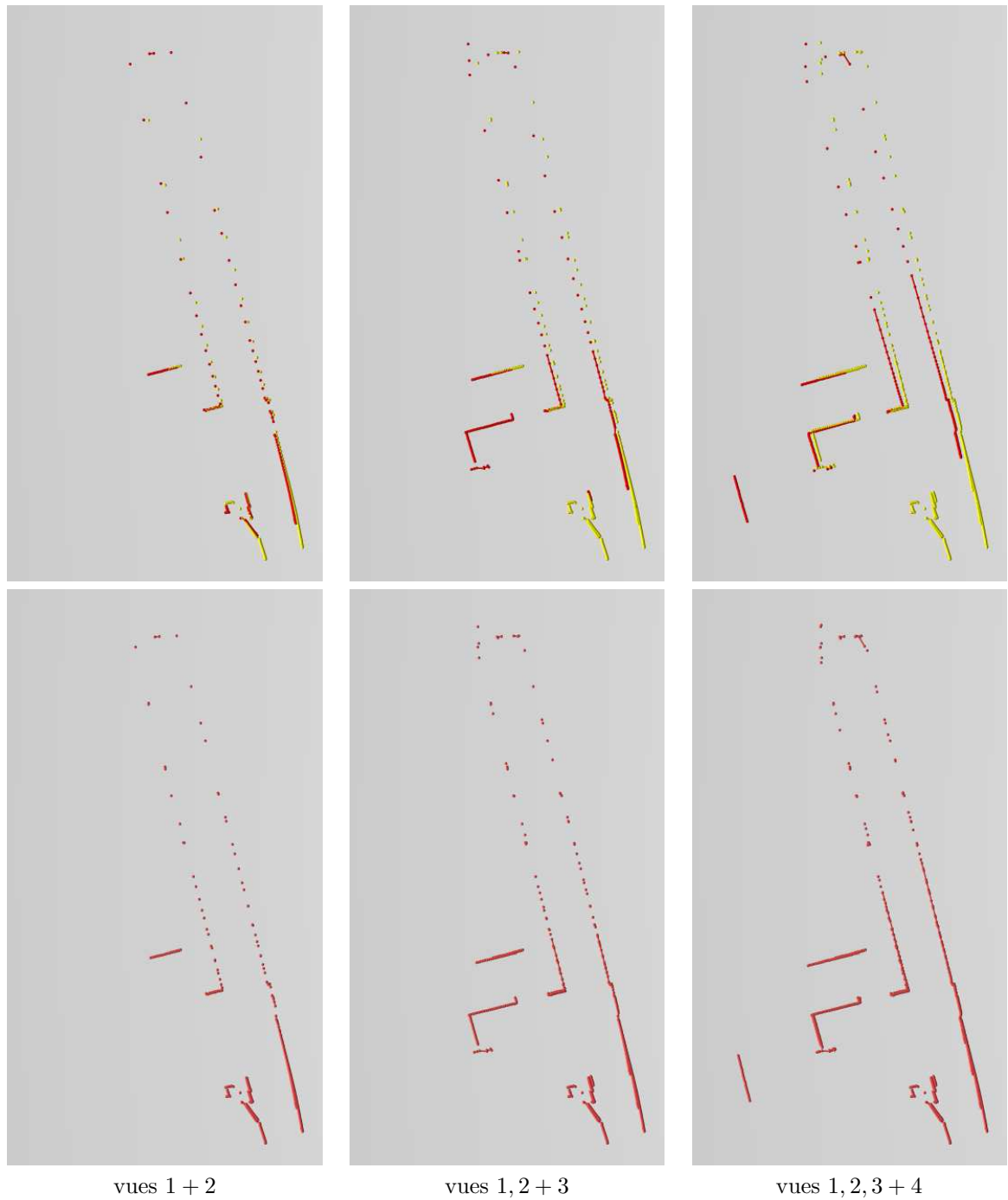


FIG. 5.1 – Recalage des 3 premiers profils, égal pour les deux approches.

Lors des essais présentés sur les figures 5.1, 5.2 et 5.3, nous avons réussi à corriger la position finale du robot mobile. Les résultats numériques sont présentés en tableau 5.1. Par contre les conditions d'exécution pour arriver aux conditions de succès ont été :

- L'ICP a été réglé à un recalage fin exclusivement.
- La position initiale est donnée par les odomètres du robot. En cas de non convergence, l'acquisition est rejetée.
- L'utilisation d'espaces de *clusters* plus denses est avantageux pour améliorer la convergence.

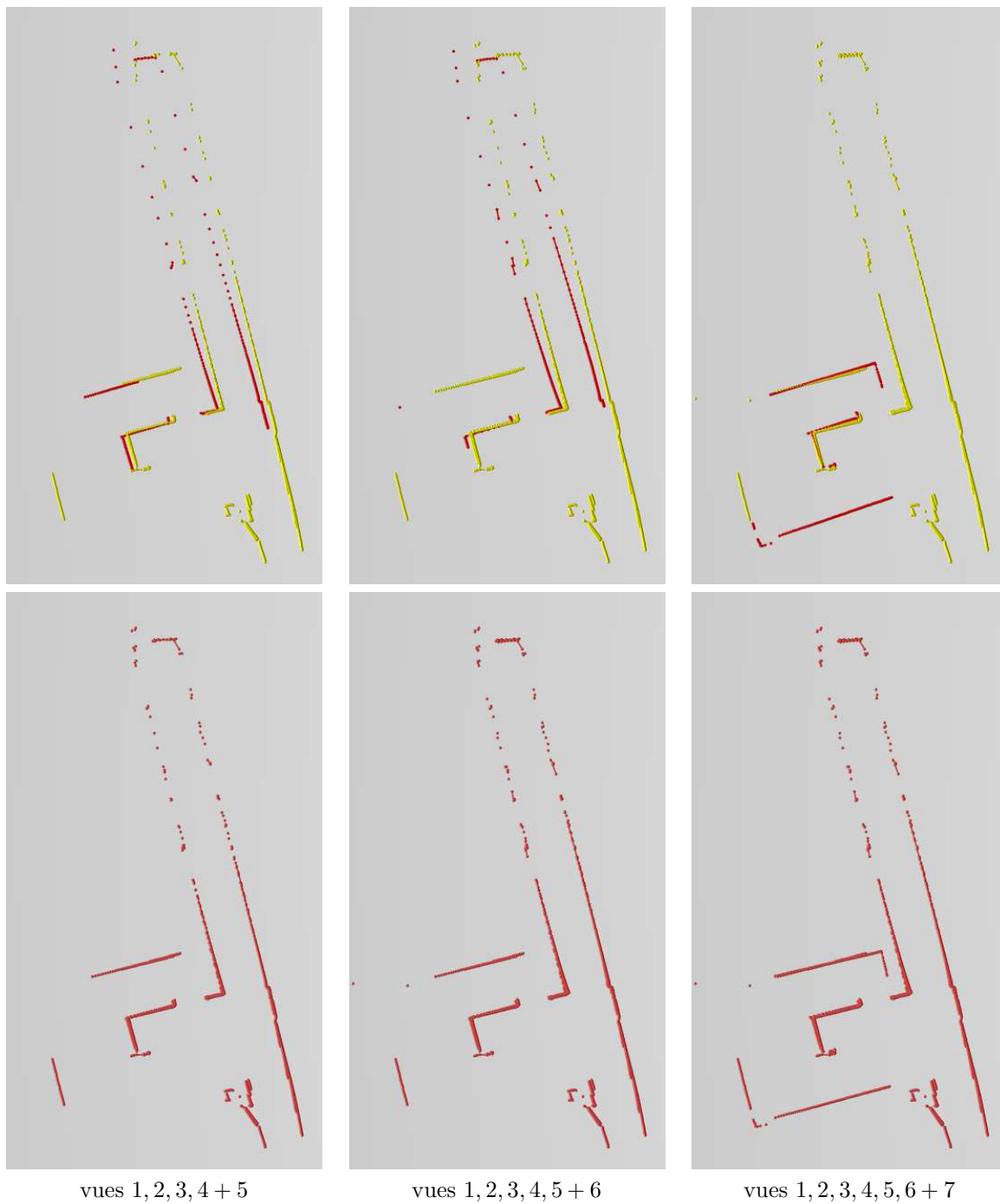


FIG. 5.2 – Recalage avec approche 1 : recalage cumulatif.

Nous notons que les temps de calcul obtenus permettraient d'exécuter ce traitement à bord, en temps réel, à plus de 10Hz. Mais une réécriture des algorithmes d'ICP est préférable pour adapter l'algorithme au cas bidimensionnel, afin de simplifier beaucoup de sous-routines.

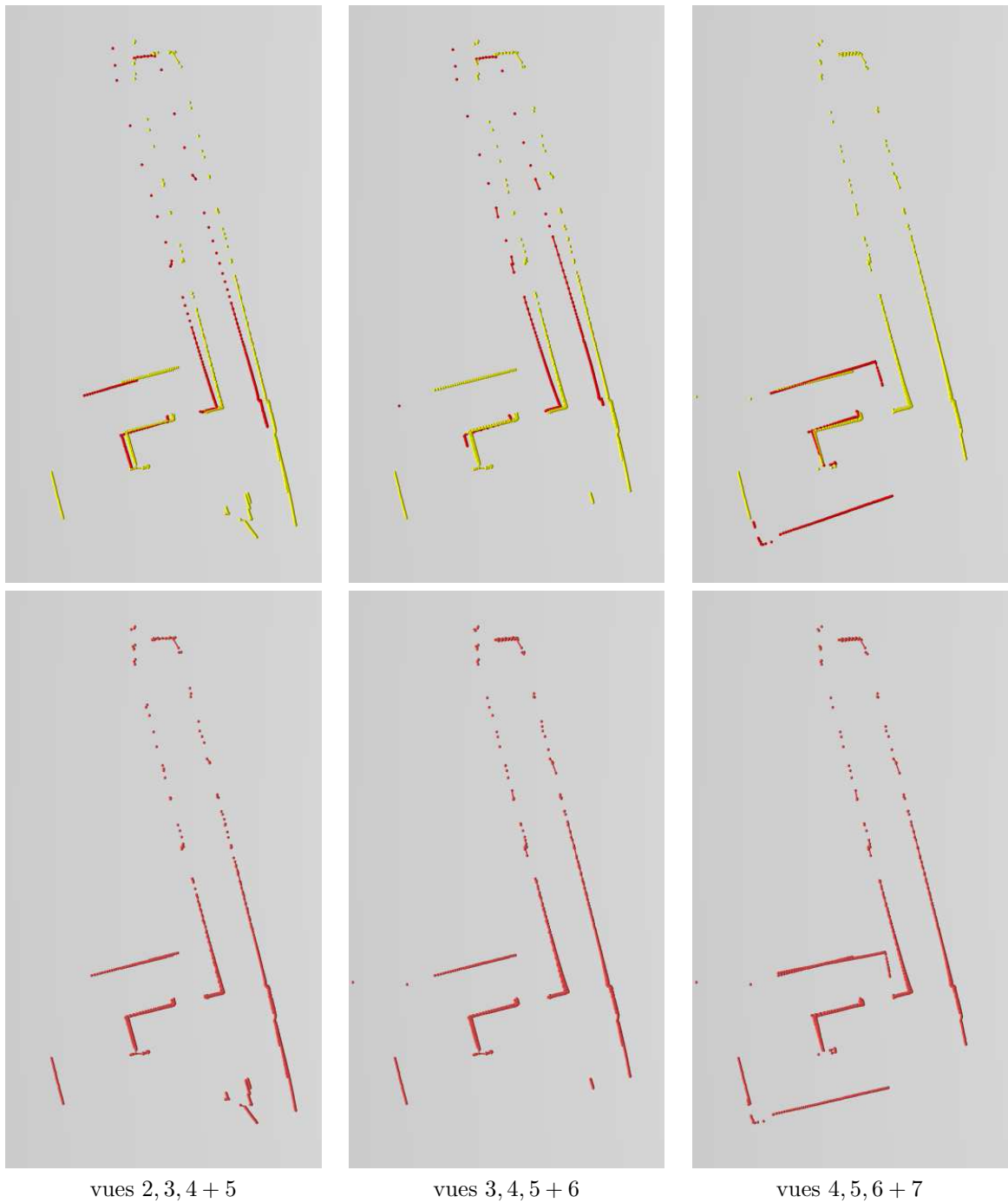


FIG. 5.3 – Recalage avec approche 2 en utilisant uniquement une fenêtre de 3 profils.

### 5.3 Résultats de BPA.

L'algorithme de la boule pivotante ou BPA que nous avons décrit ci-dessus, permet principalement de construire un maillage à partir d'un nuage de points résultant d'une phase de recalage, typiquement d'un processus ICP.

### 5.3.1 Exemples.

La méthode BPA a été appliquée avec des données de sources diverses décrites ci-dessous :

1. Surfaces synthétiques : surfaces planes, sphères et intersections de telles primitives, avec ou sans maillage local pour chacune des vues.
2. Images de capteur laser : séquences “bunny”, “dragon”, “curvblock”.
3. Images de profilométrie : adaptées à la modélisation corporelle. Ce sont aussi des nuages de points purs (sans maillage local).
4. Images de stéréo-reconstruction : Remaillage des images à partir de maillages locaux existants.
5. Images post-ICP : avec les images de capteur laser et stéréo-reconstruction qui ont besoin d’un remaillage après l’étape de recalage.

#### Maillage après un recalage par ICP

Les résultats d’un processus de recalage avec ICP ont été utilisés pour évaluer le comportement de l’algorithme BPA en produisant un nouveau maillage après ICP.

Nous montrons ici un exemple synthétique et un réel. Sur la figure 5.4, on a utilisé les vues synthétiques d’un capteur laser sur un objet polyédrique ; à gauche le nuage de points 3D (9 vues recalées par ICP), et à droite, construction d’un maillage et segmentation, avec les facettes illustrées avec une couleur différente. Chacune des neuf vues qui forment l’objet entier a été décimée pour réduire le nombre de points. Le nuage complet a été obtenu avec ICP en recalant les images deux à deux, ce qui a pour effet d’introduire encore plus d’erreurs dans le processus de recalage : un recalage incrémental serait préférable (d’une vue avec le nuage fusion des vues précédentes).

Le résultat final d’ICP donne un nuage de points non distribués régulièrement avec des polygones en chevauchements multiples pour chaque vue. La méthode BPA a détecté correctement chaque surface de l’objet et a pris seulement les points nécessaires pour produire un nouveau maillage.

La figure 5.5 montre un nuage constitué de deux images 3D, qui proviennent de deux reconstructions par stéréo dense : à gauche, 2 vues recalées par ICP, acquises sur une scène par stéréo dense, à droite le maillage construit par BPA, avec deux facettes planes. Dans cet exemple, les facettes correspondant au sol et à la mire plane de calibrage stéréo qui était posé sur une chaise, ont été extraites correctement. Les pieds de la chaise apparaissent aussi : par contre, les composantes de maillage construites pour l’assise de la chaise, avaient un nombre de sommets trop faibles et ont été filtrés.

Ces images ont été décimées avant recalage par ICP. La méthode BPA est exploitée pour produire un nouveau maillage et pour faire simultanément, une segmentation en primitives surfaciques. La distribution de points de ce type d’image est localement dense, mais est non régulière, en particulier sur le sol. De ce fait le rayon de la boule doit être choisi plus grand que la distance entre la plupart des couples de points voisins pour réduire le nombre de trous.

#### Images obtenues par un profilomètre

L’algorithme BPA a aussi été évalué sur des nuages de points acquis par profilométrie pour la modélisation du corps humain : nous montrons sur la figure 5.6, des vues partielles sur le résultat obtenu par BPA appliqué à deux nuages acquis sur un buste, avec bras vers le bas ou vers le haut.

La méthode d’acquisition du profilomètre fournit un ensemble de profils horizontaux, avec un arrangement horizontal annulaire des points dans lesquels les distances verticales et horizontales entre points sont différentes.

Il n’y a aucun maillage local disponible, donc le triangle-souche du BPA doit être cherché directement dans le nuage de points. La précision ici est élevée (erreur inférieure au mm) ; les nuages sont obtenus par fusion d’un grand nombre de profils (200 images en tout) acquises par 4 caméras disposées autour de la personne. Le recalage des profils est garanti par un étalonnage supposé parfait du dispositif qui supporte

les caméras et les lasers.

Par rapport aux cas précédents, la difficulté réside ici dans la complexité de la surface à modéliser. En particulier les repliements sous les bras, rendent complexes la construction du maillage. Dans la comparaison décrite dans [Lagarde 04], plusieurs méthodes ont été comparées afin de traiter ce problème :

- L’algorithme des *Marching Cubes* (en version originale ou simplifiée) donne des bons résultats, mais le modèle obtenu peut présenter des ouvertures ce qui peut faire échouer des traitements exploitant ce maillage (calcul de volume).
- La construction du maillage à partir d’une surface déformable, produit par définition, un modèle fermé, mais cette méthode requiert un opérateur expert afin de régler les paramètres et traiter les cas complexes tels que les repliements.

Le BPA, moyennant un choix adéquat du rayon de la boule, n’est pas perturbé par ces repliements. Pour réduire le temps de calcul, nous évaluons comment l’utiliser localement, sur le maillage produit par l’algorithme des *Marching Cubes* pour boucher les ouvertures d’un maillage.

Figure	Points	Polygones initiaux	Rayon boule	Polygones finaux	Segments	temps CPU [s]
5.5	8561	15998	0.31	5250	6	138.00
5.4	6362	10838	0.3	1919	9	86.00
5.6	9356	0	7	4247	24	76.00
5.6	5952	0	7	3428	16	64.00
3.11	480	756	0.47	348	2	2.00
3.10	320	504	0.47	135	1	0.92

TAB. 5.2 – Résultats du processus BPA sur des données de différentes sources

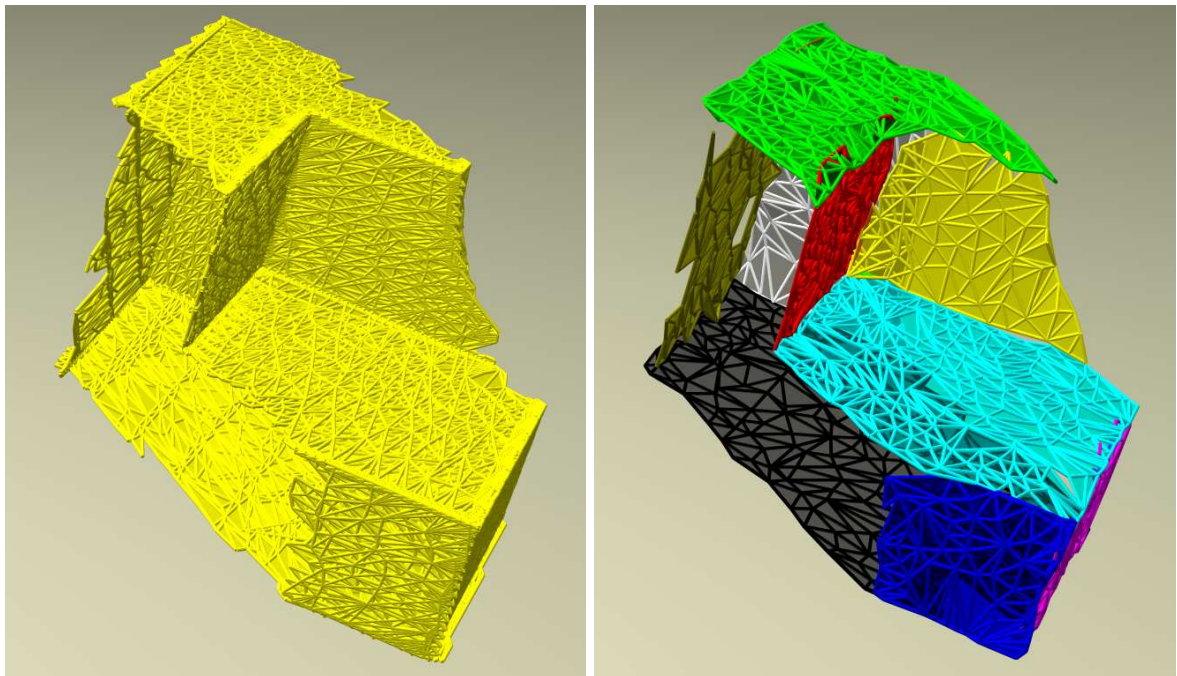


FIG. 5.4 – Résultat de BPA sur des images synthétiques d’un objet polyédrique.

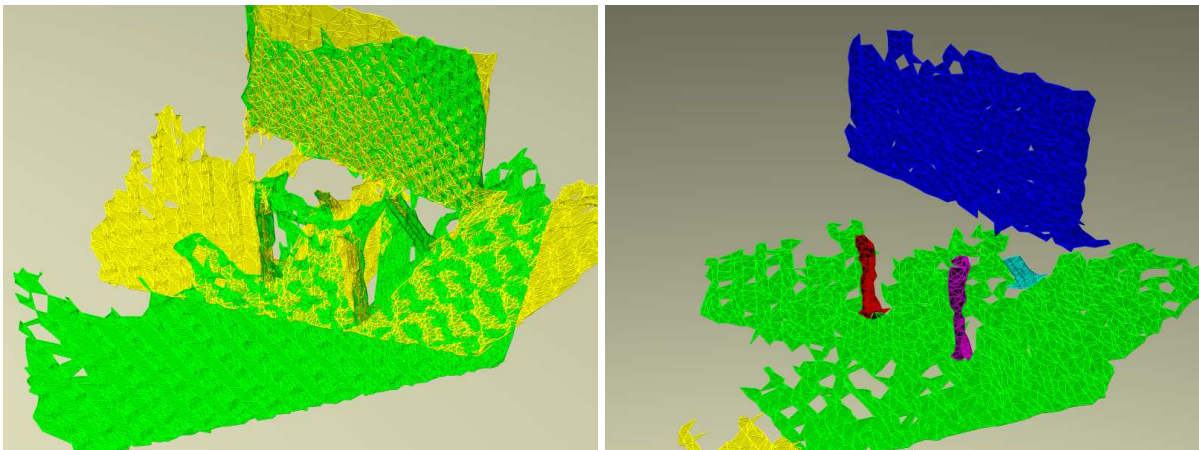


FIG. 5.5 – Résultat de BPA sur des images acquises par stéréovision.

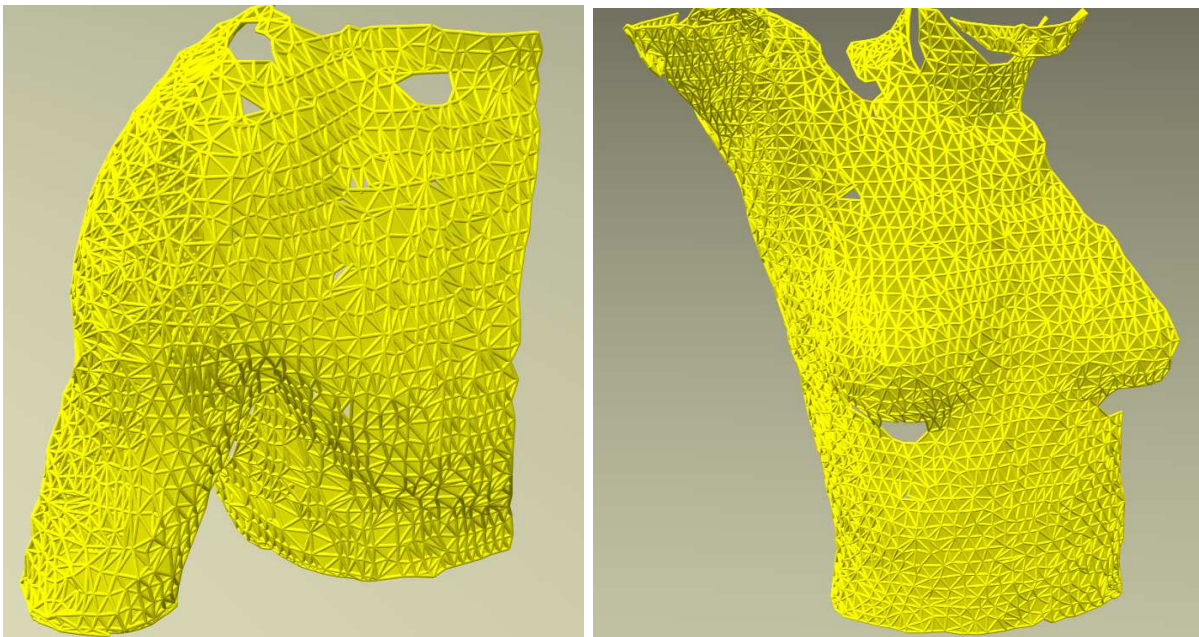


FIG. 5.6 – Construction d'un maillage à partir de deux segments d'images acquises par profilométrie.

### 5.3.2 Performances.

Nous avons appliqué l'algorithme BPA décrit ci-dessus, sur un grand nombre d'images 3D, soit acquises par nos propres capteurs, soit récupérées dans une des bases d'images disponibles sur le Web. Pour les différents exemples commentés ci-dessus, les temps de calcul sur un Pentium III 933MHz, sont rappelés avec la table 5.2. Globalement les temps d'exécution sont comparables ou meilleurs que ceux nécessaires pour faire converger une surface déformable; la méthode de "Marching cubes" reste largement la plus rapide.

## 5.4 Résultats de l'intégration.

Nous avons fait des tests de reconstruction avec un capteur stéréoscopique sur deux objets texturés, un objet géométrique et un objet de forme libre, en utilisant différentes stratégies de recalage et de sélection de points.

### 5.4.1 Stratégie.

Nous allons décrire la création d'un modèle à partir de son acquisition en utilisant les méthodes décrites dans ce travail. La liste des traitements que nous avons dû effectuer pour cette intégration, est la suivante :

- étalonnage du capteur stéréoscopique.
- planification des positions du bras pour décrire une trajectoire autour d'un objet (version simplifiée d'une approche résolvant le problème NBV).
- acquisition des images.
- obtention des données 3D par stéréocorrélation.
- extraction des points de contrôle par décimation de triangles (avec le logiciel libre QSLIM ou GTS) ou par extraction des contours.
- recalage des vues.
- fusion.

Notre méthode de modélisation est indépendante du type de capteur comme nous l'avons démontré au long des chapitres précédents. Mais dans ce chapitre de présentation des résultats expérimentaux, nous avons choisi d'utiliser une tête stéréoscopique avec des caméras connectées par liaison IEEE1394 (ou *Firewire*).

### 5.4.2 Etalonnage.

Le système stéréoscopique utilisé au LAAS est basé sur la corrélation entre les pixels des images gauche et droite. Cette solution est acceptable pour les scènes à l'extérieur où la texture est clairement différentiable : l'objectif alors est l'acquisition d'un modèle du terrain exploitable pour la navigation des robots mobiles.

Notre application est différente : il s'agit de la modélisation d'objets plutôt petits à l'intérieur. Dans ce contexte, la stratégie de stéréocorrélation a un grand taux d'échec à cause de l'absence de texture et le manque de lumière suffisante. Par conséquent, nous avons réduit la distance entre caméras au maximum pour diminuer l'écart sur les vues et ainsi approcher l'objet à modéliser des caméras afin d'avoir la résolution la meilleure dans les images ; cela permet d'acquérir des informations de texture sur la surface de l'objet. En plus, la profondeur de champ doit être grande et pour éviter le problème de lumière nous utilisons un éclairage artificiel. On peut constater sur les images en figure 5.7 et 5.10 que nous ne sommes pas parvenus à une configuration optimale pour le banc stéréo porté par le robot. Si les caméras sont rapprochées de l'objet l'image devient floue et la stéréo échoue.

La stéréovision a besoin de données de calibrage précises sur les caméras. Ce calibrage est fait soit par la méthode développée au LAAS-CNRS, soit avec une application Matlab. Les deux techniques donnent des résultats similaires ; l'environnement et la robustesse de Matlab le rendent plus agréable à utiliser.

### 5.4.3 Pompe.

#### Acquisition de données.

L'objet à modéliser est une pompe à eau (environ 10cm de haut, 5cm de large). C'est une pièce mécanique texturée avec différents types de formes : surfaces planes, cylindriques, ouvertures... Le capteur est déplacé autour de l'objet avec deux types d'éclairage à chaque vue.



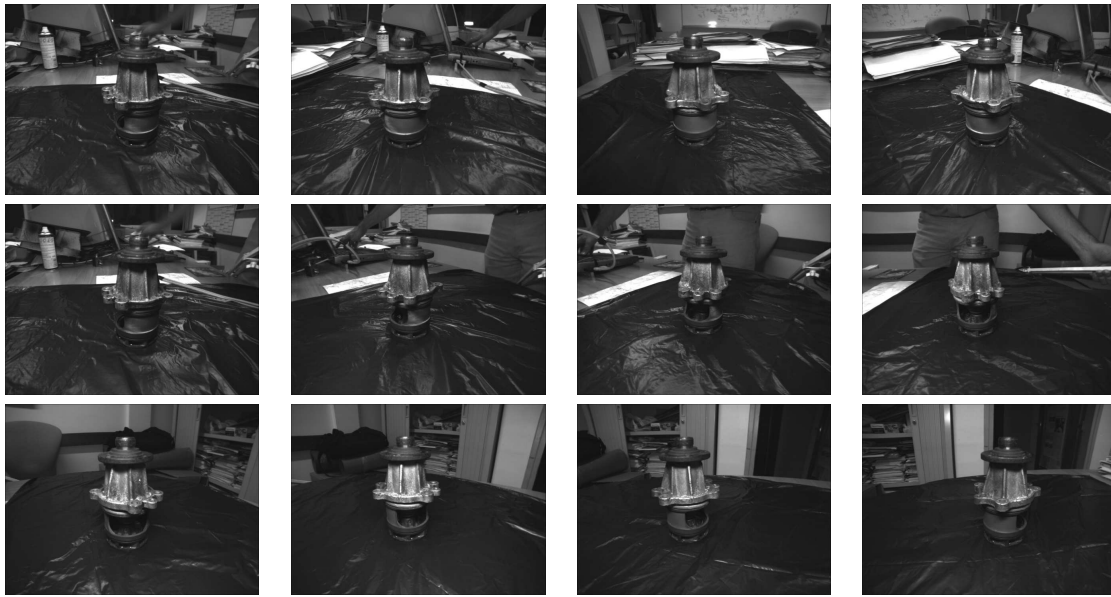


FIG. 5.7 – Images prises par la caméra stéréoscopique sur la pompe (caméra droite).

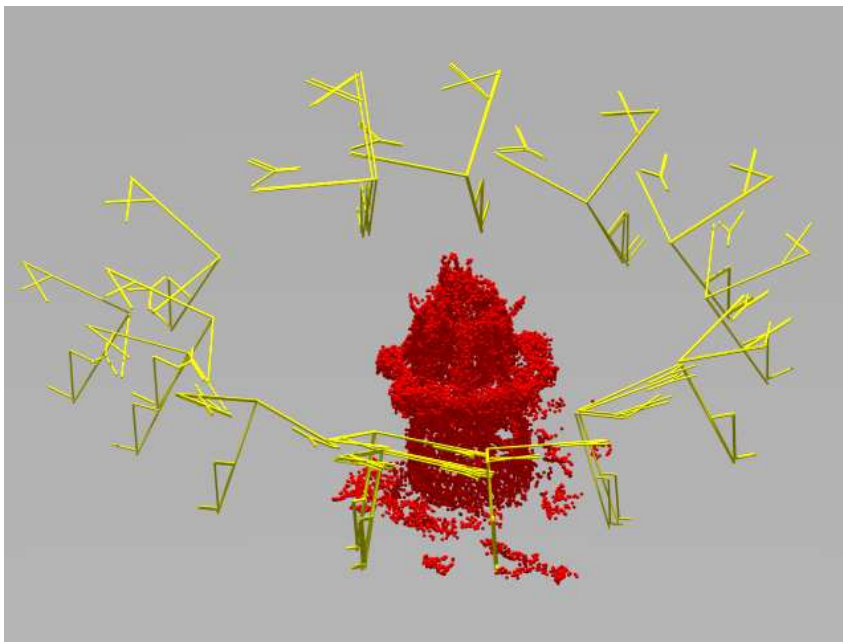


FIG. 5.8 – Différents points de vue autour de l’objet “Pompe”.

### Recalage.

Nous avons décidé d’utiliser une méthode de recalage semi-incrémental pour optimiser les ressources de calcul et réduire l’erreur correspondante : chaque vue est initialement recalée avec sa paire prise à la même position mais avec un éclairage modifié. Ensuite les résultats de chaque paire de recalage sont rangés dans un arbre par groupes de deux à la façon d’un championnat de tennis, chaque fois en prenant en compte la similitude entre les paires à recaler pour assurer la convergence. Pour la dernière phase avec

Phase 1	Phase 2	Phase 3	Phase 4
1, 2			
	1, 2, 3, 4		
3, 4			
		1..9	
5, 6			
	5, 6, 8, 9		
8, 9			
		1..17	
10, 11			1..17,26..31
	10, 11, 12, 13		
			1..17,26..31,18..25
12, 13			
		10..17	
14, 15			
	14, 15, 16, 17		
16, 17			
18, 19			
	18, 19, 20, 2		
20, 21			
		18..25	
22, 23			
	22, 23, 24, 25		
24, 25			
26, 27			
	26, 27, 28, 29		
28, 29			
		26..31	
30, 31			

TAB. 5.3 – Ordre de recalage des 31 vues de l'objet de test.

4 images à recaler, nous utilisons le processus incrémental qui nous donne l'objet complet.

Le recalage est fait en deux phases. La première est un recalage grossier, qui utilise les points représentatifs obtenus par décimation d'un maillage triangulaire construit à partir de l'image 3D donnée par la stéréocorrélation. L'objectif de cette phase est d'obtenir rapidement la position correcte entre les vues. La deuxième phase est le recalage fin par lequel la position entre vues doit être améliorée; nous avons utilisé les transformations finales données par la phase antérieure avec les nuages complets (non décimés).

### Fusion.

La génération de maillage pour trouver une représentation surfacique de l'objet peut être faite de différentes façons :

- **incrémentale** : à chaque pas de recalage, le maillage est construit avec les points additionnels. Cette méthode peut être synchrone ou asynchrone par rapport à l'algorithme de recalage, car le recalage est basé sur des points seulement, ainsi le maillage est généré à la fin d'un recalage entre deux vues ou à la fin d'un déplacement multiple en utilisant les transformations obtenues pour chaque recalage par paires.

Le maillage incrémental est plus stable et plus rapide car l'erreur de recalage sur les points déjà maillés n'est pas prise en compte et le nombre de points est moindre dans les phases initiales.

- **globale** : le maillage est généré à partir du nuage intégrant toutes les vues recalées. L'avantage de la technique est le contrôle sur la densité finale du nuage de points, et son inconvénient est la présence plus représentative de l'erreur de recalage.

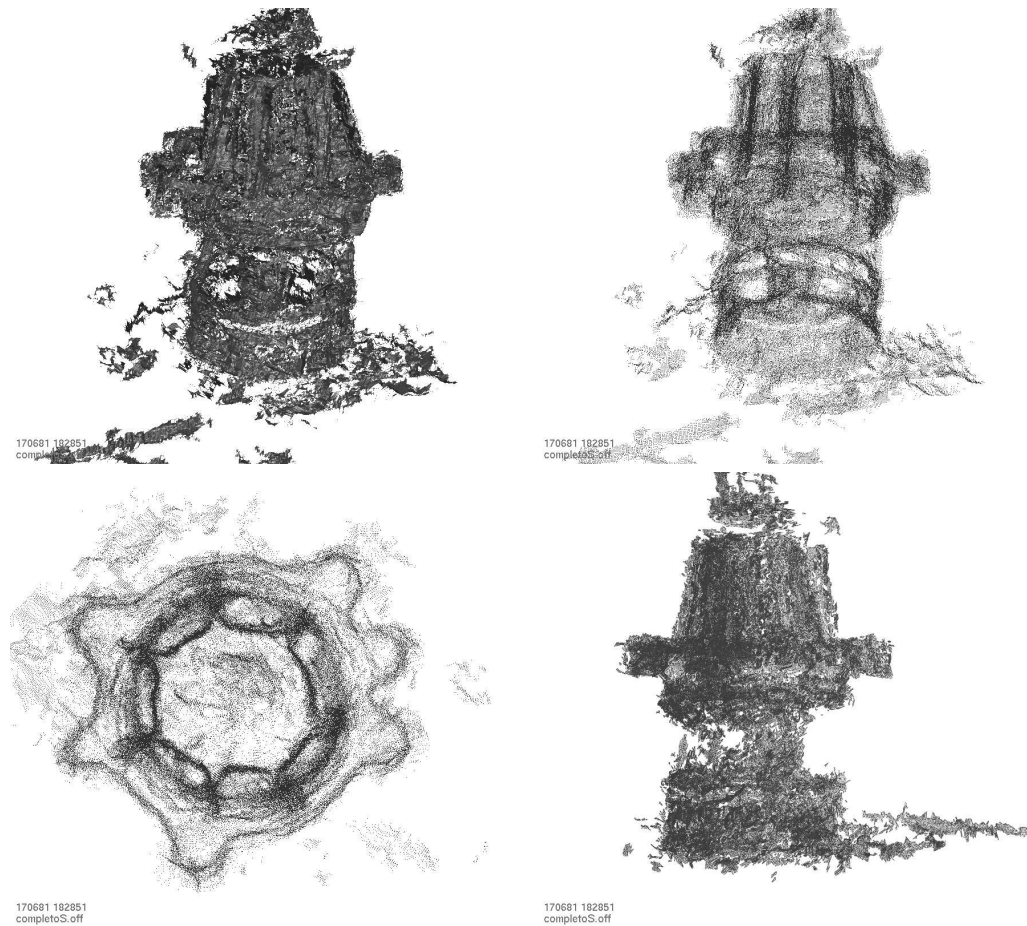


FIG. 5.9 – Les nuages de points des vues fusionnées.

#### 5.4.4 Tête-de-dragon.

Les 12 images prises par la caméra droite sont montrées sur la figure 5.10.

##### Recalage.

Dans cet exemple nous avons utilisé l'extraction des contours pour obtenir les points de contrôle.

La stratégie de recalage a été légèrement différente par rapport à l'exemple précédent ; il est fait dans 8 séquences.

La figure 5.11 montre les nuages de points acquis depuis les images sur l'objet *Tête de Dragon*.

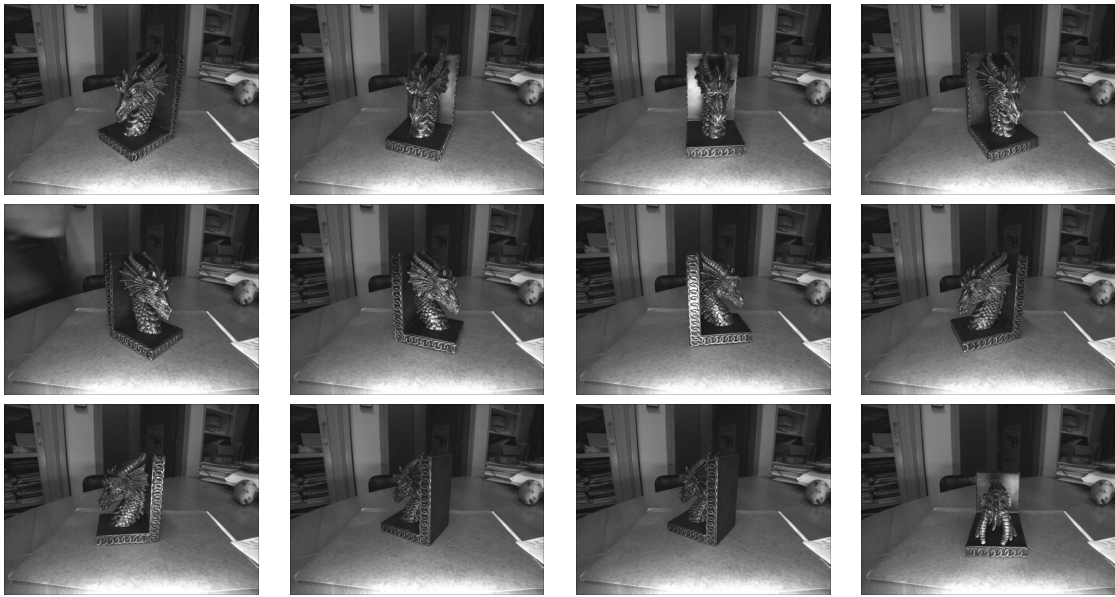


FIG. 5.10 – Images prises par la caméra stéréoscopique (caméra droite).

## 5.5 Conclusion

Les expérimentations in situ se sont avérées difficiles ; les imprécisions de la stéréovision, rendent difficile la construction des maillages par la méthode BPA. Nous poursuivons les travaux pour résoudre ce problème.

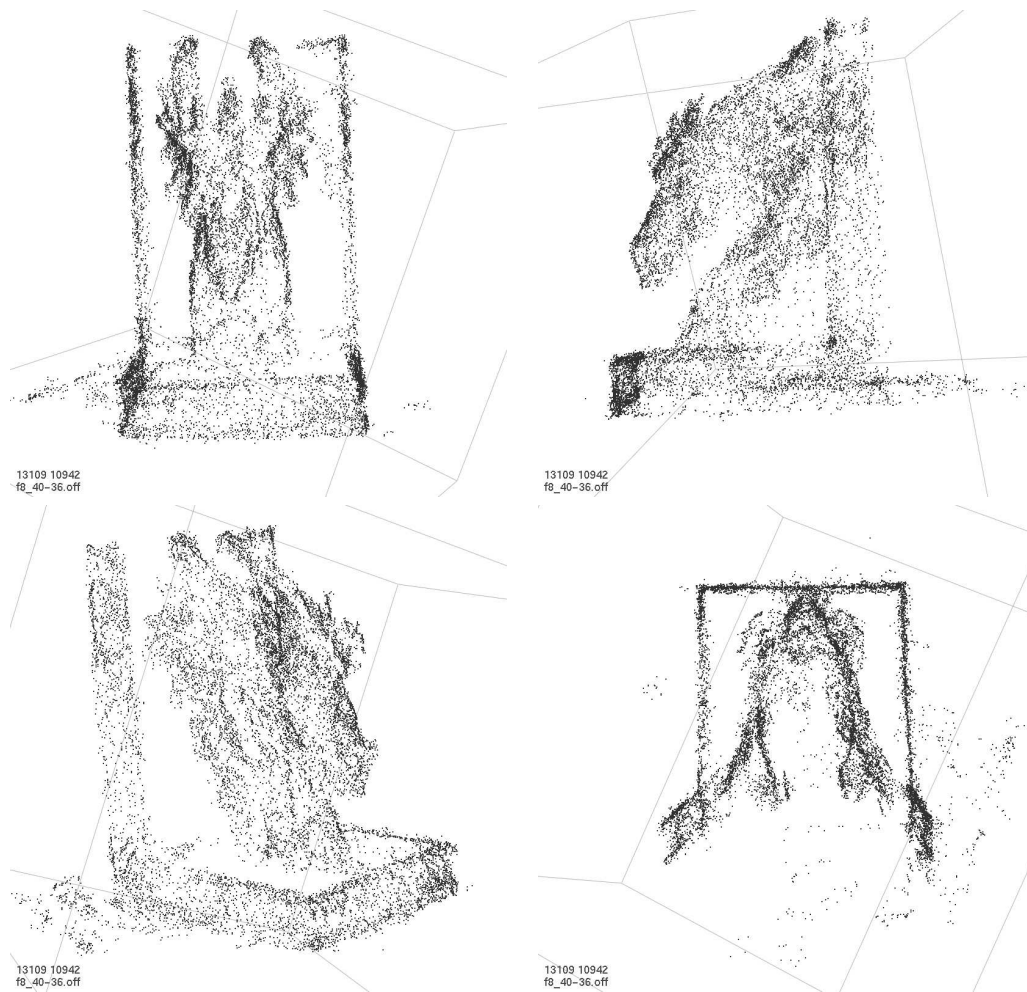


FIG. 5.11 – *Les nuages de points de contrôle des vues fusionnées.*

# Conclusion générale

Le résultat du travail que nous avons effectué durant ces quatre années de thèse, consiste en un ensemble d'outils qu'il reste à intégrer sur un robot, pour obtenir un système de modélisation 3D automatique exploitable pour différentes applications. En vu des limitations des outils existants, tel que LFM de *UK Robotics*, que nous avons utilisé dans le cadre du projet CAMERA, rappelons que notre ambition était la création d'un système de modélisation qui fonctionne sans aucune intervention d'un opérateur.

Nous avons traité le problème en étudiant successivement, trois fonctions différentes, le recalage d'images 3D, la construction d'un maillage triangulaire et la sélection du point de vue optimal ou *Next Best View*, en gardant toujours une indépendance par rapport au type d'image 3D à traiter. Pour ces trois fonctions, nous avons analysé l'état de l'art, sélectionné le type d'algorithme le plus adapté à nos besoins, développé notre propre version de cet algorithme et évalué les résultats obtenus.

Pour les trois fonctions développées, nos résultats sont les suivants :

- une méthode de recalage robuste pour une grande variété d'images tridimensionnelles, avec un mécanisme de commutation automatique entre recalage grossier et fin. La méthode se fonde sur l'algorithme ICP, qui est une technique déjà très étudiée et améliorée dans de nombreux travaux, depuis les versions initiales proposées en 1992 environ. Nous avons gardé sa conception d'origine pour gagner en vitesse et en généralité par rapport aux types d'images à recalcer.
- un processus de maillage 3D adapté à nos besoins. Il se fonde sur l'algorithme BPA, et sa principale qualité est la flexibilité d'emploi : il peut être exploité de manière globale (maillage d'un nuage de points) ou incrémentale (fusion d'une image 3D avec un maillage existant), il permet d'intégrer tous les types d'images 3D et nos nombreux tests attestent de sa robustesse.  
Rappelons que des travaux sur le thème de la modélisation 3D, avait été menés dans notre équipe, par J.Bozier [Bozier 00a] : construction incrémentale d'un modèle 3D de l'environnement, par fusion de maillages locaux dans un maillage global. La décision de développer une nouvelle méthode de construction de maillage 3D, n'a pas été facile à prendre parce que le temps de développement et de mise au point pour ce type d'algorithme est très long : mais,
  - l'approche incrémentale développée par J.Bozier était plus orientée vers la modélisation d'un environnement. Pour modéliser un objet, une construction globale à partir d'un nuage de points 3D est possible.
  - la fonction de fusion de maillages est très complexe ; elle s'est révélée peu robuste.
  - les alternatives existantes ne sont pas applicables aux conditions de notre problème ("Marching cubes", surfaces déformables, Delaunay)
- une contribution plus ponctuelle sur une méthode traitant du problème *Next Best View*, afin de fermer la boucle du processus de modélisation automatique. En ce domaine, dans le contexte *Modélisation incrémentale d'un environnement*, l'approche classique consiste à exhiber et une fonction d'utilité associée à chaque point de vue potentiel, puis à rechercher par optimisation non linéaire, le point de vue qui maximise cette fonction. Cette approche s'est avérée très difficile : maxima locaux, lenteur de convergence. . . Nous avons proposé donc une méthode sous-optimale adaptée au contexte *Modélisation d'un objet*, et à notre configuration robotique : un capteur 3D porté par un bras manipulateur mobile.

Nous avons donc développé des traitements sur des données 3D ; ce faisant, nous avons réalisé plusieurs outils ou analyses qui peuvent s'avérer très utiles :

- la méthode d'ICP a été analysée en détail, notamment les différentes méthodes qui permettent d'estimer la transformation 3D optimale qui relie deux ensembles de points 3D appariés, parmi lesquelles celles fondées sur la technique SVD, sur les quaternions et sur les doubles quaternions.
- la mise-au-point des traitements réalisés dans le processus de modélisation et la nécessité d'exploiter plusieurs types d'images 3D, avec différents formats, a rendu indispensable le développement d'un logiciel de visualisation 3D, intégré aux traitements. Ce logiciel, appelé *PointView* est fondé sur OpenGL.
- dans le contexte Robotique, nous comptons intégrer nos travaux avec l'acquisition d'images par stéréocorrélation. Nous avons exploité les outils (stéréovision dense, calibrage...) développés dans notre groupe de recherche. Néanmoins, nous avons adapté certaines méthodes pour en améliorer les performances : calibrage rapide (fondé sur OpenCV), analyse des résultats avec un éclairage artificiel, densification des images par interpolation...
- le développement du programme de maillage par BPA, se fonde sur une bibliothèque regroupant les opérateurs de base sur les triangles. Cette bibliothèque est disponible pour développer d'autres logiciels plus performants dans le futur.

Les points en suspens laissés dans notre travail sont :

- Un point faible dans la méthode de recalage par ICP est encore la sélection de l'estimée initiale et la détection d'un recalage faux. La solution de ces problèmes est importante pour obtenir une méthode de recalage complètement automatique. Néanmoins, nous sommes arrivés à une technique robuste de correction des estimées initiales, par triangulation des vues (cycle de 3 vues).
- La construction d'un maillage par l'algorithme BPA de la boule pivotante, manque de souplesse lorsque les données sensorielles sont très bruitées, ce qui est le cas des données acquises par stéréovision. Notre comparaison avec les méthodes de type *Surfaces Déformables*, a montré que BPA produit une surface plus irrégulière, cela pour suivre les points de mesure. Ce défaut doit être résolu avec une stratégie d'insertion de points ; on peut envisager l'inclusion de caractéristiques élastiques dans l'algorithme BPA pour allier les avantages des deux méthodes.
- La méthode de *Next Best View* donne une solution sous-optimale, d'autant que la fonction de qualité des points de vue est très coûteuse à minimiser. Il conviendrait d'améliorer cette fonction et de lui donner une forme qui favorise la recherche de l'optimum. Par ailleurs, l'intégration du planificateur *Next Best View* avec un planificateur de trajectoire est nécessaire pour déterminer si un point de vue potentiel pour la tâche de modélisation, est accessible par le mécanisme robotique qui porte le capteur.
- enfin, le temps a manqué pour intégrer ces trois fonctions, à bord d'un robot, pour obtenir vraiment un système de modélisation automatique. Notons aussi que du fait de la complexité des algorithmes ICP et BPA, les temps des traitements restent importants. Pour l'instant, seules les acquisitions des images sur l'objet à modéliser, sont réalisées en ligne sur le bras mobile *Hilare2-bis* ; la construction du modèle est réalisée hors-ligne.

Nous concluons sur ce que nous considérons comme le problème essentiel : l'acquisition des données sensorielles. Les résultats donnés par la stéréocorrélation ne se sont pas révélés aussi acceptables que nous le souhaitions, et nous avons perdu énormément d'énergie à chercher la meilleure configuration, le meilleur calibrage, le meilleur paramétrage... de notre méthode de stéréovision.

Bien que nous exploitions une méthode de stéréovision dense, du fait des problèmes d'appariement, nous perdons de nombreuses informations contenues dans les images d'origine, en comparaison avec d'autres méthodes de modélisation basées sur des images 2D par "convex hull" et surfaces déformables comme par exemple dans les travaux [Lazebnik 01] ou [Hernández 04].

Il est clair que la méthode de modélisation présentée dans ce mémoire, est faite pour travailler avec des données intrinsèquement tridimensionnelles, comme celles d'un capteur laser. Pour mieux exploiter la stéréovision et éviter des pertes d'information, il conviendrait de réaliser des traitements supplémentaires

---

pour les données image 2D non appariées.



FIG. 12 – *Capteur 3D optique à temps de vol.*

Signalons l'arrivée prometteuse sur le marché de nouveaux capteurs 3D, de résolution "suffisante", peu encombrants, légers et peu chers dans le futur, donc capteurs que nous pourrions monter sur notre manipulateur. Il s'agit de capteurs optiques à temps de vol; un des capteurs disponibles est présenté en figure 12; il conviendrait d'évaluer ces capteurs pour l'acquisition des données 3D exploitables pour la modélisation 3D d'objets depuis un bras mobile.

Au delà de l'acquisition, pour réaliser cette tâche en temps réel sur un système embarqué, il faudra résoudre le problème de la performance, pour les trois fonctions étudiées dans notre travail, les algorithmes ICP, BPA et NBV. Dans les trois cas, il existe de grandes possibilités de parallélisation et ces versions parallèles devraient être analysées.





# Annexe A

## Représentations des rotations par les quaternions

### A.1 Quaternions.

Les quaternions, introduits par Sir W. R. Hamilton le 16 octobre 1843, permettent de représenter les rotations 3D avec les avantages suivants par rapport à la matrice de rotation traditionnelle obtenue à partir des angles d'Euler :

- Il n'y a pas de "Gimbal Lock", problème qui apparaît avec l'usage des angles d'Euler, puisque la rotation finale dépend de l'ordre de multiplication des matrices de rotation. Par conséquent, avec les angles d'Euler, il est possible qu'un axe de rotation coïncide avec un autre, ce qui diminue les degrés de liberté.
- Le quaternion permet de faire directement une rotation autour d'un axe arbitraire.
- L'interpolation des quaternions donnent des rotations prévisibles et progressives, grâce à une représentation continue.
- Le nombre de termes à traiter est égal 4 et n'atteint pas 9 comme dans le cas de la matrice de rotation. Par conséquent les calculs sont moins lourds.
- Le problème des déformations ou "sheer", à cause d'une accumulation des erreurs numériques dans les multiplications successives des matrices de transformation, est évité, car le quaternion maintient sa norme.

Les quaternions sont une extension des nombres complexes, pour lesquels on a les règles de base suivantes :

règle :	$\mathbf{i} * \mathbf{i} = -1$
définition :	$\mathbf{z} = a + b\mathbf{i}$
conjugué :	$\mathbf{z}' = a - b\mathbf{i}$
amplitude	$\ \mathbf{z}\  = \sqrt{\mathbf{z} * \mathbf{z}'} = \sqrt{a^2 + b^2}$
multiplication :	$\mathbf{z}_1 * \mathbf{z}_2 = (a_1 a_2 - b_1 b_2) + (a_1 b_2 + b_1 a_2)\mathbf{i}$

L'extension des nombres complexes pour les quaternions consiste à augmenter jusqu'à trois la dimension complexe  $\mathbf{i}, \mathbf{j}, \mathbf{k}$ , tel que ses combinaisons soient définies par :

*	$\mathbf{i}$	$\mathbf{j}$	$\mathbf{k}$
$\mathbf{i}$	-1	$\mathbf{k}$	$-\mathbf{j}$
$\mathbf{j}$	$-\mathbf{k}$	-1	$\mathbf{i}$
$\mathbf{k}$	$\mathbf{j}$	$-\mathbf{i}$	-1

TAB. A.1 – Combinaisons entre dimensions complexes du quaternion

Les propriétés qui en découlent sont :

$$\begin{aligned}
 \text{définition :} & \quad \mathbf{q} = w + x\mathbf{i} + y\mathbf{j} + z\mathbf{k} \\
 \text{conjugué :} & \quad \mathbf{q}' = w - x\mathbf{i} - y\mathbf{j} - z\mathbf{k} \\
 \text{amplitude :} & \quad \|\mathbf{q}\| = \sqrt{\mathbf{q} * \mathbf{q}'} = \sqrt{w^2 + x^2 + y^2 + z^2} \\
 \text{quaternion unité :} & \quad \|\mathbf{q}\| = 1 \Rightarrow \mathbf{q}^{-1} = \mathbf{q}' \\
 \text{associativité :} & \quad (\mathbf{q}_1 * \mathbf{q}_2) * \mathbf{q}_3 = \mathbf{q}_1 * (\mathbf{q}_2 * \mathbf{q}_3) \\
 \text{non-commutativité :} & \quad \mathbf{q}_1 * \mathbf{q}_2 \neq \mathbf{q}_2 * \mathbf{q}_1
 \end{aligned}$$

Les représentations possibles du quaternion sont :

$$\begin{aligned}
 \text{Combinaison linéaire :} & \quad \mathbf{q} = w + x\mathbf{i} + y\mathbf{j} + z\mathbf{k}. \\
 \text{Vecteur 4D :} & \quad \mathbf{q} = [x \ y \ z \ w]. \\
 \text{un scalaire et un vecteur 3D :} & \quad \mathbf{q} = (s, \check{\mathbf{q}}), \text{ où } s = w \text{ et } \check{\mathbf{q}} = [x \ y \ z].
 \end{aligned}$$

Si le quaternion a une amplitude unitaire, il représente une rotation d'un angle  $\theta$  autour d'un axe de rotation ayant un vecteur unitaire  $\vec{u}$  :

$$\begin{aligned}
 \mathbf{q} &= (s, \check{\mathbf{v}}) \\
 s &= \cos \frac{\theta}{2} \\
 \check{\mathbf{v}} &= \vec{u} \sin \frac{\theta}{2}
 \end{aligned} \tag{A.1}$$

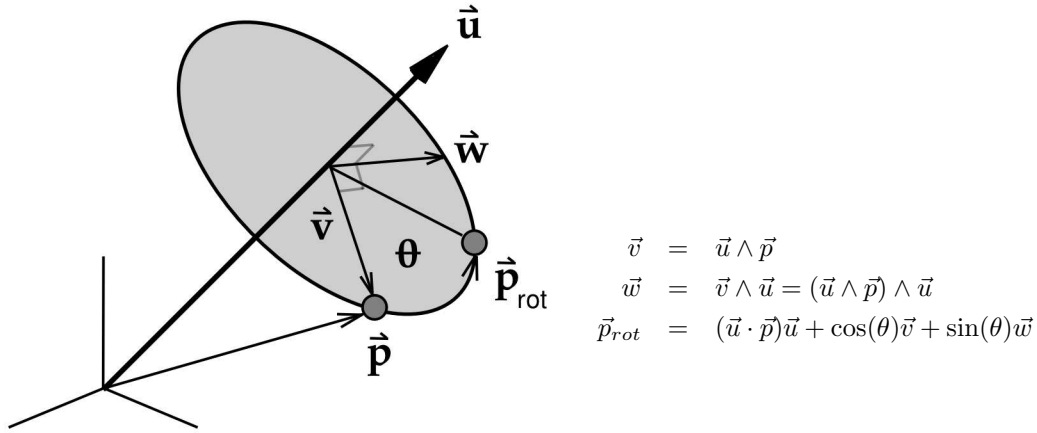


FIG. A.1 – Le quaternion représente une rotation d'un angle  $\theta$  autour d'un vecteur unitaire  $\vec{u}$ .

L'application d'une rotation sur un point 3D  $\vec{p}$ , se représente de la manière suivante, si  $\vec{p}$  est converti au préalable en un quaternion avec un scalaire zéro :

$$\begin{aligned}
 \mathbf{P} &= (0, \vec{p}) \\
 \mathbf{P}_{rot} &= \mathbf{q}\mathbf{P}\mathbf{q}^{-1}
 \end{aligned} \tag{A.2}$$

Avant d'approfondir la représentation d'une rotation par des quaternions, on doit analyser la multiplication de deux quaternions  $q_1$  et  $q_2$ , formulée en fonction de leur représentations par un scalaire  $s$  et un vecteur 3D  $\check{\mathbf{q}}$ , ce qui donne :

$$\mathbf{q}_1 * \mathbf{q}_2 = \left[ w_1\check{\mathbf{q}}_2 + w_2\check{\mathbf{q}}_1 - \check{\mathbf{q}}_2 \wedge \check{\mathbf{q}}_1 \quad -\check{\mathbf{q}}_2 \cdot \check{\mathbf{q}}_1 + w_1w_2 \right]$$

ou suivant une forme matricielle :

$$\mathbf{q}_1 * \mathbf{q}_2 = \mathbf{q}_1 \mathbf{R}_{row}(\mathbf{q}_2) = \mathbf{q}_2 \mathbf{L}_{row}(\mathbf{q}_1) \quad (\text{A.3})$$

avec :

$$\begin{aligned} \mathbf{R}_{row}(\mathbf{q}) &= \begin{bmatrix} w\mathbf{I} - \mathbf{K}(\check{\mathbf{q}}) & -\check{\mathbf{q}}^T \\ \check{\mathbf{q}} & w \end{bmatrix} \\ \mathbf{L}_{row}(\mathbf{q}) &= \begin{bmatrix} w\mathbf{I} + \mathbf{K}(\check{\mathbf{q}}) & -\check{\mathbf{q}}^T \\ \check{\mathbf{q}} & w \end{bmatrix} \end{aligned}$$

La déduction d'A.3 est la suivante :

$$\begin{aligned} \mathbf{q}_1 &= w_1 + x_1 \mathbf{i} + y_1 \mathbf{j} + z_1 \mathbf{k} = (w_1, \check{\mathbf{q}}_1) \\ \mathbf{q}_2 &= w_2 + x_2 \mathbf{i} + y_2 \mathbf{j} + z_2 \mathbf{k} = (w_2, \check{\mathbf{q}}_2) \\ \mathbf{q}_1 * \mathbf{q}_2 &= (w_1 + x_1 \mathbf{i} + y_1 \mathbf{j} + z_1 \mathbf{k}) * (w_2 + x_2 \mathbf{i} + y_2 \mathbf{j} + z_2 \mathbf{k}) \\ &= w_1 w_2 + w_1 x_2 \mathbf{i} + w_1 y_2 \mathbf{j} + w_1 z_2 \mathbf{k} \\ &\quad + x_1 w_2 \mathbf{i} + x_1 x_2 \mathbf{i} \mathbf{i} + x_1 y_2 \mathbf{i} \mathbf{j} + x_1 z_2 \mathbf{i} \mathbf{k} \\ &\quad + y_1 w_2 \mathbf{j} + y_1 x_2 \mathbf{j} \mathbf{i} + y_1 y_2 \mathbf{j} \mathbf{j} + y_1 z_2 \mathbf{j} \mathbf{k} \\ &\quad + z_1 w_2 \mathbf{k} + z_1 x_2 \mathbf{k} \mathbf{i} + z_1 y_2 \mathbf{k} \mathbf{j} + z_1 z_2 \mathbf{k} \mathbf{k} \\ &= w_1 w_2 + w_1 x_2 \mathbf{i} + w_1 y_2 \mathbf{j} + w_1 z_2 \mathbf{k} \\ &\quad + x_1 w_2 \mathbf{i} - x_1 x_2 + x_1 y_2 \mathbf{k} - x_1 z_2 \mathbf{j} \\ &\quad + y_1 w_2 \mathbf{j} - y_1 x_2 \mathbf{k} - y_1 y_2 + y_1 z_2 \mathbf{i} \\ &\quad + z_1 w_2 \mathbf{k} + z_1 x_2 \mathbf{j} - z_1 y_2 \mathbf{i} - z_1 z_2 \\ &= w_1 w_2 - x_1 x_2 - y_1 y_2 - z_1 z_2 \\ &\quad + (w_1 x_2 + x_1 w_2 + y_1 z_2 - z_1 y_2) \mathbf{i} \\ &\quad + (w_1 y_2 - x_1 z_2 + y_1 w_2 + z_1 x_2) \mathbf{j} \\ &\quad + (w_1 z_2 + x_1 y_2 - y_1 x_2 + z_1 w_2) \mathbf{k} \end{aligned}$$

Nous aboutissons à la représentation de vecteur 4D, pour séparer le résultat en un vecteur et une matrice. Il existe deux descriptions :

$$\begin{aligned} \mathbf{q}_1 * \mathbf{q}_2 &= \begin{bmatrix} w_1 x_2 + x_1 w_2 + y_1 z_2 - z_1 y_2 \\ w_1 y_2 - x_1 z_2 + y_1 w_2 + z_1 x_2 \\ w_1 z_2 + x_1 y_2 - y_1 x_2 + z_1 w_2 \\ w_1 w_2 - x_1 x_2 - y_1 y_2 - z_1 z_2 \end{bmatrix}^T \\ &= [x_1 \ y_1 \ z_1 \ w_1] \begin{bmatrix} w_2 & -z_2 & y_2 & -x_2 \\ z_2 & w_2 & -x_2 & -y_2 \\ -y_2 & x_2 & w_2 & -z_2 \\ x_2 & y_2 & z_2 & w_2 \end{bmatrix} \\ &= \mathbf{q}_1 \mathbf{R}_{row}(\mathbf{q}_2) \\ &= [x_2 \ y_2 \ z_2 \ w_2] \begin{bmatrix} w_1 & z_1 & -y_1 & -x_1 \\ -z_1 & w_1 & x_1 & -y_1 \\ y_1 & -x_1 & w_1 & -z_1 \\ x_1 & y_1 & z_1 & w_1 \end{bmatrix} \\ &= \mathbf{q}_2 \mathbf{L}_{row}(\mathbf{q}_1) \quad (\text{A.4}) \end{aligned}$$

Les matrices  $\mathbf{L}_{row}(\mathbf{q})$  et  $\mathbf{R}_{row}(\mathbf{q})$  de A.4 s'écrivent à partir d'une division sous la forme suivante :

$$\begin{aligned}
 \mathbf{R}_{row}(\mathbf{q}) &= \left[ \begin{array}{c} \begin{bmatrix} w & -z & y \\ z & w & -x \\ -y & x & w \end{bmatrix} \\ \begin{bmatrix} -x \\ -y \\ -z \\ w \end{bmatrix} \end{array} \right] = \left[ \begin{array}{c} w\mathbf{I} - \begin{bmatrix} 0 & z & -y \\ -z & 0 & x \\ y & -x & 0 \end{bmatrix} \\ \check{\mathbf{q}} \\ w \end{array} \right] \\
 &= \left[ \begin{array}{c} w\mathbf{I} - \mathbf{K}(\check{\mathbf{q}}) \\ \check{\mathbf{q}} \\ w \end{array} \right] \\
 \mathbf{L}_{row}(\mathbf{q}) &= \left[ \begin{array}{c} \begin{bmatrix} w & z & -y \\ -z & w & x \\ y & -x & w \end{bmatrix} \\ \begin{bmatrix} -x \\ -y \\ -z \\ w \end{bmatrix} \end{array} \right] = \left[ \begin{array}{c} w_1\mathbf{I} + \begin{bmatrix} 0 & z & -y \\ -z & 0 & x \\ y & -x & 0 \end{bmatrix} \\ \check{\mathbf{q}} \\ w \end{array} \right] \\
 &= \left[ \begin{array}{c} w\mathbf{I} + \mathbf{K}(\check{\mathbf{q}}) \\ \check{\mathbf{q}} \\ w \end{array} \right] \tag{A.5}
 \end{aligned}$$

Caractéristiques des matrices  $\mathbf{R}_{row}(\mathbf{q})$  et  $\mathbf{L}_{row}(\mathbf{q})$  :

$$\begin{aligned}
 \mathbf{R}_{row}(\mathbf{q}^T) &= \left[ \begin{array}{c} w\mathbf{I} - \mathbf{K}(-\check{\mathbf{q}}) \\ -\check{\mathbf{q}} \\ w \end{array} \right] = \left[ \begin{array}{c} w\mathbf{I} - \mathbf{K}(\check{\mathbf{q}})^T \\ -\check{\mathbf{q}} \\ w \end{array} \right] = \mathbf{R}_{row}(\mathbf{q})^T \\
 \mathbf{L}_{row}(\mathbf{q}^T) &= \left[ \begin{array}{c} w\mathbf{I} + \mathbf{K}(-\check{\mathbf{q}}) \\ -\check{\mathbf{q}} \\ w \end{array} \right] = \left[ \begin{array}{c} w\mathbf{I} + \mathbf{K}(\check{\mathbf{q}})^T \\ -\check{\mathbf{q}} \\ w \end{array} \right] = \mathbf{L}_{row}(\mathbf{q})^T
 \end{aligned}$$

Le matrice oblique symétrique ("skew-symmetric")  $\mathbf{K}(\check{\mathbf{q}})$  est la forme matricielle du produit vectoriel :

$$\begin{aligned}
 \check{\mathbf{p}} &= [x_1 \ y_1 \ z_1] \\
 \check{\mathbf{q}} &= [x_2 \ y_2 \ z_2] \\
 \check{\mathbf{p}}\mathbf{K}(\check{\mathbf{q}}) &= [x_1 \ y_1 \ z_1] \begin{bmatrix} 0 & z_2 & -y_2 \\ -z_2 & 0 & x_2 \\ y_2 & -x_2 & 0 \end{bmatrix} = [(-z_2y_1 + z_1y_2) \ (-z_1x_2 + x_1z_2) \ (-x_1y_2 + y_1x_2)] \\
 &= \check{\mathbf{q}} \wedge \check{\mathbf{p}} \tag{A.6}
 \end{aligned}$$

et elle obéit à la règle de ce type de matrices, qui est :

$$\mathbf{K}(-\check{\mathbf{q}}) = -\mathbf{K}(\check{\mathbf{q}}) = \mathbf{K}(\check{\mathbf{q}})^T \tag{A.7}$$

et

$$\check{\mathbf{p}}\mathbf{K}(\check{\mathbf{q}})\check{\mathbf{p}}^T = 0 \tag{A.8}$$

Les expressions A.5 sont portées dans A.4 et développées pour obtenir une expression supplémentaire du produit :

$$\begin{aligned}
 \mathbf{q}_1 * \mathbf{q}_2 &= \mathbf{q}_1 \mathbf{R}_{row}(\mathbf{q}_2) = [\check{\mathbf{q}}_1, w_1] \left[ \begin{array}{c} w_2\mathbf{I} - \mathbf{K}(\check{\mathbf{q}}_2) \\ \check{\mathbf{q}}_2 \\ w_2 \end{array} \right] \\
 &= [ \check{\mathbf{q}}_1(w_2\mathbf{I} - \mathbf{K}(\check{\mathbf{q}}_2)) + w_1\check{\mathbf{q}}_2 \quad , \quad -\check{\mathbf{q}}_1\check{\mathbf{q}}_2^T + w_1w_2 ] \\
 &= [ w_1\check{\mathbf{q}}_2 + w_2\check{\mathbf{q}}_1 - \check{\mathbf{q}}_1\mathbf{K}(\check{\mathbf{q}}_2) \quad , \quad -\check{\mathbf{q}}_1\check{\mathbf{q}}_2^T + w_1w_2 ] \\
 \mathbf{q}_1 * \mathbf{q}_2 &= \mathbf{q}_2 \mathbf{L}_{row}(\mathbf{q}_1) = [\check{\mathbf{q}}_2, w_2] \left[ \begin{array}{c} w_1\mathbf{I} + \mathbf{K}(\check{\mathbf{q}}_1) \\ \check{\mathbf{q}}_1 \\ w_1 \end{array} \right] \\
 &= [ \check{\mathbf{q}}_2(w_1\mathbf{I} + \mathbf{K}(\check{\mathbf{q}}_1)) + w_2\check{\mathbf{q}}_1 \quad , \quad -\check{\mathbf{q}}_2\check{\mathbf{q}}_1^T + w_1w_2 ] \\
 &= [ w_1\check{\mathbf{q}}_2 + w_2\check{\mathbf{q}}_1 + \check{\mathbf{q}}_2\mathbf{K}(\check{\mathbf{q}}_1) \quad , \quad -\check{\mathbf{q}}_2\check{\mathbf{q}}_1^T + w_1w_2 ] \\
 \Rightarrow \mathbf{q}_1 * \mathbf{q}_2 &= [ w_1\check{\mathbf{q}}_2 + w_2\check{\mathbf{q}}_1 - \check{\mathbf{q}}_2 \wedge \check{\mathbf{q}}_1 \quad , \quad -\check{\mathbf{q}}_2 \cdot \check{\mathbf{q}}_1 + w_1w_2 ] \tag{A.9}
 \end{aligned}$$

L'idée est d'obtenir la matrice de rotation  $\mathbf{R}$  en fonction du quaternion  $\mathbf{q}$ . Pour arriver au but, on applique la rotation  $\mathbf{R}$  au point 3D  $\vec{x}$ ; comme dans A.2 :

$$\vec{X}_{rot} = \mathbf{R}\vec{X} \Rightarrow \mathbf{X}_{rot} = [\vec{X}_{rot}, 0] = \mathbf{q}[\vec{X}, 0]\mathbf{q}^T = \mathbf{q}\mathbf{X}\mathbf{q}^T \quad (\text{A.10})$$

qui donne une multiplication triple de quaternions. Nous utilisons les matrices déduites au A.4 en appliquant  $\mathbf{R}_{row}$  à la paire droite et ensuite  $\mathbf{L}_{row}$  au terme résultant, à partir de la gauche. Ou bien nous appliquons  $\mathbf{L}_{row}$  vers la gauche et ensuite  $\mathbf{R}_{row}$  vers la droite, ce qui donne le même résultat grâce à l'indépendance des quaternions à l'ordre de multiplication.

$$\begin{aligned} \mathbf{q}\mathbf{X}\mathbf{q}^T &= \mathbf{q}(\mathbf{X}\mathbf{q}^T) & \mathbf{q}\mathbf{X}\mathbf{q}^T &= (\mathbf{q}\mathbf{X})\mathbf{q}^T \\ &= \mathbf{q}(\mathbf{X}\mathbf{R}_{row}(\mathbf{q}^T)) & &= (\mathbf{X}\mathbf{L}_{row}(\mathbf{q}))\mathbf{q}^T \\ &= (\mathbf{X}\mathbf{R}_{row}(\mathbf{q}^T))\mathbf{L}_{row}(\mathbf{q}) & &= (\mathbf{X}\mathbf{L}_{row}(\mathbf{q}))\mathbf{R}_{row}(\mathbf{q}^T) \\ &= \mathbf{X}\mathbf{R}_{row}(\mathbf{q}^T)\mathbf{L}_{row}(\mathbf{q}) & &= \mathbf{X}\mathbf{L}_{row}(\mathbf{q})\mathbf{R}_{row}(\mathbf{q}^T) \\ &= \mathbf{X}\mathbf{R}_{row}(\mathbf{q}^T)\mathbf{L}_{row}(\mathbf{q}) & &= \mathbf{X}\mathbf{L}_{row}(\mathbf{q})\mathbf{R}_{row}(\mathbf{q}^T) \\ &= \mathbf{X}\mathbf{Q}_{row}(\mathbf{q}) & (\text{A.11}) &= \mathbf{X}\mathbf{Q}_{row}(\mathbf{q}) \end{aligned} \quad (\text{A.12})$$

La matrice de rotation apparaît dans le déterminant de dimension 3x3 dans  $\mathbf{Q}_{row}$  :

$$\begin{aligned} \mathbf{Q}_{row}(\mathbf{q}) &= \mathbf{R}_{row}(\mathbf{q}^T)\mathbf{L}_{row}(\mathbf{q}) = \mathbf{L}_{row}(\mathbf{q})\mathbf{R}_{row}(\mathbf{q}^T) \\ &= \begin{bmatrix} w^2\mathbf{I} + 2w\mathbf{K}(\check{\mathbf{q}}) + \mathbf{K}(\check{\mathbf{q}})\mathbf{K}(\check{\mathbf{q}}) + \check{\mathbf{q}}^T\check{\mathbf{q}} & 0 \\ 0 & 1 \end{bmatrix} \end{aligned} \quad (\text{A.13})$$

qui vient du développement de :

$$\begin{aligned} \mathbf{Q}_{row}(\mathbf{q}) &= \begin{bmatrix} w\mathbf{I} + \mathbf{K}(\check{\mathbf{q}}) & \check{\mathbf{q}}^T \\ -\check{\mathbf{q}} & w \end{bmatrix} \begin{bmatrix} w\mathbf{I} + \mathbf{K}(\check{\mathbf{q}}) & -\check{\mathbf{q}}^T \\ \check{\mathbf{q}} & w \end{bmatrix} \\ &= \begin{bmatrix} (w\mathbf{I} + \mathbf{K}(\check{\mathbf{q}}))(w\mathbf{I} + \mathbf{K}(\check{\mathbf{q}})) + \check{\mathbf{q}}^T\check{\mathbf{q}} & -(w\mathbf{I} + \mathbf{K}(\check{\mathbf{q}}))\check{\mathbf{q}}^T + \check{\mathbf{q}}^T w \\ -\check{\mathbf{q}}(w\mathbf{I} + \mathbf{K}(\check{\mathbf{q}})) + \check{\mathbf{q}} w & \check{\mathbf{q}}\check{\mathbf{q}}^T + w^2 \end{bmatrix} \\ &= \begin{bmatrix} w^2\mathbf{I} + 2w\mathbf{K}(\check{\mathbf{q}}) + \mathbf{K}(\check{\mathbf{q}})\mathbf{K}(\check{\mathbf{q}}) + \check{\mathbf{q}}^T\check{\mathbf{q}} & -w\check{\mathbf{q}}^T - \mathbf{K}(\check{\mathbf{q}})\check{\mathbf{q}}^T + \check{\mathbf{q}}^T w \\ -w\check{\mathbf{q}} - \check{\mathbf{q}}\mathbf{K}(\check{\mathbf{q}}) + \check{\mathbf{q}} w & 1 \end{bmatrix} \\ &= \begin{bmatrix} w^2\mathbf{I} + 2w\mathbf{K}(\check{\mathbf{q}}) + \mathbf{K}(\check{\mathbf{q}})\mathbf{K}(\check{\mathbf{q}}) + \check{\mathbf{q}}^T\check{\mathbf{q}} & 0 \\ 0 & 1 \end{bmatrix} \end{aligned}$$

où :

$$\check{\mathbf{q}}_1^T \check{\mathbf{q}}_2 = \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} \begin{bmatrix} x_2 & y_2 & z_2 \end{bmatrix} = \begin{bmatrix} x_1x_2 & x_1y_2 & x_1z_2 \\ y_1x_2 & y_1y_2 & y_1z_2 \\ z_1x_2 & z_1y_2 & z_1z_2 \end{bmatrix} \Rightarrow \check{\mathbf{q}}^T\check{\mathbf{q}} = \begin{bmatrix} x^2 & xy & xz \\ xy & y^2 & yz \\ xz & yz & z^2 \end{bmatrix}$$

et :

$$\begin{aligned} \mathbf{K}(\check{\mathbf{q}}_1)\mathbf{K}(\check{\mathbf{q}}_2) &= \begin{bmatrix} 0 & z_1 & -y_1 \\ -z_1 & 0 & x_1 \\ y_1 & -x_1 & 0 \end{bmatrix} \begin{bmatrix} 0 & z_2 & -y_2 \\ -z_2 & 0 & x_2 \\ y_2 & -x_2 & 0 \end{bmatrix} \\ &= \begin{bmatrix} -z_1z_2 - y_1y_2 & y_1x_2 & z_1x_2 \\ x_1y_2 & -z_1z_2 - x_1x_2 & z_1y_2 \\ x_1z_2 & y_1z_2 & -y_1y_2 - x_1x_2 \end{bmatrix} \\ &= \check{\mathbf{q}}_2^T\check{\mathbf{q}}_1 - (x_1x_2 + y_1y_2 + z_1z_2)\mathbf{I} = \check{\mathbf{q}}_2^T\check{\mathbf{q}}_1 - \check{\mathbf{q}}_2\check{\mathbf{q}}_1^T\mathbf{I} \end{aligned} \quad (\text{A.14})$$

$$\Rightarrow \mathbf{K}(\check{\mathbf{q}})\mathbf{K}(\check{\mathbf{q}}) = \check{\mathbf{q}}^T\check{\mathbf{q}} - \check{\mathbf{q}}\check{\mathbf{q}}^T\mathbf{I} \quad (\text{A.15})$$

La combinaison de A.15 et A.13 aboutit :

$$\begin{aligned} \mathbf{Q}_{row}(\mathbf{q}) &= \begin{bmatrix} w^2\mathbf{I} + 2w\mathbf{K}(\check{\mathbf{q}}) + \check{\mathbf{q}}^T\check{\mathbf{q}} - \check{\mathbf{q}}\check{\mathbf{q}}^T\mathbf{I} + \check{\mathbf{q}}^T\check{\mathbf{q}} & 0 \\ 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} w^2\mathbf{I} + 2w\mathbf{K}(\check{\mathbf{q}}) + 2\check{\mathbf{q}}^T\check{\mathbf{q}} - \check{\mathbf{q}}\check{\mathbf{q}}^T\mathbf{I} & 0 \\ 0 & 1 \end{bmatrix} \end{aligned} \quad (\text{A.16})$$

Finalement la matrice de rotation est la partie supérieur gauche de l'égalité A.16 :

$$\mathbf{R} = (w^2 - \check{\mathbf{q}}\check{\mathbf{q}}^T)\mathbf{I} + 2w\mathbf{K}(\check{\mathbf{q}}) + 2\check{\mathbf{q}}^T\check{\mathbf{q}} \quad (\text{A.17})$$

Le résultat A.17 est développé suivant les éléments de matrice et il donne la formule de conversion entre le quaternion et la matrice de rotation :

$$\mathbf{R} = \begin{bmatrix} w^2 + x^2 - y^2 - z^2 & 2wz + 2xy & -2wy + 2xz \\ -2wz + 2xy & w^2 - x^2 + y^2 - z^2 & 2wx + 2yz \\ 2wy + 2xz & -2wx + 2yz & w^2 - x^2 - y^2 + z^2 \end{bmatrix} \quad (\text{A.18})$$

Nous arrivons à l'égalité A.17 si le résultat A.10 est traité à partir de ses éléments scalaires et vectoriels :

$$\begin{aligned} [\check{\mathbf{q}}, w] [\bar{x}, 0] [-\check{\mathbf{q}}, w] &= \left[ w\bar{x} + \bar{x}\mathbf{K}(\check{\mathbf{q}}) \quad , \quad -\check{\mathbf{q}} \cdot \bar{x} \right] [-\check{\mathbf{q}}, w] \\ &= \left[ (\check{\mathbf{q}} \cdot \bar{x})\check{\mathbf{q}} + w \left( w\bar{x} + \bar{x}\mathbf{K}(\check{\mathbf{q}}) \right) - \left( w\bar{x} + \bar{x}\mathbf{K}(\check{\mathbf{q}}) \right) \mathbf{K}(-\check{\mathbf{q}}) \quad , \quad \left( w\bar{x} + \bar{x}\mathbf{K}(\check{\mathbf{q}}) \right) \cdot \check{\mathbf{q}} - (\check{\mathbf{q}} \cdot \bar{x})w \right] \\ &= \left[ (\bar{x} \cdot \check{\mathbf{q}})\check{\mathbf{q}} + w^2\bar{x} + w\bar{x}\mathbf{K}(\check{\mathbf{q}}) + \left( w\bar{x} + \bar{x}\mathbf{K}(\check{\mathbf{q}}) \right) \mathbf{K}(\check{\mathbf{q}}) \quad , \quad \left( w\bar{x} + \bar{x}\mathbf{K}(\check{\mathbf{q}}) \right) \cdot \check{\mathbf{q}} - (\check{\mathbf{q}} \cdot \bar{x})w \right] \\ &= \left[ \bar{x}(\check{\mathbf{q}}^T\check{\mathbf{q}}) + w^2\bar{x} + w\bar{x}\mathbf{K}(\check{\mathbf{q}}) + w\bar{x}\mathbf{K}(\check{\mathbf{q}}) + \bar{x}\mathbf{K}(\check{\mathbf{q}})\mathbf{K}(\check{\mathbf{q}}) \quad , \quad \underbrace{(w\bar{x}) \cdot \check{\mathbf{q}} - (\bar{x} \wedge \check{\mathbf{q}}) \cdot \check{\mathbf{q}} - (\check{\mathbf{q}} \cdot \bar{x})w}_{=0} \right] \\ &= \left[ \bar{x}(\check{\mathbf{q}}^T\check{\mathbf{q}}) + w^2\bar{x} + 2w\bar{x}\mathbf{K}(\check{\mathbf{q}}) + \bar{x}(\check{\mathbf{q}}^T\check{\mathbf{q}} - \check{\mathbf{q}}\check{\mathbf{q}}^T\mathbf{I}) \quad , \quad 0 \right] \\ &= \left[ \bar{x} \left( (w^2 - \check{\mathbf{q}}\check{\mathbf{q}}^T)\mathbf{I} + 2w\mathbf{K}(\check{\mathbf{q}}) + 2\check{\mathbf{q}}^T\check{\mathbf{q}} \right) \quad , \quad 0 \right] \end{aligned} \quad (\text{A.19})$$

D'autre part, avec A.17 et la représentation du quaternion comme une rotation autour d'un axe représentée par A.1, nous arrivons à la description géométrique de la rotation et l'équation de rotation de Rodrigues :

$$\begin{aligned} \check{\mathbf{q}} &= \bar{u} \sin \frac{\theta}{2} \quad ; \quad w = \cos \frac{\theta}{2} \\ \mathbf{R} &= (w^2 - \check{\mathbf{q}}\check{\mathbf{q}}^T)\mathbf{I} + 2w\mathbf{K}(\check{\mathbf{q}}) + 2\check{\mathbf{q}}^T\check{\mathbf{q}} \\ &= \left( \cos^2 \frac{\theta}{2} - \sin^2 \frac{\theta}{2} \bar{u}\bar{u}^T \right) \mathbf{I} + 2 \cos \frac{\theta}{2} \sin \frac{\theta}{2} \mathbf{K}(\bar{u}) + 2 \sin^2 \frac{\theta}{2} \bar{u}^T \bar{u} \end{aligned} \quad (\text{A.20})$$

Avec les identités trigonométriques :

$$\cos^2 \frac{\theta}{2} - \sin^2 \frac{\theta}{2} = \cos \theta \quad ; \quad 2 \cos \frac{\theta}{2} \sin \frac{\theta}{2} = \sin \theta \quad ; \quad 2 \sin^2 \frac{\theta}{2} = 1 - \cos \theta$$

et le résultat de A.15 pour  $\mathbf{K}^2$  :

$$\bar{u}^T \bar{u} = \mathbf{K}(\bar{u})^2 + \bar{u}\bar{u}^T \mathbf{I} \quad ; \quad \bar{u}\bar{u}^T = 1 \quad \rightarrow \quad \bar{u}^T \bar{u} = \mathbf{K}(\bar{u})^2 + \mathbf{I} \quad (\text{A.21})$$

A.20 devient "l'équation de rotation de Rodrigues" :

$$\begin{aligned} \mathbf{R} &= \cos \theta \mathbf{I} + \sin \theta \mathbf{K}(\bar{u}) + (1 - \cos \theta) (\mathbf{K}(\bar{u})^2 + \mathbf{I}) \\ &= \mathbf{I} + \sin \theta \mathbf{K}(\bar{u}) + (1 - \cos \theta) \mathbf{K}(\bar{u})^2 \end{aligned} \quad (\text{A.22})$$

qui provient de la déduction géométrique des vecteurs de la figure A.1 :

$$\begin{aligned} \vec{p}_{rot} = \mathbf{R}\vec{p} &= (\mathbf{I} + \sin \theta \mathbf{K}(\bar{u}) + (1 - \cos \theta) \mathbf{K}(\bar{u})^2) \vec{p} \\ &= \vec{p} + \sin \theta \mathbf{K}(\bar{u}) \vec{p} + (1 - \cos \theta) \mathbf{K}(\bar{u})^2 \vec{p} \\ &= \vec{p} + \sin \theta (\bar{u} \wedge \vec{p}) + (1 - \cos \theta) (\bar{u} \wedge (\bar{u} \wedge \vec{p})) \end{aligned} \quad (\text{A.23})$$

## A.2 Quaternions doubles.

Les quaternions doubles, dual-quaternions ou quaternions de nombre double ont été introduits par E. Study (1903) en utilisant la théorie des nombres doubles de W.K. Clifford (1845-1879).

Ces quaternions doubles sont idéaux pour représenter une rotation et une translation à la différence des quaternions qui forment une rotation uniquement A.1. [Shevlin 96], [Goddard 97].

Ils sont définis par :

$$\hat{\mathbf{q}} = \mathbf{r} + \epsilon \mathbf{s} \quad (\text{A.24})$$

$$\begin{aligned}
\mathbf{q}\mathbf{R}_{row}(\mathbf{p}) &= \mathbf{p}\mathbf{L}_{row}(\mathbf{q}) = \mathbf{q}\mathbf{p} \\
\mathbf{q}\mathbf{R}_{row}(\mathbf{q})^T &= \mathbf{q}\mathbf{L}_{row}(\mathbf{q})^T = \mathbf{q}\mathbf{q}^T \\
\mathbf{R}_{row}(\mathbf{q}^T) &= \mathbf{R}_{row}(\mathbf{q})^T \\
\mathbf{L}_{row}(\mathbf{q}^T) &= \mathbf{L}_{row}(\mathbf{q})^T \\
\mathbf{R}_{row}(\mathbf{q}\mathbf{R}_{row}(\mathbf{p})) &= \mathbf{R}_{row}(\mathbf{q})\mathbf{R}_{row}(\mathbf{p}) \\
\mathbf{L}_{row}(\mathbf{q}\mathbf{L}_{row}(\mathbf{p})) &= \mathbf{L}_{row}(\mathbf{q})\mathbf{L}_{row}(\mathbf{p}) \\
\mathbf{R}_{row}(\mathbf{p})^T\mathbf{L}_{row}(\mathbf{q}) &= \mathbf{L}_{row}(\mathbf{q})\mathbf{R}_{row}(\mathbf{p})^T \\
\mathbf{R}_{row}(\mathbf{p})\mathbf{L}_{row}(\mathbf{q}) &= \mathbf{L}_{row}(\mathbf{q})\mathbf{R}_{row}(\mathbf{p}) \\
\mathbf{R}_{row}(\mathbf{q})^T\mathbf{R}_{row}(\mathbf{q}) &= \mathbf{R}_{row}(\mathbf{q})\mathbf{R}_{row}(\mathbf{q})^T = \mathbf{q}\mathbf{q}^T\mathbf{I} \\
\mathbf{L}_{row}(\mathbf{q})^T\mathbf{L}_{row}(\mathbf{q}) &= \mathbf{L}_{row}(\mathbf{q})\mathbf{L}_{row}(\mathbf{q})^T = \mathbf{q}\mathbf{q}^T\mathbf{I}
\end{aligned}$$

TAB. A.2 – Résumé des égalités entre matrices des quaternions  $\mathbf{R}_{row}(\mathbf{q})$  et  $\mathbf{L}_{row}(\mathbf{q})$ 

où  $\mathbf{r}$  et  $\mathbf{s}$  sont les quaternions simples et  $\epsilon$  est un *quasiscalar* défini dans les nombres doubles. De ce fait,  $\epsilon$  obéit à la règle  $\epsilon^2 = 0$ .

Le conjugué du quaternion double est décrit par :

$$\hat{\mathbf{q}}' = \mathbf{r}' + \epsilon \mathbf{s}' \quad (\text{A.25})$$

L'amplitude carrée est :

$$\begin{aligned}
\|\hat{\mathbf{q}}\|^2 = \hat{\mathbf{q}}\hat{\mathbf{q}}' &= (\mathbf{r} + \epsilon \mathbf{s})(\mathbf{r}' + \epsilon \mathbf{s}') = \mathbf{r}\mathbf{r}' + \epsilon \mathbf{s}\mathbf{r}' + \epsilon \mathbf{r}\mathbf{s}' + \epsilon^2 \mathbf{s}\mathbf{s}' \\
&= \mathbf{r}\mathbf{r}' + \epsilon(\mathbf{s}\mathbf{r}' + \mathbf{r}\mathbf{s}')
\end{aligned} \quad (\text{A.26})$$

qui donne un nombre double avec une partie réelle positive. L'inverse  $\hat{\mathbf{q}}^{-1} = \hat{\mathbf{q}}'/\|\hat{\mathbf{q}}\|^2$  existe seulement si la partie réelle est différente de zéro. De manière similaire aux quaternions unitaires qui représentent une rotation, les quaternions doubles unitaires représentent une rotation et une translation. La définition d'un quaternion double unitaire est d'avoir une valeur de  $1 + \epsilon 0$ . Cela donne une contrainte au niveau de l'équation A.26 :

$$\mathbf{r}\mathbf{r}' = 1 \quad , \quad \mathbf{s}\mathbf{r}' + \mathbf{r}\mathbf{s}' = 0 \quad \rightarrow \quad \vec{r} \cdot \vec{r}' = 1 \quad , \quad \vec{r} \cdot \vec{s}' = 0 \quad (\text{A.27})$$

Le but est d'obtenir une expression de la transformation 3D en fonction du quaternion double.

Le problème est qu'il n'y a pas de représentation pour un point 3D avec la translation et la rotation à la façon de  $\mathbf{X}_{rot} = \mathbf{q}\mathbf{X}\mathbf{q}^T$  [Daniilidis 99]. Nous devons utiliser la représentation d'une droite par le vecteur direction  $\vec{n}$  et par un point  $\vec{p}$ . Le produit vectoriel  $\vec{p} \wedge \vec{n}$  est le moment de la droite. Les contraintes sont  $\|\vec{n}\|^2 = 1$  et  $\vec{n}^T \vec{m} = 0$ .

Si la droite est représentée par un double quaternion comme  $\hat{\mathbf{n}} = \mathbf{n} + \epsilon \mathbf{m}$ , il doit avoir le quaternion double transformé par  $\mathbf{R} + \vec{t}$  donné par  $\hat{\mathbf{n}}_{rot} = \hat{\mathbf{q}}\hat{\mathbf{n}}\hat{\mathbf{q}}^T$ .

Dans l'espace 3D, la transformation donne :

$$\vec{n}_{rot} = \mathbf{R}\vec{n}$$



$$\begin{aligned}\vec{m}_{rot} &= \vec{p}_{rot} \wedge \vec{n}_{rot} = (\mathbf{R}\vec{p} + \vec{t}) \wedge \mathbf{R}\vec{n} = \mathbf{R}(\vec{p} \wedge \vec{n}) + \vec{t} \wedge \mathbf{R}\vec{n} \\ &= \mathbf{R}\vec{m} + \vec{t} \wedge \mathbf{R}\vec{n}\end{aligned}\quad (\text{A.28})$$

En faisant la multiplication du double quaternion, toujours avec  $\epsilon^2 = 0$  :

$$\begin{aligned}\hat{\mathbf{q}} \hat{\mathbf{n}} \hat{\mathbf{q}}^T &= (\mathbf{r} + \epsilon \mathbf{s})(\mathbf{n} + \epsilon \mathbf{m})(\mathbf{r}^T + \epsilon \mathbf{s}^T) \\ &= (\mathbf{r} \mathbf{n} + \epsilon(\mathbf{s} \mathbf{n} + \mathbf{r} \mathbf{m}))(\mathbf{r}^T + \epsilon \mathbf{s}^T) \\ &= \mathbf{r} \mathbf{n} \mathbf{r}^T + \epsilon(\mathbf{s} \mathbf{n} \mathbf{r}^T + \mathbf{r} \mathbf{m} \mathbf{r}^T + \mathbf{r} \mathbf{n} \mathbf{s}^T)\end{aligned}\quad (\text{A.29})$$

et avec la définition d'un nouveau quaternion :

$$\mathbf{s} = \mathbf{t} \mathbf{r} \quad (\text{A.30})$$

pour  $\hat{\mathbf{q}} = \mathbf{r} + \epsilon \mathbf{s}$ , où  $\vec{t}$  est la translation :

$$\begin{aligned}\hat{\mathbf{q}} \hat{\mathbf{n}} \hat{\mathbf{q}}^T &= \mathbf{r} \mathbf{n} \mathbf{r}^T + \epsilon(\mathbf{r} \mathbf{m} \mathbf{r}^T + \mathbf{t} \mathbf{r} \mathbf{n} \mathbf{r}^T + \mathbf{r} \mathbf{n} (\mathbf{t} \mathbf{r})^T) \\ &= \mathbf{r} \mathbf{n} \mathbf{r}^T + \epsilon(\mathbf{r} \mathbf{m} \mathbf{r}^T + (\mathbf{t} \mathbf{r} \mathbf{n} \mathbf{r}^T + \mathbf{r} \mathbf{n} \mathbf{r}^T \mathbf{t}^T))\end{aligned}\quad (\text{A.31})$$

où on peut voir que chaque élément correspond à :

$$\begin{aligned}\vec{n}_{rot} = \mathbf{R}\vec{n} &\Leftrightarrow \mathbf{n}_{rot} = \mathbf{r} \mathbf{n} \mathbf{r}^T \\ \vec{m}_{rot} = \mathbf{R}\vec{m} &\Leftrightarrow \mathbf{m}_{rot} = \mathbf{r} \mathbf{m} \mathbf{r}^T\end{aligned}\quad (\text{A.32})$$

Le produit vectoriel en fonction des quaternions réels est :

$$\begin{aligned}\mathbf{t} &= \frac{1}{2} [\vec{t}, 0] \quad , \quad \mathbf{n}_{rot} = [\vec{n}_{rot}, 0] \\ \mathbf{t} * \mathbf{n}_{rot} &= \frac{1}{2} [\vec{t} \wedge \vec{n}_{rot}, -\vec{t} \cdot \vec{n}_{rot}] \\ \mathbf{n}_{rot} * \mathbf{t}^T &= \frac{1}{2} [-\vec{n}_{rot} \wedge \vec{t}, \vec{n}_{rot} \cdot \vec{t}] \\ (\mathbf{t} * \mathbf{n}_{rot} + \mathbf{n}_{rot} * \mathbf{t}^T) &= [\vec{t} \wedge \vec{n}_{rot}, 0]\end{aligned}\quad (\text{A.33})$$

et finalement :

$$\hat{\mathbf{q}} \hat{\mathbf{n}} \hat{\mathbf{q}}^T = \mathbf{n}_{rot} + \epsilon(\mathbf{m}_{rot} + (\mathbf{t} \mathbf{n}_{rot} + \mathbf{n}_{rot} \mathbf{t}^T)) = \mathbf{n}_{rot} + \epsilon(\mathbf{m}_{rot} + [\vec{t} \wedge \vec{n}_{rot}, 0]) \quad (\text{A.34})$$

qui correspond au résultat trouvé dans l'espace 3D.

En complément, [Walker 91] a fait une analogie entre le quaternion double  $\hat{\mathbf{q}} = \mathbf{r} + \epsilon \mathbf{s}$  et le quaternion simple avec la formule A.1 de la rotation autour d'un vecteur  $\vec{n}$  :

$$\hat{\mathbf{q}} = \left[ \sin \frac{\hat{\theta}}{2} \hat{\vec{n}}, \cos \frac{\hat{\theta}}{2} \right] \quad (\text{A.35})$$

où  $\hat{\theta}$  est un angle double et  $\hat{\vec{n}}$  est un vecteur double de la forme suivante :

$$\hat{\theta} = \theta + \epsilon t \quad , \quad \hat{\vec{n}} = \vec{n} + \epsilon(\vec{p} \wedge \vec{n}) \quad (\text{A.36})$$

$\vec{n}$  est le vecteur unitaire de l'axe de rotation et donne aussi la direction de la translation. L'axe de rotation passe par un point  $\vec{p}$ ; la transformation fait une rotation  $\theta$  autour de l'axe et une translation  $t$  le long de cet axe.

En développant A.35 avec A.36 :

$$\hat{\mathbf{q}} = \left[ \sin \frac{\theta + \epsilon t}{2} (\vec{n} + \epsilon (\vec{p} \wedge \vec{n})), \cos \frac{\theta + \epsilon t}{2} \right] \quad (\text{A.37})$$

avec les identités trigonométriques :

$$\sin \frac{\theta + \epsilon t}{2} = \sin \frac{\theta}{2} + \epsilon \frac{t}{2} \cos \frac{\theta}{2} \quad , \quad \cos \frac{\theta + \epsilon t}{2} = \cos \frac{\theta}{2} - \epsilon \frac{t}{2} \sin \frac{\theta}{2} \quad (\text{A.38})$$

A.37 devient :

$$\begin{aligned} \hat{\mathbf{q}} &= \left[ \left( \sin \frac{\theta}{2} + \epsilon \frac{t}{2} \cos \frac{\theta}{2} \right) (\vec{n} + \epsilon (\vec{p} \wedge \vec{n})), \cos \frac{\theta}{2} - \epsilon \frac{t}{2} \sin \frac{\theta}{2} \right] \\ &= \left[ \sin \frac{\theta}{2} \vec{n} + \epsilon \frac{t}{2} \cos \frac{\theta}{2} \vec{n} + \epsilon \sin \frac{\theta}{2} (\vec{p} \wedge \vec{n}), \cos \frac{\theta}{2} - \epsilon \frac{t}{2} \sin \frac{\theta}{2} \right] \\ &= \left[ \sin \frac{\theta}{2} \vec{n}, \cos \frac{\theta}{2} \right] + \epsilon \left[ \frac{t}{2} \cos \frac{\theta}{2} \vec{n} + \sin \frac{\theta}{2} (\vec{p} \wedge \vec{n}), -\frac{t}{2} \sin \frac{\theta}{2} \right] \end{aligned} \quad (\text{A.39})$$

donc :

$$\begin{aligned} \mathbf{r} &= \left[ \sin \frac{\theta}{2} \vec{n}, \cos \frac{\theta}{2} \right] \\ \mathbf{s} &= \left[ \frac{t}{2} \cos \frac{\theta}{2} \vec{n} + \sin \frac{\theta}{2} (\vec{p} \wedge \vec{n}), -\frac{t}{2} \sin \frac{\theta}{2} \right] \end{aligned} \quad (\text{A.40})$$

Les résultats de A.30 et A.31 sont utiles pour arriver à une représentation matricielle avec des quaternions doubles comme déjà déduit pour les quaternions simples en A.11.

Pour le quaternion double :  $\hat{\mathbf{q}} = \mathbf{r} + \epsilon \mathbf{s}$  la rotation est donnée par :

$$\mathbf{r} \mathbf{X} \mathbf{r}^T = \mathbf{X} \mathbf{R}_{row}(\mathbf{r})^T \mathbf{L}_{row}(\mathbf{r}) = \mathbf{X} \mathbf{Q}_{row}(\mathbf{r}) \quad (\text{A.41})$$

et la translation est calculée à partir de A.30 et transformée pour le côté droit avec A.3 :

$$\mathbf{s} = \mathbf{t} \mathbf{r} \Rightarrow \mathbf{t} = \mathbf{s} \mathbf{r}^T = \mathbf{s} \mathbf{R}_{row}(\mathbf{r})^T \quad (\text{A.42})$$

ce qui finalement permet d'écrire

$$\mathbf{R}\vec{x} + \vec{t} \Rightarrow \mathbf{X} \mathbf{R}_{row}(\mathbf{r})^T \mathbf{L}_{row}(\mathbf{r}) + \mathbf{s} \mathbf{R}_{row}(\mathbf{r})^T \quad (\text{A.43})$$



## Annexe B

# Pseudo-code pour la fonction ICP

Nous présentons en figure B.1, un pseudo-code sur la méthode d'appariement entre points 3D, méthode présentée sous forme d'un organigramme en section 2.5.

```
void Voisin_Cluster(int prof,int rayon,OptimPoints p0,OptimPoints p1)
/* prof : profondeur d'empilement */
/* rayon : nombre de clusters voisins a visiter */
/* p0 p1 : listes de points des nuages a recalcr */
5 {
  /* Initialisation du vecteur de distances */
  for(i=0;i<p0.nDots;i++) { Dmtmp[i].i=i; for(j=0;j<depth;j++) Dmtmp[i].dmin[j]=DMAX; }

  /* Balayage de la liste de clusters*/
10 for(ii=0;ii<Bv.nbul;ii++) {
  /* obtention d'indices spatiales du cluster global : */
  i0=Bul[ii].ix; j0=Bul[ii].iy; k0=Bul[ii].iz;

  gi0=bucket_check(i0,j0,k0,0); /* recherche du cluster du nuage 0 */
15 if(gi0!=-1) {
  bin0=Bu[i0][j0][k0].v[gi0]; /* indice du cluster 0 */
  npt0=B[0][bin0].npt; /* nombre des points dans le cluster du nuage */

  /* balayage de voisinage r du cluster avec des indices i0 j0 k0 : */
20 for(i1=i0-rayon;i1<=i0+rayon;i1++)
  for(j1=j0-rayon;j1<=j0+rayon;j1++)
  for(k1=k0-rayon;k1<=k0+rayon;k1++)
  /* verification : cluster reste a l'interieur de l'espace : */
  if( i1>=0 && i1<Bv.n[0] &&
25 j1>=0 && j1<Bv.n[1] &&
  k1>=0 && k1<Bv.n[2] && Bu[i1][j1][k1].npt!=0)
  {
  gi1=bucket_check(i1,j1,k1,1); /* recherche du cluster du nuage 1 */
  if(gi1!=-1) {
30 bin1=Bu[i1][j1][k1].v[gi1]; /* indice du cluster 1 */
  npt1=B[1][bin1].npt;
  /* Processus de comparaison :
  calcul de distances entre points des nuages de B[1][bind1] et B[0][bind0] */
  for(i3=0;i3<npt0;i3++) { /* boucle central des points */
35 pin0=B[0][bin0].v[i3]; /* indice du point */
  for(i4=0;i4<npt1;i4++) { /* boucle de voisinage */
  pin1=B[1][bin1].v[i4];
  /* difference des vecteurs : */
  x=p0.x[pin0] - p1.x[pin1];
40 y=p0.y[pin0] - p1.y[pin1];
  z=p0.z[pin0] - p1.z[pin1];
  d=sqrt(x*x+y*y+z*z); /* calcul de la distance : */
  /* Comparaison avec les minima d'empilement : */
  for(i6=0;i6<prof;i6++) {
45 if(d<Dmtmp[pin0].dmin[i6]) {
```

```

    for(i5=prof-1;i5>=i6+1;i5--) {
        Dmtmp[pin0].dmin[i5]=Dmtmp[pin0].dmin[i5-1];
        Dmtmp[pin0].imin[i5]=Dmtmp[pin0].imin[i5-1]; }
    Dmtmp[pin0].dmin[i6]=d; Dmtmp[pin0].imin[i6]=pin1;
50     break; }
        } } } }
    } } }
}

```

Nous présentons en figure B.2, un pseudo-code sur la stratégie d'itérations de l'algorithme ICP, stratégies présentées sous forme d'un organigramme en section 2.7.

```

int icp_main()
{
    Initialize_Matrices();
    Initialize_Clusters_Uni(bsize);
5   Creation_Clusters(p1);
    iter=0;
    do {
        Creation_Clusters(p2);
        Creation_Clusters_Uni();
10   Voisin_Cluster( prof,rayon,p1,p2 ); /* cherche appairements */

        /* change de modes de fontionnement d'algorithme */
        if(algoflag==1) {
            if(veci>2) veci=2;
15   bucket_filter1( prof,prof*2-3 );
            prof=4; }
        if(algoflag==2) {
            veci=1;
            bucket_filter1( prof,prof*2-3 );
20   prof=3; }
        dmax=Zhang( dista,delta,rang ); /* zhang filter */

        errv=*err;
        Cal_Transformation(&p1, &plist,poid, m,err); /* Transformation 3D */
25

        /* application de la transformation : */
        for(i=0;i<np2;i++) change_repere_xmt( &p2.x[i],&p2.y[i],&p2.z[i],m );
        iter++;

30   /* ----- */
        if((algoflag==2) && (actcam2==1) ) /* arrete automatique : */
            if( ( (*err-errv)<0 ) && ( _ABSi((*err-errv)/errinit*100.)<perc2 ) ) break;
        /* change automatique des modes de fontionnement d'algorithme : */
        if( (actcam1==1) && (errinit!=0) && (iter>0) && ((*err-errv)<0) )
35   {
            slope_percentage=_ABSi((*err-errv)/errinit*100.);
            if( algoflag==0 && slope_percentage<0.7 ) algoflag=1;
            else if( algoflag==1 && slope_percentage<perc1 ) {
                algoflag=2;
                Initialize_Clusters_Uni(bsize);
                Creation_Clusters(p1); }
            else if( algoflag==2 && slope_percentage<perc2 ) break;
        }
        /* ----- */
45   } while(iter<itermax);
}

```

---

```

void Voisin_Cluster(int prof,int rayon,OptimPoints p0,OptimPoints p1)
/* prof : profondeur d'empilement */
/* rayon : nombre de clusters voisins a visiter */
/* p0 p1 : listes de points des nuages a recalculer */
5 {
/* Initialisation du vecteur de distances */
for(i=0;i<p0.nDots;i++) { Dmtmp[i].i=i; for(j=0;j<depth;j++) Dmtmp[i].dmin[j]=DMAX; }

/* Balayage de la liste de clusters*/
10 for(ii=0;ii<Bv.nbul;ii++) {
/* obtention d'indices spatiales du cluster global : */
i0=Bul[ii].ix; j0=Bul[ii].iy; k0=Bul[ii].iz;

gi0=bucket_check(i0,j0,k0,0); /* recherche du cluster du nuage 0 */
15 if(gi0!=-1) {
bin0=Bu[i0][j0][k0].v[gi0]; /* indice du cluster 0 */
npt0=B[0][bin0].npt; /* nombre des points dans le cluster du nuage */

/* balayage de voisinage r du cluster avec des indices i0 j0 k0 : */
20 for(i1=i0-rayon;i1<=i0+rayon;i1++)
for(j1=j0-rayon;j1<=j0+rayon;j1++)
for(k1=k0-rayon;k1<=k0+rayon;k1++)
/* verification : cluster reste a l'interieur de l'espace : */
if( i1>=0 && i1<Bv.n[0] &&
25 j1>=0 && j1<Bv.n[1] &&
k1>=0 && k1<Bv.n[2] && Bu[i1][j1][k1].npt !=0)
{
gi1=bucket_check(i1,j1,k1,1); /* recherche du cluster du nuage 1 */
if(gi1!=-1) {
30 bin1=Bu[i1][j1][k1].v[gi1]; /* indice du cluster 1 */
npt1=B[1][bin1].npt;
/* Processus de comparaison :
calcul de distances entre points des nuages de B[1][bind1] et B[0][bind0] */
for(i3=0;i3<npt0;i3++) { /* boucle central des points */
35 pin0=B[0][bin0].v[i3]; /* indice du point */
for(i4=0;i4<npt1;i4++) { /* boucle de voisinage */
pin1=B[1][bin1].v[i4];
/* difference des vecteurs : */
x=p0.x[pin0] - p1.x[pin1];
40 y=p0.y[pin0] - p1.y[pin1];
z=p0.z[pin0] - p1.z[pin1];
d=sqrt(x*x+y*y+z*z); /* calcul de la distance : */
/* Comparaison avec les minima d'empilement : */
for(i6=0;i6<prof;i6++) {
45 if(d<Dmtmp[pin0].dmin[i6]) {
for(i5=prof-1;i5>=i6+1;i5--) {
Dmtmp[pin0].dmin[i5]=Dmtmp[pin0].dmin[i5-1];
Dmtmp[pin0].imin[i5]=Dmtmp[pin0].imin[i5-1]; }
Dmtmp[pin0].dmin[i6]=d; Dmtmp[pin0].imin[i6]=pin1;
50 break; }
} } } }
} } }
}

```

---

FIG. B.1 – L'algorithme d'appariement basé sur des espaces de *clusters*

```

int icp_main()
{
  Initialize_Matrices();
  Initialize_Clusters_Uni(bsize);
5  Creation_Clusters(p1);
  iter=0;
  do {
    Creation_Clusters(p2);
    Creation_Clusters_Uni();
10   Voisin_Cluster( prof,rayon,p1,p2 ); /* cherche appairements */

    /* change de modes de fontionnement d'algorithmme */
    if(algorithme==1) {
      if(veci>2) veci=2;
15   bucket_filter1( prof,prof*2-3 );
      prof=4; }
    if(algorithme==2) {
      veci=1;
      bucket_filter1( prof,prof*2-3 );
20   prof=3; }
    dmax=Zhang( dista,delta,rang ); /* zhang filter */

    errv=*err;
    Cal_Transformation(&p1, &plist,poid, m,err); /* Transformation 3D */
25

    /* application de la transformation : */
    for(i=0;i<np2;i++) change_repere_xmt( &p2.x[i],&p2.y[i],&p2.z[i],m );
    iter++;

30   /* ----- */
    if((algorithme==2) && (actcam2==1) ) /* arrete automatique : */
      if( ( (*err-errv)<0 ) && ( _ABSi((*err-errv)/errinit*100.)<percam2 ) ) break;
    /* change automatique des modes de fontionnement d'algorithmme : */
    if( ( actcam1==1 ) && ( errinit !=0 ) && ( iter>0 ) && ((*err-errv)<0) )
35   {
      slope_percentage=_ABSi((*err-errv)/errinit*100.);
      if( algorithme==0 && slope_percentage<0.7 ) algorithme=1;
      else if( algorithme==1 && slope_percentage<percam1 ) {
        algorithme=2;
40       Initialize_Clusters_Uni(bsize);
        Creation_Clusters(p1); }
      else if( algorithme==2 && slope_percentage<percam2 ) break;
    }
    /* ----- */
45  } while(iter<itermax);
}

```

---

FIG. B.2 – L'algorithmme d'ICP

# Bibliographie

- [3DDB sh] 3DDB. *Range Image Database*. Ohio State University (MSU/WSU), <http://sampl.eng.ohio-state.edu/sampl/data/3DDB/RID/index.htm>, /.
- [3Dscanrep sh] 3Dscanrep. *3D scanning repository*. Stanford, <http://graphics.stanford.edu/data/3Dscanrep/>, /.
- [Ahmed 97] M. N. Ahmed, S. M. Yamany & A. A. Farag. *Fast Algorithm for Registration of Free-Form Curves and Surfaces*. In IEEE International Conference on Image Processing (ICIP '97), Santa Barbara, USA, janvier 1997.
- [And 98] O. J. And. *Statistical Analysis of Two 3-D Registration and Modeling Strategies*. ISPRS Journal of Photogrammetry and Remote Sensing, vol. 53, no. 6, pages 320–341, décembre 1998.
- [Arun 87] K. Arun, T. Huang & S. Blostein. *Least-squares fitting of two 3-D points sets*. IEEE Transactions Pattern Analysis and Machine Intelligence, vol. 9, no. 5, pages 698–700, septembre 1987.
- [Attali 97] D. Attali. *r-Regular Shape Reconstruction from Unorganized Points*. In Symposium on Computational Geometry, pages 248–253, 1997.
- [Bajaj 95] C. L. Bajaj, F. Bernardini & G. Xu. *Automatic Reconstruction of Surfaces and Scalar Fields From 3D Scans*. Computer Graphics, vol. 29, pages 109–118, 1995.
- [Bajaj 97] C. L. Bajaj, F. Bernardini & G. Xu. *Reconstructing Surfaces and Functions on Surfaces from Unorganized Three-Dimensional Data*. Algorithmica, vol. 19, no. 1/2, pages 243–261, 1997.
- [Banta 95] J.E. Banta, Y. Zhien, X.Z. Wang, G. Zhang, M.T. Smith & M.A. Abidi. *A “Best-Next-View” algorithm for three-dimensional scene reconstruction using range images*. In Proc. of SPIE Conf. on Intelligent Robots and Computer Vision XIV, Philadelphia, PA, volume 2588, pages 418–429, octobre 1995.
- [Bellon 99] O. Pereira Bellon, A. Direne & L. Silva. *Edge Detection to Guide Range Image Segmentation by Clustering Techniques*. In IEEE Int. Conf. on Image Processing, Kobe, Japan, octobre 1999.
- [Benjemaa 97] R. Benjemaa & F. Schmitt. *Fast global registration of 3d sampled surfaces using a multi-z-buffer technique*. In International Conference on 3-D Digital Imaging and Modeling (3DIM97), Ottawa, Canada, pages 113–120, mai 1997.
- [Bentley 75] J. L. Bentley. *Multidimensional binary search trees used for associative searching*. Communications of the Association for Computing Machinery, vol. 9, no. 18, page 509–517, 1975.
- [Bergevin 95] R. Bergevin, D. Laurendeau & D. Poussart. *Registering Range Views of Multi-Part Objects*. Computer Vision and Image Understanding (CVIU), no. 61, pages 1–16, 1995.
- [Bergevin 96] R. Bergevin, M. Soucy, H. Gagnon & D. Laurendeau. *Towards a General Multi-View Registration Technique*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 18, no. 5, pages 540–547, 1996.



- [Bernardini 99] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva & G. Taubin. *The Ball-Pivoting Algorithm for Surface Reconstruction*. IEEE Transactions on Visualization and Computer Graphics, vol. 5, no. 4, pages 349–359, 1999.
- [Besl 92] P. J. Besl & N. D. McKay. *A Method for Registration of 3-D Shapes*. IEEE Transactions Pattern Analysis and Machine Intelligence, vol. 14, no. 2, pages 239–256, février 1992.
- [Blais 95] G. Blais & M. D. Levine. *Registering Multiview Range Data to Create 3D Computer Objects*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 17, no. 8, pages 820–824, août 1995.
- [Boissonnat 00] J.D. Boissonnat & F. Cazals. *Smooth surface reconstruction via natural neighbour interpolation of distance functions*. In Symposium on Computational Geometry, pages 223–232, 2000.
- [Bozier 00a] J. Bozier. *Construction incrémentale de modèles géométriques d'objets dans un contexte robotique*. Thèse de doctorat, Thèse Université Paul Sabatier, Toulouse, Rapport LAAS n.00086, mars 2000.
- [Bozier 00b] J. Bozier, M. Devy & A. Sappa. *A Geometrical Approach for the Incremental Modelling of Free Form Surface by Triangular Meshes*. In 8th International Symposium on Intelligent Robotic Systems (SIRS'2000), Reading, UK, Rapport LAAS N.00332, pages 13–21, juillet 2000.
- [Burtnyk 95] N. Burtnyk & M. A. Greenspan. *Multiple view registration of range data using signature search*. In American Nuclear Society Sixth Topical Meeting on Robotics and Remote Systems, Monterey, CA, pages 745–753, février 1995.
- [Carmichael 99] O. Carmichael, D. F. Huber & M. Hebert. *Large Data Sets and Confusing Scenes in 3-D surface Matching and Recognition*. In 2nd International Conference on 3-D Digital Imaging and Modeling (3DIM99), Ottawa, Canada, pages 358–367, octobre 1999.
- [Chen 92] Xin Chen. *Modélisation géométrique par vision artificielle*. Thèse de doctorat, These de l'Ecole Nationale Supérieure des Télécommunication, Paris, 1992.
- [Chen 98] C.-S. Chen, Y.-P. Hung & J.-B. Cheng. *A Fast Automatic Method for Registration of Partially-Overlapping Range Images*. In 6th IEEE International Conference on Computer Vision (ICCV1998), Bombay, India, page 242, 1998.
- [Chetverikov 02] D. Chetverikov, D. Svirko, D. Stepanov & P. Krsek. *The Trimmed Iterative Closest Point Algorithm*. In 16th International Conference on Pattern Recognition (ICPR 2002), Quebec City, Canada, pages 545–548, 2002.
- [Chua 96] C. S. Chua & R. Jarvis. *3D Free-Form Surface Registration and Object Recognition*. International Journal of Computer Vision, vol. 17, pages 119–149, 1996.
- [Ciampalini 97] A. Ciampalini *et al.* *Multiresolution decimation based on global error*. The Visual Computer, vol. 13(5), juin 1997.
- [Colly 89] J. Colly, M. Devy, M. Ghallab & L.H. Pampagnin. *Sensory control for 3d environment modelling*. In International Advanced Robotics Programme, 1st workshop on multi-sensor fusion and environment modelling, Toulouse, pages 1–14, 1989.
- [Coutaller 03] P. Le Coutaller. *Modélisation par des surfaces déformables*. Rapport de dess, Université Paul Sabatier, septembre 2003.
- [Curless 96] B. Curless & M. Levoy. *A Volumetric Method for Building Complex Models from Range Images*. Computer Graphics, vol. 30, no. Annual Conference Series, pages 303–312, 1996.
- [Dalley 02] G. Dalley & P. Flynn. *Pair-wise range image registration : a study in outlier classification*. Computer Vision and Image Understanding (CVIU), vol. 87, no. 1/2/3, pages 104–115, 2002.

- 
- [Daniilidis 99] K. Daniilidis. *Hand-eye calibration using dual quaternions*. International Journal of Robotics Research, vol. 18, pages 286–298, 1999.
- [Delingette 94] H. Delingette. *Simplex Meshes : a General Representation for 3D Shape Reconstruction*. Rapport de recherche rr-2214, INRIA, Sophia Antipolis, France, mars 1994.
- [Delingette 97] H. Delingette. *General Object Reconstruction based on Simplex Meshes*. Rapport de recherche rr-3111, INRIA, Sophia Antipolis, France, février 1997.
- [Dias 03] P. Dias, V. Sequeira, J.G.M. Gonçalves & F. Vaz. *Registration and Fusion of Intensity and Range Data for 3D Modelling of Real World Scenes*. In Proc. Fourth International Conference on 3-D Digital Imaging and Modeling,, octobre 2003.
- [Dorai 97] C. Dorai, J. Weng & A. K. Jain. *Optimal Registration of Object Views Using Range Data*. IEEE Transactions Pattern Analysis and Machine Intelligence, vol. 19, no. 10, pages 1131–1138, 1997.
- [Dorai 98] C. Dorai, G. Wang, A. K. Jain & C. Mercer. *Registration and Integration of Multiple Object Views for 3D Model Construction*. IEEE Transactions Pattern Analysis and Machine Intelligence, vol. 20, no. 1, pages 83–89, 1998.
- [Eggert 96] D. W. Eggert, A. W. Fitzgibbon & R. B. Fisher. *Simultaneous registration of multiple range views for use in reverse engineering*. In 13th International Conference on Pattern Recognition (ICPR 1996), Vienna, Austria, pages 243–247, août 1996.
- [Feldmar 94] J. Feldmar, N. Ayache & F. Betting. *3D-2D projective registration of free-form curves and surfaces*. Rapport technique 2434, INRIA, décembre 1994.
- [Feldmar 96] J. Feldmar & N. Ayache. *Rigid, Affine and Locally Affine Registration of Free-Form Surfaces*. International Journal of Computer Vision, vol. 18, no. 2, mai 1996.
- [Feldmar 97] J. Feldmar, J. Declerck, G. Malandain & N. Ayache. *Extension of the ICP Algorithm to Non-Rigid Intensity-Based Registration of 3D Volumes*. Computer Vision and Image Understanding (CVIU), vol. 66, no. 2, pages 193–206, mai 1997.
- [Fischer 99] D. Fischer, P. Kohlhepp & F. Bulling. *An evolutionary algorithm for the registration of 3-d surface representations*. In Pattern Recognition, pages 53–69, 1999.
- [Fisher 97] R. Fisher, A. Fitzgibbon & D. Eggert. *Extracting Surface Patches from Complete Range Descriptions*. Proc. International Conference on Recent Advances in 3-D Digital Imaging and Modeling, Ottawa, pages 148–155, mai 1997.
- [Fisher 04] R. B. Fisher. *Applying knowledge to reverse engineering problems*. In Computer-Aided Design, volume 36, pages 501–510, 2004.
- [Floriani 88] L. De Floriani & E. Puppo. *Constrained Delaunay Triangulation for Multiresolution Surface Description*. In International Conference on Pattern Recognition, novembre 1988.
- [Frey 97] P.J. Frey & H. Borouchaki. *Surface mesh evaluation*. In Proceedings, 6th International Meshing Roundtable, Sandia National Laboratories, pages 363–373, octobre 1997.
- [Garcia 01] D. Garcia, J.J. Orteu & M. Devy. *Calibrage précis d'un capteur de vision stéréoscopique*. In Congrès Francophone de Vision par Ordinateur (ORASIS 2001), Cahors (France), Rapport LAAS N.01037, pages 107–116, juin 2001.
- [García 98] M. A. García, S. Velázquez & A. D. Sappa. *A Two-Stage Algorithm for Planning the Next View From Range Images*. In British Machine Vision Conference, 1998.
- [Gelfand 03] N. Gelfand, L. Ikemoto, S. Rusinkiewicz & M. Levoy. *Geometrically Stable Sampling for the ICP Algorithm*. In 4th International Conference on 3-D Digital Imaging and Modeling (3DIM2003), Banff, Canada, octobre 2003.
- [Goddard 97] J. S. Goddard. *Pose and Motion Estimation from Vision using Dual Quaternion-based Extended Kalman Filtering*. Thèse de doctorat, University of Tennessee, 1997.

- [Godin 01] G. Godin, D. Laurendeau & R. Bergevin. *A Method for the Registration of Attributed Range Images*. In 3rd International Conference on 3-D Digital Imaging and Modeling (3DIM2001), Québec City, Canada, pages 179–186, Québec, Canada, mai 2001.
- [González-Baños 00] H. González-Baños & J.-C. Latombe. *Robot Navigation for Automatic Model Construction using Safe Regions*. In Proc. of Int. Symposium on Experimental Robotics (ISER'01), décembre 2000.
- [Gourdon 93] A. Gourdon & N. Ayache. *Registration of a curve on a surface using differential properties*. Research report RR-2145, INRIA, décembre 1993.
- [Granger 02] S. Granger & X. Pennec. *Multi-scale EM-ICP : A Fast and Robust Approach for Surface Registration*. In 7th European Conference on Computer Vision (ECCV 2002), Copenhagen, Denmark, volume 2353 of *LNCS*, pages 418–432, 2002.
- [Greenspan 01] M. A. Greenspan & G. Godin. *A Nearest Neighbor Method for Efficient ICP*. In 3rd International Conference on 3-D Digital Imaging and Modeling (3DIM2001), Québec City, Canada, mai 2001.
- [Greenspan 03] M. A. Greenspan & M. Yurick. *An Approximate K-D Tree Search for Efficient ICP*. In 4th International Conference on 3-D Digital Imaging and Modeling (3DIM2003), Banff, Canada, octobre 2003.
- [Guéziec 92] A. Guéziec & N. Ayache. *Smoothing and Matching of 3-D Space Curves*. In 2nd European Conference on Computer Vision, Santa Margherita Ligure (Italy), 1992.
- [Haagensen 00] E. Haagensen & H. Lincoln. *The templars secret island*. The Windrush Press, Gloucestershire, 2000.
- [Hähnel 03] D. Hähnel, S. Thrun & W. Burgard. *An Extension of the ICP Algorithm for Modeling Nonrigid Objects with Mobile Robots*. In International Joint Conference on Artificial Intelligence (IJCAI), 2003.
- [Haralick 89] R. Haralick, H. Joo, C. Lee, X. Zhuang, V. Vaidya & M.B. Kim. *Pose Estimation from Corresponding Point Data*. In Machine Vision for Inspection and Measurement, 1989.
- [Hebert 95] M. Hebert, R. Hoffman, A. Johnson & J. Osborn. *Sensor Based Interior Modeling*. In American Nuclear Society 6th Topical Meeting on Robotics and Remote Systems (ANS '95), pages 731–737, février 1995.
- [Hebert 98] M. Hebert & O. Carmichael. *Unconstrained Registration of Large 3D Points Sets for Complex Model Building*. In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS1998), Victoria, Canada, pages 360–367, octobre 1998.
- [Hebert 01] M. Hebert, M. Deans, D. F. Huber, B. Nabbe & N. Vandapeland. *Progress in 3-D Mapping and Localization*. In 9th International Symposium on Intelligent Robotic Systems (SIRS'2001), Toulouse, France, 2001.
- [Hernández 04] C. Hernández & F. Schmitt. *Une approche par modèle déformable pour la reconstruction 3D de haute qualité d'objets photographiés*. In 14ème Congrès Francophone AFRIF-AFIA de Reconnaissance des Formes et Intelligence Artificielle (RFIA'2004), Toulouse (France), 2004.
- [Higuchi 93] K. Higuchi, M. Hebert & K. Ikeuchi. *Building 3-D Models from Unregistered Range Images*. Rapport technique CMU-CS-93-214, Computer Science Department, Carnegie Mellon University, Pittsburgh, PA, novembre 1993.
- [Hoover 96] A. Hoover & al. *An Experimental Comparison of Range Image Segmentation Algorithms*. IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 18, no. 7, pages 673–689, juillet 1996.
- [Hoppe 92] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald & W. Stuetzle. *Surface reconstruction from unorganized points*. Computer Graphics, vol. 26, no. 2, pages 71–78, 1992.

- 
- [Hoppe 94] H. Hoppe. *Surface Reconstruction from Unorganized Points*. Thèse de doctorat, University of Washington, 1994.
- [Horn 87] B. K. P. Horn. *Closed-Form Solution of Absolute Orientation Using Unit Quaternions*. Journal Optical Society America, vol. 4, no. 4, pages 629 – 642, avril 1987.
- [Huber 99] D. F. Huber & M. Hebert. *A New Approach to 3-D Terrain Mapping*. In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS1999), Piscataway, USA, pages 1121–1127. IEEE, octobre 1999.
- [Huber 00] D. F. Huber, O. Carmichael & M. Hebert. *3D map reconstruction from range data*. In IEEE International Conference on Robotics and Automation (ICRA2000), San Francisco, USA, volume 1, pages 891–897, avril 2000.
- [Huber 01a] D. F. Huber. *Automatic 3D Modeling Using Range Images Obtained from Unknown Viewpoints*. In 3rd International Conference on 3-D Digital Imaging and Modeling (3DIM2001), Québec City, Canada, pages 153–160. IEEE Computer Society, mai 2001.
- [Huber 01b] D. F. Huber & M. Hebert. *Fully Automatic Registration of Multiple 3D Data Sets*. In IEEE Computer Society Workshop on Computer Vision Beyond the Visible Spectrum (CVBVS 2001), décembre 2001.
- [Huber 02] D. F. Huber. *Automatic Three-dimensional Modeling From Reality*. Thèse de doctorat, Carnegie Mellon University, 2002.
- [Huber 03] D. F. Huber & M. Hebert. *3D Modeling Using a Statistical Sensor Model and Stochastic Search*. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2003), Madison, Wisconsin, pages 858–865, juin 2003.
- [Hung 99] Y.-P. Hung, C.-S. Chen, I.-B. Hsieh & C.-S. Fuh. *Reconstruction of complete 3D object model from multiview range images*. In SPIE Proceedings, pages 138–145, janvier 1999.
- [Jiang 94] X. Jiang & H. Bunke. *Fast Segmentation of Range Images into Planar Regions by Scan Line Grouping*. Machine Vision and Applications, vol. 7, no. 2, pages 115–122, 1994.
- [Jiang 98] X. Jiang & H. Bunke. *Range Image Segmentation : Adaptive Grouping of Edges into Regions*. In Computer Vision-ACCV'98 (R. Chin and T. Pong, Eds.), Springer-Verlag, Berlin/New York, pages 299–306, 1998.
- [Jiang 99] X. Jiang & H. Bunke. *Edge Detection in Range Images Based on Scan Line Approximation*. Computer Vision and Image Understanding, vol. 73, no. 2, pages 183–199, février 1999.
- [Johnson 95] A. Johnson, P. Leger, R. Hoffman, M. Hebert & J. Osborn. *3-D object modeling and recognition for telerobotic manipulation*. In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS1995) Pittsburgh, USA, volume 1, pages 103 – 110, août 1995.
- [Johnson 96] A. Johnson & S. Kang. *Registration and integration of textured 3-D data*. 25 tech. report crl 96/4, Digital Equipment Corp. Cambridge Research Lab., octobre 1996.
- [Johnson 97a] A. Johnson & M. Hebert. *Surface registration by matching oriented points*. In International Conference on 3-D Digital Imaging and Modeling (3DIM97), Ottawa, Canada, mai 1997.
- [Johnson 97b] A. Johnson, R. Hoffman, J. Osborn & M. Hebert. *A system for semi-automatic modeling of complex environments*. In International Conference on 3-D Digital Imaging and Modeling (3DIM97), Ottawa, Canada, mai 1997.
- [Johnson 99] A. Johnson & M. Hebert. *Using spin Images for efficient object recognition in cluttered 3D scenes*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 21, no. 5, mai 1999.

- [Jung 04] Il-Kyun Jung & S. Lacroix. *Cartographie haute résolution à partir d'images stéréoscopiques acquises à basses altitudes*. In 14ème Congrès Francophone AFRIF-AFIA de Reconnaissance des Formes et Intelligence Artificielle (RFIA'2004), Toulouse (France), 2004.
- [Kamgar-Parsi 91] B. Kamgar-Parsi, J. L. Jones & A. Rosenfeld. *Registration of Multiple Overlapping Range Images : Scenes Without Distinctive Features*. IEEE Transactions Pattern Analysis and Machine Intelligence, vol. 13, no. 9, pages 857–871, 1991.
- [Kass 88] M. Kass, A. Witkin & D. Terzopoulos. *Snakes : Active contour models*. International Journal of Computer Vision, vol. 1, no. 4, pages 321–331, 1988.
- [Keller 03] Y. Keller & A. Averbuch. *Implicit similarity : a new approach to multi-sensor image registration*. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2003), Madison, Wisconsin, juin 2003.
- [Kim 04] S. Kim, K. Kim & W. Woo. *3D registration for image-based virtual environment generation using color and depth information*. In IEEE International Conference on Image Processing (ICIP 2004), Singapore, 2004.
- [Klein 00] K. Klein & V. Sequeira. *The view-cube : an efficient method of view planning for 3D modelling from range data*. In Proceedings Fifth IEEE Workshop on Applications of Computer Vision (WACV 2000), Palm Springs, CA, décembre 2000.
- [Kobbelt 99] L. Kobbelt, J. Vorsatz, U. Labsik & HP. Seidel. *A shrink wrapping approach to remeshing polygonal surfaces*. EUROGRAPHICS 99, vol. 18, no. 3, 1999.
- [Lachaud 98] J-O. Lachaud. *Extraction de surfaces à partir d images tridimensionnelles : approche discrète et approche par modèle déformable, PhD thesis*. Thèse de doctorat, Université Joseph Fourier, Grenoble, France, 1998.
- [Lagarde 04] J. M. Lagarde, M. Devy, L. Moreno, P. Lacroix & P. Lecoutaller. *Projet BODY SCAN : modélisation corporelle par profilométrie*. In RFIA'04, 2004.
- [Langis 01] C. Langis, M. A. Greenspan & G. Godin. *The Parallel Iterative Closest Point Algorithm*. In 3rd International Conference on 3-D Digital Imaging and Modeling (3DIM2001), Québec City, Canada, mai 2001.
- [Lazebnik 01] S. Lazebnik, E. Boyer & J. Ponce. *On Computing Exact Visual Hulls of Solids Bounded by Smooth Surfaces*. In Proceedings of the 2001 IEEE Conference on Computer Vision and Pattern Recognition, Kauai, Hawaii, pages 156–161, décembre 2001.
- [Lee 02] H. Lee, J. Lee & H. S. Yang. *Real-time LOD : Marching-cube-and- octree-based 3D Object Level-of-detail Modeling*. Rapport technique, VRRRC, septembre 2002.
- [Levoy 00] M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, J. Shade & D. Fulk. *The Digital Michelangelo Project : 3D Scanning of Large Statues*. In Kurt Akeley, editeur, Siggraph 2000, Computer Graphics Proceedings, pages 131–144. ACM Press / ACM SIGGRAPH / Addison Wesley Longman, 2000.
- [Liu 03] Y. Liu & W. Heidrich. *Interactive 3D Model Acquisition and Registration*. In 11th Pacific Conference on Computer Graphics and Applications, page 115. IEEE Computer Society, 2003.
- [Liu 04] Y. Liu. *Improving ICP with easy implementation for free-form surface matching*. Pattern Recognition, no. 37, pages 211–226, 2004.
- [Lorensen 87] W.E. Lorensen & H.E. Cline. *Marching cubes : A high resolution 3D surface construction algorithm*. Computer Graphics, pages 163–169, 1987.
- [Lozano 02] M. Lozano & M. Devy. *Perception planning for an exploration task of a 3D environment*. In International Conference on Pattern Recognition (ICPR'2002), Québec (Canada), Rapport LAAS N.02485, août 2002.

- 
- [Luck 00] J. Luck, C. Little, & W. Hoff. *Registration of Range Data Using a Hybrid Simulated Annealing and Iterative Closest Point Algorithm*. In IEEE International Conference on Robotics and Automation (ICRA2000), San Francisco, USA, avril 2000.
- [M. K. Reed 99] P. K. Allen M. K. Reed. *3-D Modeling from Range Imagery : An Incremental Method with a Planning Component*. Image and Vision Computing, vol. 17, page 99 111, 1999.
- [Maier 03] D. Maier, J. Hesser & R. Männer. *Fast and Accurate Closest Point Search on Triangulated Surfaces and its Application to Head Motion Estimation*. 3rd WSEAS International Conference on SIGNAL, SPEECH and IMAGE PROCESSING (WSEAS ICOSSIP 2003), 2003.
- [Mallet 00] A. Mallet, S. Lacroix & L. Gallo. *Position Estimation in Outdoor Environments using Pixel Tracking and Stereovision*. In IEEE International Conference on Robotics and Automation, San Francisco, Ca (USA), pages 3519–3524, April 2000.
- [Massios 98] N. A. Massios & R. B. Fisher. *A Best Next View Selection Algorithm incorporating a Quality Criterion*. In British Machine Vision Conference, pages 780–789, 1998.
- [Masuda 02] T. Masuda. *Registration and integration of multiple range images by matching signed distance fields for object shape modeling*. Computer Vision and Image Understanding (CVIU), vol. 87, no. 1/2/3, pages 51–65, 2002.
- [McCormick 02] N.H. McCormick & R.B. Fisher. *Edge-constrained marching Triangles*. In Proc. Int. Symp. on 3D Data Processing Visualization and Transmission (3DPVT), Padova, Italy, pages 348–351, juin 2002.
- [Medioni 91] G. Medioni & Y. Chen. *Object Modeling by Registration of Multiple Range Images*. In IEEE International Conference on Robotics and Automation (ICRA1991), Sacramento, USA, avril 1991.
- [Morency 02] L.-P. Morency & T. Darrell. *Stereo Tracking using ICP and Normal Flow Constraint*. In 16th International Conference on Pattern Recognition (ICPR 2002), Quebec City, Canada, 2002.
- [Morency 03] L.-P. Morency, & R. Gupta. *Robust Real-time Egomotion from Stereo Images*. In IEEE International Conference on Image Processing (ICIP 2003), Barcelona, Spain, septembre 2003.
- [Murino 01] V. Murino, L. Ronchetti, U. Castellani & A. Fusiello. *Reconstruction of complex environments by robust pre-aligned IC*. In 3rd International Conference on 3-D Digital Imaging and Modeling (3DIM2001), Québec City, Canada, mai 2001.
- [Neugebauer 97] P. J. Neugebauer. *Geometrical cloning of 3D objects via simultaneous registration of multiple range images*. In International Conference on Shape Modeling and Applications (SMA '97), page 130. IEEE Computer Society, 1997.
- [Nishino 02] K. Nishino & K. Ikeuchi. *Robust simultaneous registration of multiple range images*. In 5th Asian Conference on Computer Vision, pages 454–461, janvier 2002.
- [Nüchter 03] A. Nüchter, H. Surmann & J. Hertzberg. *Planning Robot Motion for 3D Digitalization of Indoor Environments*. In icar2003, 2003.
- [Ohta 98] N. Ohta & K. Kanatani. *Optimal estimation of three-dimensional rotation and reliability evaluation*. Lecture Notes in Computer Science, vol. 1406, pages 175–, 1998.
- [Okatani 02] I. S. Okatani & K. Deguchi. *A method for fine registration of multiple view range images considering the measurement error properties*. Computer Vision and Image Understanding (CVIU), vol. 87, no. 1/2/3, pages 66–77, 2002.
- [Penec 95] X. Penec & J.-P. Thirion. *Validation of 3D Registration Methods based on Points and Frames*. In 5th IEEE International Conference on Computer Vision (ICCV1995), Cambridge, USA, pages 557–562, juin 1995.

- [Penec 96] X. Penec. *Multiple Registration and Mean Rigid Shape - Application to the 3D case*. In Image Fusion and Shape Variability Techniques (16th Leeds Annual Statistical Workshop), pages 178–185, juillet 1996.
- [Penec 97a] X. Penec. *Registration of Uncertain Geometric Features : Estimating the Pose and its Accuracy*. In J. Le Moigne, editeur, First Image Registration Workshop, Greenbelt, Maryland, USA, pages 263–272, novembre 1997.
- [Penec 97b] X. Penec & J.-P. Thirion. *A Framework for Uncertainty and Validation of 3D Registration Methods based on Points and Frames*. International Journal of Computer Vision, vol. 3, no. 25, pages 203–229, 1997.
- [Pito 96] R. Pito. *A sensor based solution to the next best view problem*. In ICPR96, 1996.
- [Pritchett 98] P. Pritchett & A. Zisserman. *Matching and Reconstruction from Widely Separated Views*. In R. Koch & L. Van Gool, editeurs, 3D Structure from Multiple Images of Large-Scale Environments, LNCS 1506, pages 78–92. Springer-Verlag, juin 1998.
- [Pulli 97] K. Pulli. *Surface Reconstruction and Display from Range and Color Data*. Thèse de doctorat, University of Washington, 1997.
- [Pulli 98] K. Pulli, H. Abi-Rached, T. Duchamp, L. G. Shapiro & W. Stuetzle. *Acquisition and Visualization of Colored 3D Objects*. In 14th International Conference on Pattern Recognition (ICPR 1998), Brisbane, Australia, pages 11–15, août 1998.
- [Pulli 99] K. Pulli. *Multiview Registration for Large Data Sets*. In 2nd International Conference on 3-D Digital Imaging and Modeling (3DIM99), Ottawa, Canada, pages 160–168, octobre 1999.
- [Robertson 02] C. Robertson & R. B. Fisher. *Parallel Evolutionary Registration of Range Data*. In Computer Vision and Image Understanding (CVIU), volume 87, pages 39–50, 2002.
- [Rodrigues 02] M. Rodrigues, R. B. Fisher & Y. Liu. *Special Issue on Registration and Fusion of Range Images*. In Computer Vision and Image Understanding (CVIU), volume 87, pages 1–7, 2002.
- [Rodriquez 90] J. J. Rodriquez & J. K. Aggarwal. *Matching Aerial Images to 3-D Terrain Maps*. IEEE Transactions Pattern Analysis and Machine Intelligence, vol. 12, no. 12, pages 1138–1149, 1990.
- [Rosen 90] K. Rosen. Discrete mathematics and its applications. MacGraw-Hill, New York, second edition, 1990.
- [Ruggeri 01] M. Ruggeri, P. Dias, V. Sequeira & J.G.M. Goncalves. *Interactive tools for quality enhancement in 3D modelling from reality*. In 9th International Symposium on Intelligent Robotic Systems (SIRS'2000), Toulouse (France), Rapport LAAS N.01091, juillet 2001.
- [Rusinkiewicz 01] S. Rusinkiewicz & M. Levoy. *Efficient Variants of the ICP Algorithm*. In 3rd International Conference on 3-D Digital Imaging and Modeling (3DIM2001), Québec City, Canada, pages 145–152, 2001.
- [Sanchiz 99] J.M. Sanchiz & R.B. Fisher. *A next-best-view algorithm for 3D scene recovery with 5 degrees of freedom*. In BMVC99, 1999.
- [Sappa 01a] A. Sappa & M. Devy. *Fast Range Image Segmentation by an Edge Detection Strategy*. In Proc. Third International Conference on 3D Digital Imaging and Modeling, Quebec (Canada), mai 2001.
- [Sappa 01b] A. Sappa, M. Devy & A. Restrepo-Specht. *Segmentation d'images de profondeur par extraction des contours 3D*. In ORASIS 2001, Congrès francophone de vision, Cahors, pages 375–384, juin 2001.
- [Sappa 01c] A. Sappa, A. Restrepo-Specht & M. Devy. *Range Image Registration by using an Edge-Based Representation*. In 9th International Symposium on Intelligent Robotic Systems (SIRS'2001), Toulouse, France, juillet 2001.

- 
- [Sappa 02] A. Sappa, A. Restrepo-Specht & M. Devy. *Comparaison de méthodes de recalage de données 3D par construction d'un maillage triangulaire ou extraction des contours*. In 13ème Congrès Francophone AFRIF-AFIA de Reconnaissance des Formes et Intelligence Artificielle (RFIA'2002), Angers (France), janvier 2002.
- [Schoenenberger 00] Ch. Schoenenberger, P. Graebing & E. Hirsch. *recalage de séquences spatiales d'images en vue d'une évaluation dimensionnelle de surfaces libres*. *Traitement du Signal*, vol. 17, no. 4, pages 325–339, 2000.
- [Sequeira 96] V. Sequeira, J.G.M. Gonçalves & M.I. Ribeiro. *3D reconstruction of indoor environments*. In Proc. ICIP'96 - IEEE International Conference on Image Processing, pages 405–408, septembre 1996.
- [Sequeira 97] Vitor Sequeira. *Active Range Sensing for Three-Dimensional Environment Reconstruction*. Technical note n. i.97.26, JOINT RESEARCH CENTRE, Institute for Systems, Informatics and Safety, 21020 Ispra (VA) Italy, février 1997.
- [Sharp 99] G. C. Sharp, S. W. Lee & D. K. Wehe. *Invariant Features and the Registration of Rigid Bodies*. In IEEE International Conference on Robotics and Automation (ICRA1999), Detroit, USA, pages 932–937, 1999.
- [Sharp 01] G. C. Sharp, S. W. Lee & D. K. Wehe. *Toward Multiview Registration in Frame Space*. In IEEE International Conference on Robotics and Automation (ICRA2001), Seoul, Korea, pages 3542–3547, 2001.
- [Sharp 02a] G. C. Sharp, S. W. Lee & D. K. Wehe. *ICP Registration Using Invariant Features*. *IEEE Transactions Pattern Analysis and Machine Intelligence*, vol. 24, no. 1, pages 90–102, 2002.
- [Sharp 02b] G. C. Sharp, S. W. Lee & D. K. Wehe. *Multiview Registration of 3D Scenes by Minimizing Error Between Coordinate Frames*. In 7th European Conference on Computer Vision (ECCV 2002), Copenhagen, Denmark, pages 587–597, mai 2002.
- [Sharp 02c] G. C. Sharp, S. W. Lee & D. K. Wehe. *Registration of Range Images in the Presence of Occlusions and Missing Data*. Tech report cse-tr-453-02, University of Michigan, février 2002.
- [Sharp 02d] Gregory C. Sharp. *Automatic and Stable Multiview 3D Surface Registration, Ph.D. Thesis*. Thèse de doctorat, University of Michigan, 2002.
- [Shevlin 96] F. Shevlin. *Geometric rectification of scanned imagery*. Thèse de doctorat, University of Dublin, 1996.
- [Simon 96] D. Simon. *Fast and Accurate Shape-Based Registration*. Thèse de doctorat, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, décembre 1996.
- [Stoddart 96a] A. J. Stoddart & Brunnstrom K. *Free-Form Surface Matching using Mean Field Theory*. In 7th (BMVC 1996), Edinburgh, 1996.
- [Stoddart 96b] A J Stoddart, S Lemke, A Hilton, & T Renn. *Estimating Pose Uncertainty for Surface Registration*. In 7th (BMVC 1996), Edinburgh, 1996.
- [Sun 03] Y. Sun, J. Paik, A. Koschan, D. L. Page & M. A. Abidi. *Point Fingerprint : A New 3-D Object Representation Scheme*. *IEEE Transactions on Systems, Man, and Cybernetics-Part B : Cybernetics*, vol. 33, no. 4, pages 712–71, août 2003.
- [Tarabanis 96] K.A. Tarabanis, R.Y. Tsai & A. Kaul. *Computing occlusion-free view-points*. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 3, pages 279–292, mars 1996.
- [Thirion 92] J. P. Thirion & A. Gourdon. *The 3D Marching Lines Algorithm and Its Application to Crest Lines Extraction*. Research report RR-1672, INRIA, mai 1992.
- [Thirion 93a] J.-P. Thirion & S. Benayoun. *Image Surface Extremal Points, new feature points for Image Registration*. Research report 2003, INRIA, juillet 1993.
- [Thirion 93b] J.-P. Thirion & A. Gourdon. *The 3D Marching Lines Algorithm : new results and proofs : part 1*. Research report 1881-1, INRIA, avril 1993.



- [Thirion 93c] J.-P. Thirion & A. Gourdon. *The 3D Marching Lines Algorithm : new results and proofs : part 2*. Research report 1881-2, INRIA, avril 1993.
- [Thirion 95] J.-P. Thirion. *Fast Non-Rigid Matching of 3D Medical Images*. In mrcas95, Baltimore, novembre 1995.
- [Thirion 96] J. P. Thirion. *New Feature Points based on Geometric Invariants for 3D Image Registration*. International Journal of Computer Vision, vol. 18, no. 2, pages 121–137, mai 1996.
- [Torr 98] P. H. S. Torr & A. Zisserman. *Robust Computation and Parameterization of Multiple View Relations*. In 6th IEEE International Conference on Computer Vision (ICCV1998), Bombay, India, pages 727–732, janvier 1998.
- [Tovar 01] B. Tovar, R. Murrieta, C. Esteves. & D. Pérez. *Robot motion planning for Model Building under perception constraints*. In In proc International Symposium on Intelligent Robotic Systems, SIRS2001, 2001.
- [Tubic 03] D. Tubic, P. Hébert & D. Laurendeau. *A volumetric approach for interactive 3D modeling*. Computer Vision and Image Understanding (CVIU), vol. 92, no. 1, pages 56–77, 2003.
- [Turk 94] G. Turk & M. Levoy. *Zippered Polygon Meshes from Range Images*. In SIGGRAPH'94, pages 311–318, 1994.
- [Walker 91] M. Walker & L. Shao. *Estimating 3-d location parameters using dual number quaternions*. CVGIP : Image Understanding, vol. 54, no. 3, pages 358–367, novembre 1991.
- [Weik 97] S. Weik. *Registration of 3-D Partial Surface Models Using Luminance and Depth Information*. In International Conference on 3-D Digital Imaging and Modeling (3DIM97), Ottawa, Canada, pages 93–, mai 1997.
- [Werner 02] T. Werner & A. Zisserman. *Model Selection for Automated Architectural Reconstruction from Multiple Views*. In 13th British Machine Vision Conference (BMVC 2002), Cardiff, pages 53–62, 2002.
- [Wheeler 98] M. D. Wheeler, Y. Sato & K. Ikeuchi. *Consensus Surfaces for Modeling 3D Objects from Multiple Range Images*. In ICCV Sixth International Conference on Computer Vision, pages 917–924, 1998.
- [Williams 00] J. A. Williams & M. Bennamoun. *Multiple View Surface Registration with Error Modeling and Analysis*. In IEEE International Conference on Image Processing (ICIP 2000), Vancouver, BC, Canada, volume I, pages 545–548, septembre 2000.
- [Wong 99] L. M. Wong, C. Dumont & M. A. Abidi. *Next Best View System in a 3-D Object Modeling Task*. In Proceedings of IEEE International Symposium on Computational Intelligence in Robotics and Automation, pages 306–311, novembre 1999.
- [Wyngaerd 99] J. V. Wyngaerd, R. Koch & M. Proesmans. *Invariant-Based Registration of Surface Patches*. In 7th IEEE International Conference on Computer Vision (ICCV1999), Corfu, Greece, pages 301–, septembre 1999.
- [Wyngaerd 02] J. V. Wyngaerd & L. Van Gool. *Automatic crude patch registration : toward automatic 3D model building*. Computer Vision and Image Understanding (CVIU), vol. 87, no. 1/2/3, pages 8–26, 2002.
- [Xuan 97] J. Xuan, Y. Wang, T. Adali & Q. Zheng. *A Deformable Surface-Spine Model for 3-D Surface Registration*. In IEEE International Conference on Image Processing (ICIP '97), Washington, D.C., USA, pages 236–. IEEE Computer Society, 1997.
- [Yamany 98] S. M. Yamany, M. Ahmed, E. Hemayed & A. A. Farag. *Novel Surface Registration using the Grid Closest Point (GCP) Transform*. In IEEE International Conference on Image Processing (ICIP '98), Chicago, USA, pages 809–813, octobre 1998.

- 
- [Yamany 99] S. M. Yamany & A. A. Farag. *Free-Form Surface Registration Using Surface Signatures*. In 7th IEEE International Conference on Computer Vision (ICCV1999), Corfu, Greece, pages 1098–. IEEE Computer Society, 1999.
- [Yang 98] R. Yang & P. K. Allen. *Registering, Integrating, and Building CAD Models from Range Data*. In IEEE International Conference on Robotics and Automation (ICRA1998), Leuven, Belgium, pages 3115–3120, mai 1998.
- [Zhang 94] Z. Zhang. *Iterative Point Matching for Registration of Free-Form Curves and Surfaces*. International Journal of Computer Vision, vol. 13, no. 2, pages 119–149, février 1994.
- [Zhang 99] D. Zhang. *Harmonic Shape Images : A 3D Free-form Surface Representation and Its Applications in Surface Matching*. Thèse de doctorat, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, novembre 1999.
- [Zinßer 03] T. Zinßer, J. Schmidt & H. Niemann. *A Refined ICP Algorithm for Robust 3-D Correspondence Estimation*. In IEEE International Conference on Image Processing (ICIP 2003), Barcelona, Spain, septembre 2003.