



## Chaine 3D interactive

Vincent Daval

### ► To cite this version:

Vincent Daval. Chaine 3D interactive. Vision par ordinateur et reconnaissance de formes [cs.CV]. Université de Bourgogne, 2014. Français. NNT: . tel-01164114v1

**HAL Id: tel-01164114**

**<https://hal.science/tel-01164114v1>**

Submitted on 16 Jun 2015 (v1), last revised 15 Sep 2015 (v2)

**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.

# SPIM

## Thèse de Doctorat



école doctorale **sciences pour l'ingénieur et microtechniques**  
U N I V E R S I T É   D E   B O U R G O G N E

## Chaîne 3D interactive

■ VINCENT DAVAL





# SPIM

## Thèse de Doctorat



école doctorale sciences pour l'ingénieur et microtechniques  
UNIVERSITÉ DE BOURGOGNE

N° X X X

THÈSE présentée par

VINCENT DAVAL

pour obtenir le

Grade de Docteur de  
l'Université de Bourgogne

Spécialité : **Informatique et Instrumentation de l'Image**

Chaine 3D interactive

Soutenue publiquement le 09 décembre 2014 devant le Jury composé de :

MARC ANTONINI	Rapporteur	Directeur de recherche - I3S - Université de Sophia Antipolis
OLIVIER AUBRETON	Directeur de thèse	Maitre de conférence HDR - LE2I - Université de Bourgogne
GILLES GESQUIÈRE	Rapporteur	Professeur - LIRIS - Université Lumière de Lyon
FABRICE MÉRIAUDEAU	Examineur	Professeur - LE2I - Université de Bourgogne
FRÉDÉRIC PAYAN	Examineur	Maitre de conférence - I3S - Université de Sophia Antipolis
FRÉDÉRIC TRUCHETET	Co-directeur	Professeur - LE2I - Université de Bourgogne





# REMERCIEMENTS

Avant toute chose, je tiens à exprimer mes plus sincères remerciements aux différents acteurs qui ont contribué, de près ou de loin, à ce projet et qui m'ont permis de le mener à bien. Bien entendu, je remercie avant tout la direction du laboratoire Le2i (UMR CNRS 6306) de m'avoir accueilli au sein du département Vision du Creusot, ainsi qu'au Conseil Régional de Bourgogne avec l'aide du Fond Européen de Développement Economique Régional (FEDER), financeur de ces travaux.

Je tiens tout d'abord à remercier mon directeur de thèse, Olivier AUBRETON, ainsi que mon co-directeur, Frédéric TRUCHETET, pour la confiance qu'ils m'ont accordée pour la réalisation de cette thèse durant ces trois années. La complémentarité de leur encadrements, leur disponibilité, leur précieux conseils et le temps qu'ils m'ont accordé durant cette période m'ont permis de mener ce projet à terme.

J'exprime également mes plus sincères remerciements à Marc ANTONINI et Gilles GESQUIERES de m'avoir fait l'honneur d'accepter de rapporter mon manuscrit et d'avoir consacré de leur temps à l'examen de ce mémoire. De même, je souhaite remercier Frédéric PAYAN et Fabrice MERIAUDEAU d'avoir accepté d'être examinateurs de cette thèse.

Bien qu'une thèse soit un travail personnel, je n'aurais jamais pu aboutir à ces résultats sans le soutien et l'ambiance de travail qui règne au sein du laboratoire. De ce fait, je tiens à remercier chaleureusement l'ensemble des doctorants, permanents, stagiaires, et bien sur à Nathalie, secrétaire du laboratoire, qui véhiculent une très bonne ambiance au sein de l'équipe.

Enfin, je tiens à exprimer ma plus sincère gratitude à ma famille pour m'avoir permis de faire ce que je souhaitais et de m'avoir soutenu tout au long de ces années.



# SOMMAIRE

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	La numérisation 3D . . . . .	1
1.2	Problématique . . . . .	2
1.3	Contexte de la thèse . . . . .	4
1.4	Contributions . . . . .	4
<b>2</b>	<b>État de l'art</b>	<b>7</b>
2.1	La numérisation 3D . . . . .	7
2.1.1	Systèmes d'acquisition optique passifs . . . . .	9
2.1.1.1	Stéréovision . . . . .	9
2.1.1.2	Autres systèmes . . . . .	10
2.1.2	Systèmes optique actifs . . . . .	11
2.1.2.1	Lumière structurée . . . . .	11
2.1.2.2	Autres systèmes . . . . .	12
2.1.3	Synthèse . . . . .	12
2.2	Fonctionnement d'un scanner . . . . .	14
2.2.1	Le modèle du sténopé . . . . .	14
2.2.1.1	Du repère monde au repère caméra . . . . .	14
2.2.1.2	Du repère caméra au repère capteur . . . . .	15
2.2.1.3	Du repère capteur au repère image . . . . .	16
2.2.1.4	Modèle complet . . . . .	16
2.2.1.5	Les distorsions . . . . .	18
2.2.1.6	Calibrage du système . . . . .	20
2.2.2	Mise en correspondance . . . . .	20

2.2.2.1	Rectification des images . . . . .	20
2.2.2.2	Méthodes locales classiques . . . . .	22
2.2.2.3	Méthodes locales par propagation de germes . . . . .	23
2.2.2.4	Méthodes globales . . . . .	24
2.2.2.5	Texture aléatoire . . . . .	24
2.2.2.6	Synthèse . . . . .	25
2.2.3	Triangulation des points . . . . .	25
2.3	Extraction d'informations complémentaires . . . . .	27
2.3.1	Normale à la surface . . . . .	27
2.3.1.1	A partir du nuage de points . . . . .	27
2.3.1.2	A partir d'un maillage . . . . .	29
2.3.1.3	A partir des images . . . . .	30
2.3.2	Courbure . . . . .	30
2.3.2.1	Définition de la courbure . . . . .	30
2.3.2.2	Méthodes de calcul . . . . .	31
2.3.3	Segmentation 3D . . . . .	32
2.3.3.1	Croissance de Région . . . . .	33
2.3.3.2	Ligne de partage des eaux . . . . .	33
2.3.3.3	Partitionnement de graphes . . . . .	34
2.3.3.4	Sphère gaussienne . . . . .	34
2.3.4	Synthèse . . . . .	34
2.4	Simplification des données . . . . .	35
2.4.1	Simplification d'un nuage de points . . . . .	35
2.4.1.1	Méthodes de clustering . . . . .	35
2.4.1.2	Méthodes coarse-to-fine . . . . .	36
2.4.1.3	Méthodes itératives . . . . .	36
2.4.2	Simplification d'un maillage . . . . .	37
2.4.2.1	Regroupement de sommets . . . . .	37
2.4.2.2	Technique de décimation . . . . .	37

2.4.2.3	Surfaces enveloppantes . . . . .	39
2.4.2.4	Maillage progressif . . . . .	39
2.4.2.5	Autres méthodes . . . . .	39
2.4.3	Primitives 3D . . . . .	40
2.4.3.1	Ransac . . . . .	41
2.4.3.2	Transformée de Hough . . . . .	42
2.4.3.3	Méthode de Benkő <i>et al.</i> . . . . .	43
2.4.3.4	Méthode de Sunil <i>et al.</i> . . . . .	44
2.4.3.5	Méthode de Bénéière . . . . .	45
2.4.3.6	Synthèse . . . . .	45
<b>3</b>	<b>Extraction d'informations depuis les images</b>	<b>47</b>
3.1	Calcul des normales . . . . .	49
3.1.1	Adaptation de la méthode de Song . . . . .	49
3.1.2	Influence de la densité de points . . . . .	52
3.1.3	Caractérisation du bruit sur les normales . . . . .	54
3.2	Détection d'arêtes saillantes par imagerie . . . . .	56
3.2.1	Estimation du rayon de courbure . . . . .	56
3.2.2	Discontinuités . . . . .	59
3.2.3	Principe . . . . .	59
3.2.4	Résultats . . . . .	61
3.3	Primitives . . . . .	62
3.3.1	La sphère gaussienne . . . . .	62
3.3.2	Analyse en composantes principales . . . . .	63
3.3.2.1	Identification des primitives . . . . .	64
3.3.2.2	Modélisation des primitives . . . . .	66
3.3.2.3	Vérification . . . . .	70
3.3.2.4	Résultats . . . . .	70
3.4	Conclusion . . . . .	72

<b>4</b>	<b>Système de numérisation dynamique</b>	<b>73</b>
4.1	Fonctionnement du système . . . . .	73
4.2	Segmentation . . . . .	76
4.2.1	Méthodes globales . . . . .	78
4.2.2	Méthodes mixtes . . . . .	79
4.2.2.1	Split and Merge . . . . .	79
4.2.2.2	Optimisation . . . . .	80
4.2.3	Méthodes locales . . . . .	80
4.2.4	Approche utilisée . . . . .	80
4.3	Extraction des primitives . . . . .	82
4.3.1	Ajustement des primitives . . . . .	83
4.3.2	Affinage de la segmentation . . . . .	84
4.3.3	Synthèse . . . . .	86
4.4	Surfaces paramétriques . . . . .	86
4.4.1	Surfaces de Bézier . . . . .	87
4.4.1.1	Courbes de Bézier . . . . .	87
4.4.1.2	Estimation des points de contrôle . . . . .	88
4.4.1.3	Décomposition des objets . . . . .	92
4.4.2	Test des surfaces . . . . .	93
4.5	Incrémentation et arrêt du système . . . . .	93
4.6	Conclusion . . . . .	94
<b>5</b>	<b>Résultats</b>	<b>95</b>
5.1	Outil logiciel . . . . .	95
5.2	Résultats étape par étape . . . . .	96
5.2.1	Acquisition . . . . .	97
5.2.2	Marquage des régions . . . . .	99
5.2.2.1	Cas des primitives . . . . .	101
5.2.2.2	Cas des surfaces paramétriques . . . . .	102

5.2.3	Incrémentation du système . . . . .	103
5.2.4	Résultats . . . . .	104
5.3	Impact des différents paramètres . . . . .	107
5.4	Évaluation de notre méthode . . . . .	110
5.5	Amélioration du système . . . . .	112
5.5.1	Principe de la méthode . . . . .	113
5.5.2	Extraction des paramètres des ellipses . . . . .	114
5.5.3	Reconstruction de la forme 3D . . . . .	115
<b>6</b>	<b>Conclusion et perspectives</b>	<b>117</b>
6.1	Résumé de nos contributions . . . . .	117
6.2	Perspectives . . . . .	119
6.3	Publications . . . . .	121
	<b>Bibliographie</b>	<b>123</b>
	<b>Annexe A</b>	<b>133</b>
	<b>Annexe B</b>	<b>145</b>
	<b>Table des figures</b>	<b>151</b>
	<b>Liste des tables</b>	<b>157</b>





# INTRODUCTION

## 1.1/ LA NUMÉRISATION 3D

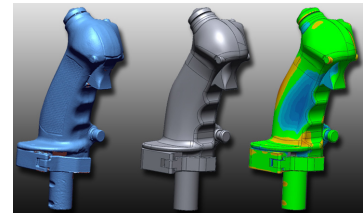
La numérisation 3D est un domaine en constante évolution qui intéresse de plus en plus le monde de la recherche et le grand public. Le concept de base consiste à copier numériquement un objet ou une scène en trois dimensions à l'aide d'un scanner 3D. Si la numérisation 3D est devenue un domaine aussi répandu, c'est en grande partie dû à la forte croissance du nombre et du type d'applications faisant appel à cette technique (figure 1.1). En effet, aujourd'hui la numérisation 3D est devenue un outil très utilisé dans de nombreux domaines comme la médecine (simulation du résultat d'une opération, suivi médical, prothèses), l'archéologie (reconstitution d'objets abîmés, duplication de pièces anciennes), le patrimoine (déformation des structures, évolution des travaux, archivage des bâtiments), ou encore des applications industrielles (contrôle qualité, rétro-conception). Chaque application a des exigences particulières, la rétro-conception demande une grande précision là où la chirurgie nécessite un système portable pour plus de souplesse. Le patrimoine, quant à lui, nécessite un système adapté aux objets de grande dimension afin de numériser l'objet ou la scène beaucoup plus rapidement.



(a) Patrimoine



(b) Médical



(c) Contrôle qualité

FIGURE 1.1 – *Diverses applications faisant appel à la numérisation 3D.*

De ce fait, il existe de nombreuses technologies différentes permettant de numériser un objet en 3D et donc, autant de type de scanners (figure 1.2). Parmi ces différentes

technologies, il y a entre autres les systèmes à temps de vol, les scanners à décalage de phase ou encore les systèmes à triangulation. Chaque technologie a ses avantages et ses inconvénients, souvent liés à son domaine d'application. Les systèmes à temps de vol, par exemple, sont moins précis que les autres technologies de scanners mais ils permettent de reconstruire des objets de grande taille en un temps record, alors que les scanners à triangulation fournissent des données beaucoup plus denses et précises mais leur champ de numérisation est de l'ordre du centimètre. Dans les cas industriels, les scanners les plus répandus et les plus adaptés aux différentes applications sont donc les scanners à triangulation.



(a) Comet



(b) Minolta



(c) Kinect



(d) Artec

FIGURE 1.2 – *Différents types de scanner 3D.*

## 1.2/ PROBLÉMATIQUE

Avec l'évolution constante des technologies, les scanners 3D fournissent de plus en plus de données avec une précision toujours plus grande. Cependant, l'augmentation considérable de la taille des données pose des problèmes, les fichiers deviennent très lourds et il peut en découler des difficultés de transmission ou de stockage. La plupart du temps, les données obtenues par les scanners sont ensuite analysées puis traitées. Inévitablement, en augmentant la taille des données, le temps d'analyse sera également augmenté.

Actuellement, la chaîne d'acquisition classique (figure 1.3) peut se décomposer en plusieurs étapes séquentielles faisant chacune appel à des compétences particulières bien distinctes, ce qui implique que chaque étape sera généralement traitée par différents opérateurs. Il y a donc très peu d'interactions entre les différentes étapes de la chaîne et chaque partie va être analysée séparément, indépendamment de son utilisation future. De ce fait, aujourd'hui, les scanners commerciaux fournissent des nuages très denses composés de plusieurs millions de points procurant ainsi la meilleure précision possible lors de l'acquisition, et ce quelle qu'en soit l'utilisation finale.

Si dans certains cas industriels cette quantité d'information semble justifiée (rétro-conception), dans bien d'autres cas (contrôle des défauts, mesure de distance, ...) calculer autant de points n'est pas forcément nécessaire. Bien au contraire, le fichier obtenu, très volumineux, posera des problèmes au niveau du stockage et du transfert des données. Le temps de calcul nécessaire pour analyser les données va lui aussi être considérablement augmenté par rapport à ce qu'il pourrait être. C'est pourquoi, avant d'être traitées, les données vont être simplifiées dans de nombreuses applications industrielles, pour réduire le nombre de points par des étapes de compression/simplification tout en préservant le maximum d'informations.

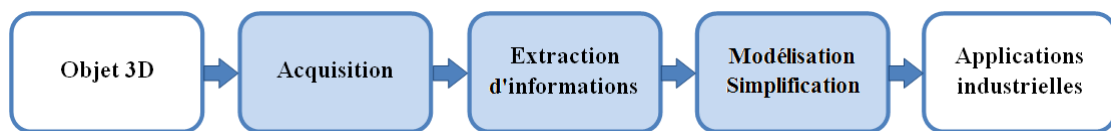


FIGURE 1.3 – *La chaîne 3D classique : de l'acquisition à la compression.*

La figure 1.3 illustre le schéma classique que l'on rencontre souvent. Dans un premier temps, l'opérateur va acquérir un nuage de points le plus dense possible à l'aide d'un scanner. L'étape suivante consiste généralement à calculer des informations complémentaires aux points 3D, comme les normales à la surface en chaque point, la courbure ou encore l'extraction de primitives géométriques simples (plans, cylindres, ...). Cette étape est généralement appliquée sur le nuage de points maillé, bien que l'on trouve aujourd'hui de plus en plus d'approches permettant d'extraire ces informations directement sur le nuage de points brut non maillé. Enfin, en fonction de l'utilisation que l'on souhaite faire des données, la dernière étape consiste à compresser/simplifier le nuage de points en fonction des différentes informations extraites lors de l'étape précédente. La partie concernant l'extraction des informations et la partie concernant la simplification sont aujourd'hui assez liées étant donné que la plupart des algorithmes de simplification de données tridimensionnelles nécessitent de calculer ces informations au préalable.

Ce schéma classique permet ainsi d'obtenir des données nettement moins volumineuses et donc plus rapides à traiter et plus simples à utiliser. Cependant cette approche amène une question : connaissant l'utilisation finale des données, pourquoi calculer autant de points lors de l'acquisition s'il faut en réduire le nombre par la suite ? N'est-il pas possible d'agir directement sur l'acquisition pour adapter les données à leur utilisation finale ? Nous proposons dans nos travaux une méthodologie permettant de répondre positivement à cette dernière interrogation.

### 1.3/ CONTEXTE DE LA THÈSE

Avant de présenter nos contributions, il convient de situer le contexte dans lequel s'inscrit cette thèse. Les travaux réalisés durant cette thèse s'inscrivent dans un projet, le projet CreActive 3D, regroupant trois partenaires : le laboratoire I3S [1], le laboratoire le2i [2] et la société Noomeo [3]. Le projet se divise en deux parties distinctes. La première partie consiste à guider l'acquisition à partir d'une analyse du maillage et la seconde partie, traitée dans ce document, consiste à extraire des informations depuis les images du scanner en vue de simplifier les données. La société Noomeo développe et produit des scanners commerciaux, principalement destinés au domaine de l'aéronautique. De ce fait, l'objectif final de ce projet consiste à appliquer nos travaux à l'un des scanners de l'entreprise : l'OptiNum (scanner portable). Le domaine principal étant l'aéronautique, les travaux que nous présentons dans la suite du document sont exclusivement destinés à des pièces manufacturées.

### 1.4/ CONTRIBUTIONS

Ce mémoire présente une approche qui permet de numériser les objets de manière dynamique, en adaptant la densité de points en fonction de la complexité de la forme de l'objet à numériser, et ce sans *a priori* sur la forme de l'objet. L'intérêt de cette approche est d'éviter de passer par le schéma classique, en adaptant la densité de points au niveau de l'acquisition afin d'obtenir des données simplifiées directement à la sortie de l'acquisition, plutôt qu'en calculant un nuage de points dense, qu'il faudra simplifier par la suite. Etant donné le contexte de la thèse, le système de numérisation présenté par la suite a été développé pour fonctionner sur des objets manufacturés. C'est pourquoi, ce mémoire sera présenté de la façon décrite dans le paragraphe suivant, à l'image de la chaîne 3D classique.

Dans un premier chapitre, il sera présenté un état de l'art de la chaîne 3D classique, étape par étape, en allant de l'acquisition à la compression. Dans cette partie, nous présenterons tout d'abord les différentes technologies de scanners qui existent aujourd'hui, ainsi que le fonctionnement du scanner mis à notre disposition par la société Noomeo. Étant donné que le système dynamique présenté dans ce mémoire utilise des données du scanner, de nombreux points présentés dans cette partie seront ré-utilisés par la suite dans d'autres chapitres. La seconde étape de la chaîne 3D traite de l'extraction d'informations complémentaires aux points 3D. Ainsi, dans un second temps, ce chapitre présentera les différents attributs généralement extraits durant cette étape, aussi bien sur le nuage de points que sur le maillage. Viendra enfin la dernière partie concernant la simplification des données et les différentes méthodes qui permettent de compresser/simplifier des données tridimensionnelles.

L'objectif de ce mémoire est de présenter un système permettant de simplifier les données durant l'acquisition. Le système développé s'inspire des différentes étapes de la chaîne 3D. La différence principale, avec les systèmes classiques, est que les données traitées sont les images fournies par le scanner plutôt que le nuage de points déterminés à partir de ces images. Ainsi, dans un second chapitre, nous présenterons les méthodes développées pour extraire des informations complémentaires à partir des images fournies par le scanner. Ces informations sont les normales 3D, les discontinuités, ainsi que des primitives géométriques.

Le troisième chapitre de ce mémoire présente comment l'objectif de notre travail, qui consiste à obtenir des données simplifiées à la sortie de l'acquisition en utilisant les données extraites lors de l'étape précédente (normales, discontinuités, primitives), est atteint. Le principe de fonctionnement complet de notre système sera présenté dans un quatrième chapitre, cette présentation s'appuyant sur ce qui se fait traditionnellement en 3D dans la chaîne classique.

Les résultats obtenus avec notre système seront présentés et comparés à des méthodes classiques (passant par toutes les étapes de la chaîne 3D) dans un cinquième chapitre. Enfin nous concluons et présenterons des perspectives envisageables pour améliorer le système.



## ÉTAT DE L'ART

Comme présenté en introduction, la chaîne 3D peut se décomposer en trois parties distinctes : l'acquisition 3D, l'extraction d'informations sur le nuage de points et la simplification/compression des données. Dans ce chapitre sont présentées une étude bibliographique et un état de l'art de chaque étape de la chaîne 3D : la numérisation 3D (introduction à la chaîne 3D), le fonctionnement d'un scanner (acquisition), l'extraction d'informations et la simplification des données.

### 2.1/ LA NUMÉRISATION 3D

Au cours des dernières années, le nombre de systèmes d'acquisition 3D commerciaux n'a cessé de progresser et leur implémentation ne cesse de s'étendre à de nouveaux domaines. Dès 1988, Paul Besl [4] présentait déjà un état de l'art sur les différentes méthodes d'acquisition 3D actives et optiques et comparait les différents systèmes existants à l'époque. Par la suite, Brian Curless [5] décrit les principes et les technologies des différents systèmes de numérisation et présente une arborescence permettant de classer et d'ordonner les différentes techniques. En s'inspirant de ces travaux, nous proposons une arborescence permettant de classer de manière non exhaustive les différentes techniques de numérisation nous concernant (figure 2.1).

Les différentes méthodes peuvent être classées en deux catégories, avec et sans contact. Actuellement, les méthodes ayant la plus grande précision sont celles avec contact. Le principe des méthodes avec contact consiste à palper la surface à l'aide d'une bille en rubis montée sur un stylet. Lors du contact entre la surface de l'objet et du palpeur, les coordonnées de la machine sont mémorisées. Ces systèmes sont généralement embarqués sur une Machine à Mesurer Tridimensionnelle (MMT). Cette technique fournit des résultats avec une précision de l'ordre du micromètre. De ce fait,



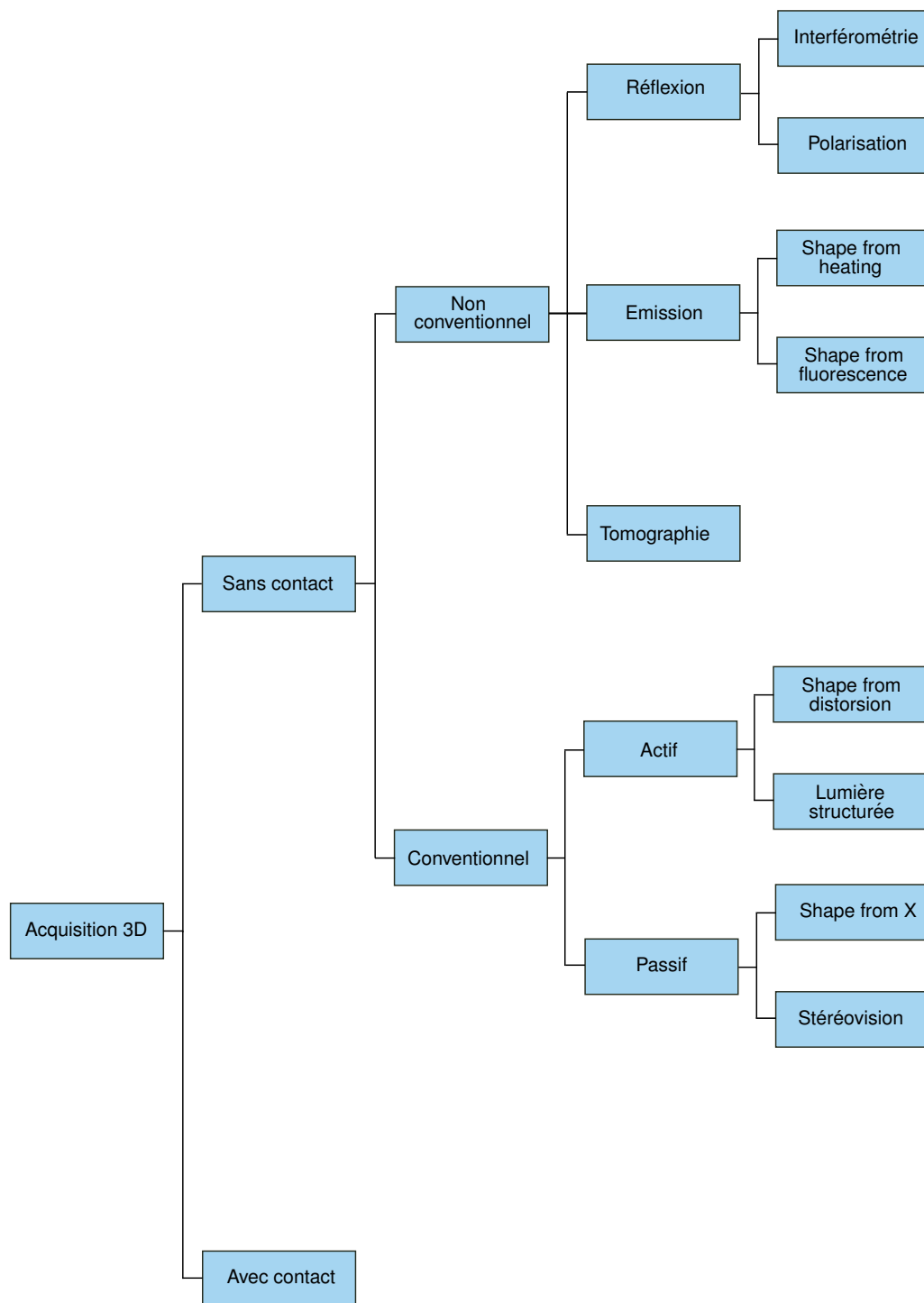


FIGURE 2.1 – Graphe non exhaustif des systèmes d'acquisition 3D.

les palpeurs sont généralement utilisés dans l'industrie pour obtenir une surface de référence. Cependant, l'inconvénient majeur d'une telle technique réside dans le temps d'acquisition (quelques points par seconde) qui peut très rapidement devenir très long en fonction de la taille de la pièce à numériser et de la résolution spatiale désirée.

Les systèmes de numérisation sans contact, quant à eux, peuvent être classés en deux catégories : les systèmes conventionnels et non conventionnels. Les approches conventionnelles peuvent être définies comme les méthodes s'approchant du système de vision humain. Parmi les méthodes non conventionnelles on retrouve 3 classes : les méthodes par réflexion (interférométrie [6, 7, 8, 9], polarisation [10, 11, 12]), par émission (« *scanning from heating* » [13], « *shape from fluorescence* » [14]) et enfin la tomographie.

Dans ce chapitre nous présentons un état de l'art des différents systèmes nous concernant, à savoir les méthodes de numérisation conventionnelles (figure 2.1), qui peuvent être décomposées en deux branches : les méthodes passives et les méthodes actives. Les techniques passives utilisent un ou plusieurs capteurs et ne nécessitent aucune source de lumière commandée pour obtenir des informations tridimensionnelles, à l'inverse des méthodes actives qui se servent de cette source pour obtenir des données 3D.

### 2.1.1/ SYSTÈMES D'ACQUISITION OPTIQUE PASSIFS

Comme nous venons de le mentionner, les méthodes passives ne nécessitent pas d'éclairage additionnel pour calculer des informations 3D. Ces systèmes peuvent faire appel à un ou plusieurs capteurs (caméra, appareil photographique) et utilisent diverses caractéristiques pour obtenir des données tridimensionnelles comme les mouvements, les différents points de vue, la netteté de l'image, etc.

#### 2.1.1.1/ STÉRÉOVISION

La stéréovision est la méthode la plus répandue et la plus étudiée parmi les approches passives. Il s'agit d'une méthode de reconstruction de profondeur calculée par triangulation à partir de deux images de la même scène prise à partir d'un point de vue différent. Pour que le calcul des informations 3D soit possible, les caméras doivent au préalable être calibrées, c'est à dire que les paramètres intrinsèques (focale, position du centre optique, ...) et les paramètres extrinsèques (position et orientation) doivent être connus.

La figure 2.2 illustre le principe de la stéréovision. Pour tout point  $P$  visible depuis les deux caméras, il existe une projection de celui-ci dans les plans images  $\Pi_1$  et  $\Pi_2$ , notés respectivement  $P_1$  et  $P_2$ . Inversement, connaissant les points  $P_1$  et  $P_2$ , ainsi que les centres optiques des caméras  $O_1$  et  $O_2$  (estimés lors de l'étape de calibrage), il est possible de remonter à la position du point  $P$  qui se trouve à l'intersection des droites  $(O_1P_1)$  et

$(O_2P_2)$ . De ce fait, seules les surfaces visibles communes aux deux caméras peuvent être reconstruites avec ce procédé.

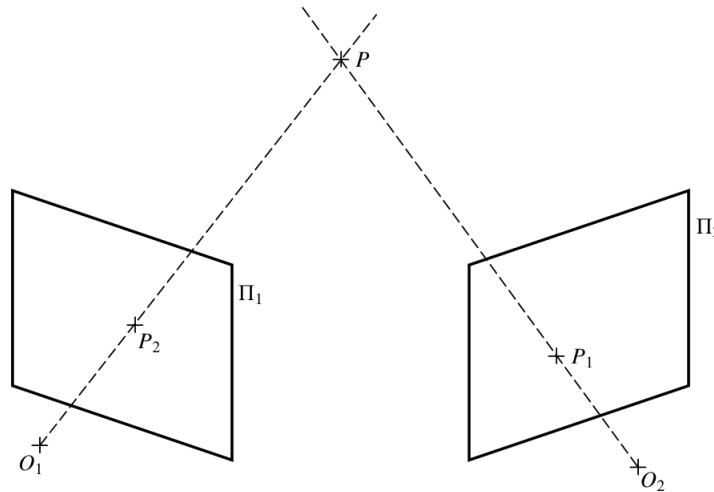


FIGURE 2.2 – *Principe de la stéréovision.*

La principale difficulté de cette méthode réside dans la mise en correspondance des points entre les différentes vues. En fonction de la scène à numériser, il est possible d'appliquer différentes techniques de mise en correspondance (texture, contrainte épipolaire, ...). Les travaux de thèse de Sylvie Chambon [15] présentent un bon état de l'art des différentes approches permettant de faire de la mise en correspondance entre deux vues.

Pour pallier ces contraintes, il est également possible de projeter une texture pseudo-aléatoire sur l'objet à numériser [16] ce qui facilite grandement la mise en correspondance entre les différentes vues. La lumière structurée ajoutée joue un rôle important dans la reconstruction en facilitant l'appariement, mais elle ne participe pas directement à la mesure de l'information tridimensionnelle. C'est la raison pour laquelle cette approche reste classée comme une méthode passive malgré l'utilisation d'une source de lumière.

### 2.1.1.2/ AUTRES SYSTÈMES

Il existe bien d'autres systèmes de numérisation passifs comme le « *shape from texture* » [17], le « *shape from motion* » [18], ou encore le « *shape from focus* » [19]. Cependant, dans le cadre de notre projet le système utilisé est un système de stéréovision à projection de lumière structurée. C'est pourquoi, nous avons choisi de nous limiter à ce type de scanners dans cette partie.

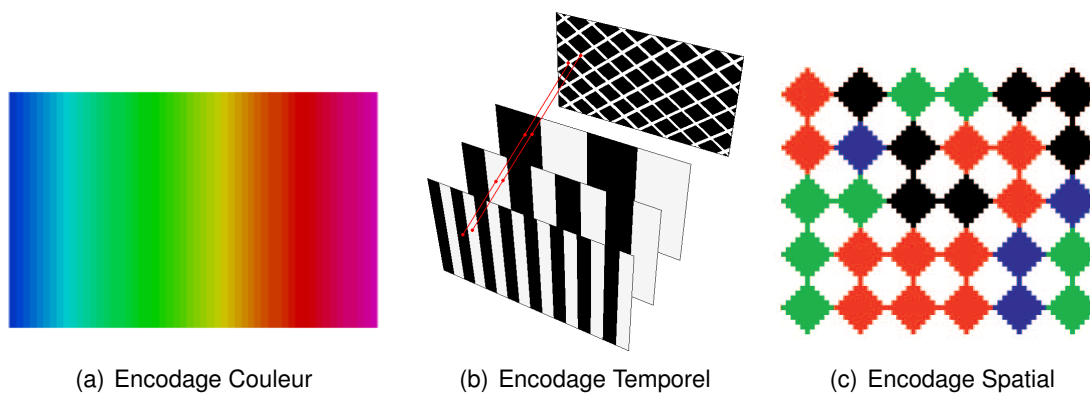
### 2.1.2/ SYSTÈMES OPTIQUE ACTIFS

Les systèmes passifs sont fortement dépendant des conditions d'éclairage et des textures de l'objet. Malheureusement, dans bien des cas, ces éléments ne peuvent pas être maîtrisés et les systèmes d'acquisition 3D deviennent alors moins performants voire totalement inopérants. Pour remédier à ces problèmes, il est possible d'utiliser des systèmes actifs : l'éclairage est maîtrisé et devient un élément important participant directement à la mesure tridimensionnelle. Il s'agit des méthodes les plus utilisées par les prestataires de services de numérisation 3D et par les fournisseurs de scanners commerciaux.

#### 2.1.2.1/ LUMIÈRE STRUCTURÉE

Les scanners à lumière structurée projettent un motif lumineux connu sur l'objet à numériser. Ce motif va alors se déformer en fonction de la surface de l'objet. Le principe d'un tel système consiste à analyser la déformation du motif projeté pour extraire la profondeur de la scène. Ces systèmes sont composés d'un ou plusieurs capteurs (caméras) et d'un projecteur, ils se rapprochent finalement d'un système de stéréovision classique ; la différence majeure se situe dans l'appariement des points entre les différentes vues. Dans le cas de la stéréovision, c'est la texture et les informations présentes sur les images qui sont utilisées alors que dans le cas de systèmes à lumière structurée, l'élément permettant la mise en correspondance entre différentes vues n'est autre que le motif projeté. Ces systèmes ont fait l'objet de très nombreuses recherches lors de ces dernières années tant les motifs et leurs méthodes d'analyse peuvent être variés. Salvi *et al.* [20] proposent un état de l'art des différentes méthodes ainsi qu'une classification de celles-ci en trois catégories distinctes : les motifs à encodage temporel [21, 22, 23], à encodage spatial [24, 25] et à encodage couleur [26, 27, 28] (figure 2.3). Chaque type d'encodage a ses propres avantages et inconvénients.

Les motifs couleurs utilisent un panel de couleurs pour encoder les points permettant ainsi une mise en correspondance entre les différentes vues assez simple. Ce type d'encodage permet de n'utiliser qu'une seule projection mais il est très sensible à la texture de l'objet à numériser. A l'inverse, les motifs temporels sont composés d'une succession de motifs en niveau de gris, rendant ainsi ce codage robuste à la texture mais très sensible aux mouvements des objets. Le dernier type d'encodage, qui peut être en couleur ou en niveau de gris, utilise le voisinage de chaque pixel qui a la particularité d'être unique dans l'image. Cette méthode a l'avantage d'être robuste à la texture et aux mouvements mais la complexité des algorithmes de décodage peut parfois poser des problèmes au niveau du temps de calcul. Le choix du motif à projeter dépend donc beaucoup de l'utilisation

FIGURE 2.3 – *Motifs de lumière structurée.*

que l'on souhaite faire du scanner : objets statiques, mobiles, acquisitions rapides, etc. Les systèmes à lumière structurée sont les scanners les plus fréquemment utilisés par les fabricants avec les systèmes à triangulation active.

#### 2.1.2.2/ AUTRES SYSTÈMES

Tout comme pour les systèmes passifs, seul les scanners à lumière structurée nous concerne dans le cadre de ce projet. Il existe cependant bien d'autres techniques actives permettant de reconstruire un objet en 3D. Citons entre autres la triangulation laser [29] ou encore les systèmes à temps de vol.

#### 2.1.3/ SYNTHÈSE

Toutes les techniques qui viennent d'être présentées ont fait l'objet de nombreuses recherches et la plupart sont utilisées par les fabricants de scanner 3D. Comme cela a été mentionné plusieurs fois, chaque système a ses avantages et ses inconvénients. Sansoni *et al.* [30] présentent un tableau comparatif des avantages et inconvénients des différentes méthodes (tableau 2.1).

Ce tableau permet de se faire une idée du type de technologie vers laquelle s'orienter en fonction de l'application visée. D'un point de vue général, les systèmes optiques présentent des avantages très intéressants : possibilité de numériser un objet sans contact, portabilité du système qui permet de l'utiliser n'importe où, bonne précision de mesure y compris pour des cas industriels, existence de différentes technologies adaptées à la plupart des objets (objets de grande taille, objets de taille microscopique, objets réfléchissants, ...).

<b>Technologies</b>	<b>Avantages</b>	<b>Inconvénients</b>
<b>Stéréovision</b>	<ul style="list-style-type: none"> <li>- Simple à mettre en œuvre</li> <li>- Bonne précision sur des cibles bien définies</li> </ul>	<ul style="list-style-type: none"> <li>- Temps de calcul</li> <li>- Données éparses</li> <li>- Limité à certaines scènes</li> <li>- Faible densité d'acquisition</li> </ul>
<b>Photogrammétrie</b>	<ul style="list-style-type: none"> <li>- Simple à mettre en œuvre</li> <li>- Bonne précision sur des cibles bien définies</li> </ul>	<ul style="list-style-type: none"> <li>- Temps de calcul</li> <li>- Données éparses</li> <li>- Limité à certaines scènes</li> <li>- Faible densité d'acquisition</li> </ul>
<b>Shape from Shading</b>	<ul style="list-style-type: none"> <li>- Simple à mettre en œuvre</li> </ul>	<ul style="list-style-type: none"> <li>- Faible précision</li> </ul>
<b>Shape from Texture</b>	<ul style="list-style-type: none"> <li>- Simple à mettre en œuvre</li> </ul>	<ul style="list-style-type: none"> <li>- Faible précision</li> </ul>
<b>Shape from Motion</b>	<ul style="list-style-type: none"> <li>- Simple à mettre en œuvre</li> </ul>	<ul style="list-style-type: none"> <li>- Moins bonne précision que la stéréovision</li> <li>- Données éparses</li> </ul>
<b>Triangulation Laser</b>	<ul style="list-style-type: none"> <li>- Simple à mettre en œuvre</li> <li>- Pas d'influence de la lumière ambiante</li> <li>- Bonne densité</li> </ul>	<ul style="list-style-type: none"> <li>- Sécurité liée au laser</li> <li>- Volume de mesure limité</li> <li>- Prix</li> <li>- Présence d'occlusions, d'ombres</li> </ul>
<b>Temps de vol</b>	<ul style="list-style-type: none"> <li>- Grand volume de mesure</li> <li>- Pas d'influence de la lumière ambiante</li> <li>- Bonne densité</li> </ul>	<ul style="list-style-type: none"> <li>- Prix</li> <li>- Précision plus faible que la triangulation</li> </ul>
<b>Interférométrie</b>	<ul style="list-style-type: none"> <li>- Précision inférieure au <math>\mu m</math></li> </ul>	<ul style="list-style-type: none"> <li>- Limité à des surfaces presque planes</li> <li>- Prix</li> </ul>
<b>Franges de Moiré</b>	<ul style="list-style-type: none"> <li>- Simple à mettre en œuvre</li> </ul>	<ul style="list-style-type: none"> <li>- Limité à des surfaces lisses</li> </ul>
<b>Lumière structurée</b>	<ul style="list-style-type: none"> <li>- Acquisition dense</li> <li>- Volume de mesure moyen</li> <li>- Performances peu influencées par la lumière ambiante</li> </ul>	<ul style="list-style-type: none"> <li>- Complexité du calcul</li> <li>- Présence d'ombres, d'occlusions</li> <li>- Prix</li> </ul>

TABLE 2.1 – Tableau comparatif des différentes techniques optiques de numérisation 3D [30].

## 2.2/ FONCTIONNEMENT D'UN SCANNER

Dans le cadre du projet qui englobe cette thèse, la société Noomeo a mis à notre disposition un scanner 3D. Ce scanner est un système composé de deux caméras et utilisant le principe de la stéréovision aidée par une projection de texture pseudo-aléatoire. Afin de bien comprendre le fonctionnement d'un scanner, et plus précisément de ce type de scanner, ainsi que certaines notions qui seront utilisées par la suite dans ce mémoire, il convient de poser les bases du fonctionnement de ce scanner.

### 2.2.1/ LE MODÈLE DU STÉNOPÉ

Un scanner 3D optique est généralement composé d'une ou plusieurs caméras, qui sont elles-mêmes composées d'un capteur photographique (de type CMOS ou CCD), d'un objectif optique et d'un zoom. Par souci de simplification, les caméras sont généralement modélisées par un modèle décrivant le processus de formation des images. Le modèle le plus simple et le plus répandu en vision par ordinateur [31, 32] est le modèle sténopé, ou modèle « *pinhole* », qui modélise une caméra par projection perspective (figure 2.4).

Ce modèle permet de transformer un point 3D  $Q$  de l'espace monde  $R_M$  en un point image  $q$  dans le repère image  $R_i$ . Cette transformation peut se décomposer en trois transformations élémentaires successives symbolisées par les flèches 1, 2 et 3 sur la figure 2.4. La première transformation s'effectue entre le repère monde  $R_M$  et le repère de la caméra  $R_c$ , la deuxième transformation entre le repère caméra  $R_c$  et le repère du capteur  $R_r$  et enfin la dernière transformation entre le capteur  $R_r$  et le repère image  $R_i$ .

#### 2.2.1.1/ DU REPÈRE MONDE AU REPÈRE CAMÉRA

La transformation entre le repère monde  $R_M$  et le repère caméra  $R_c$  (dont l'origine est située au centre optique de la caméra) est symbolisée par la flèche 1 sur la figure 2.4. Cette transformation peut être décomposée par une rotation notée  $[R]$  et une translation notée  $[t]$ . En vision par ordinateur, on utilise généralement les coordonnées homogènes car elles permettent d'exprimer le modèle sténopé par une relation linéaire. La relation entre  $R_M$  et  $R_c$  s'écrit alors :

$$\begin{pmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{pmatrix} = R \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} + t = \begin{bmatrix} & & & t_x \\ & R_{3 \times 3} & & t_y \\ & & & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = T_1 \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}, \quad (2.1)$$

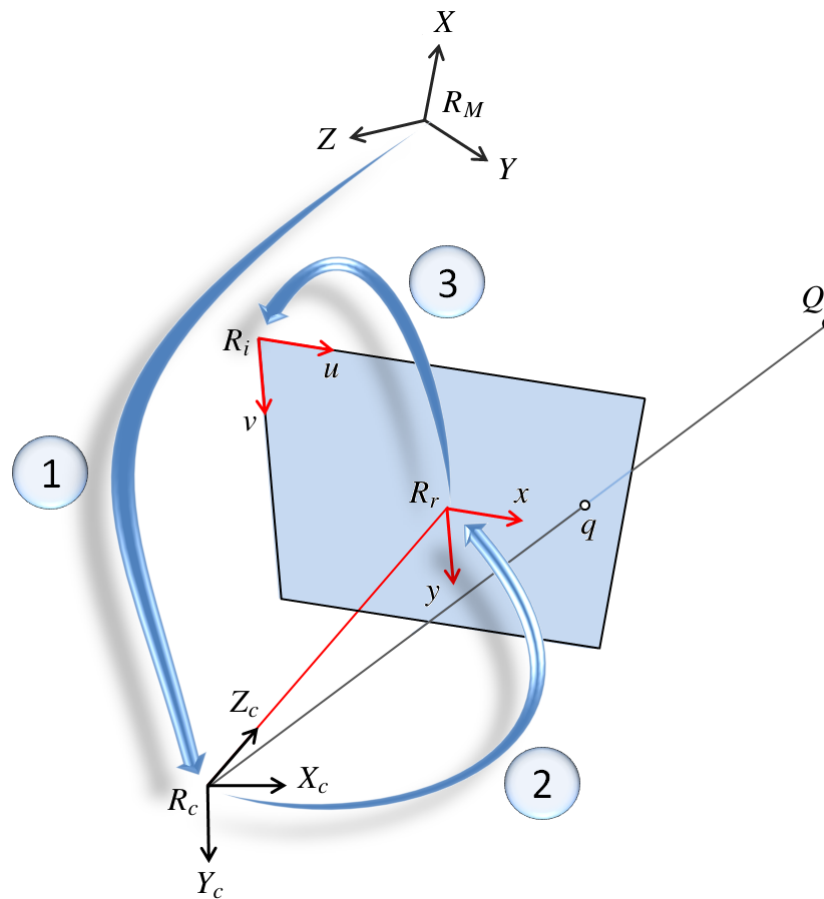


FIGURE 2.4 – Modélisation d'une caméra : modèle sténopé.

avec

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}. \quad (2.2)$$

Les paramètres de la matrice de transformation  $T_1$  entre le repère monde et le repère caméra sont communément appelés paramètres extrinsèques.

### 2.2.1.2/ DU REPÈRE CAMÉRA AU REPÈRE CAPTEUR

Le passage du repère caméra  $R_c$  au repère capteur  $R_r$  est une projection perspective qui permet de transformer un point 3D  $(X_c, Y_c, Z_c)$  en un point image  $(x, y)$ . Cette transformation est représentée par la flèche 2 sur la figure 2.4 et s'écrit de la façon suivante :



$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{pmatrix} = T_2 \begin{pmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{pmatrix}, \quad (2.3)$$

avec  $f$  la distance focale de l'objectif de la caméra. Le point  $q(x, y)$  est exprimé en unité métrique.

### 2.2.1.3/ DU REPÈRE CAPTEUR AU REPÈRE IMAGE

La troisième et dernière transformation, la flèche 3 sur la figure 2.4, permet de convertir un point  $q(x, y)$ , exprimé en unité métrique dans le repère capteur  $R_r$ , en un point  $q(u, v)$  exprimé en pixel dans le repère image  $R_i$ . La transformation entre ces deux repères s'écrit :

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{bmatrix} k_x & s & x_0 \\ 0 & k_y & y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = T_3 \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}, \quad (2.4)$$

où :

- $k_x$  et  $k_y$  désignent la densité de pixels par unité de longueur selon les directions  $x$  et  $y$  du capteur de la caméra. Dans le cas idéal, où les pixels sont carrés, on a  $k_x = k_y$ .
- $x_0$  et  $y_0$  sont les coordonnées de l'intersection entre l'axe optique de la caméra et le plan image. Il s'agit donc de l'origine du repère capteur  $R_r$ . Dans un cas parfait,  $x_0$  et  $y_0$  sont au centre de l'image.
- $s$ , appelé « *screw factor* » en anglais, est le facteur de non-orthogonalité entre les lignes et les colonnes des pixels du capteur de la caméra. En général, ce paramètre est très faible et il est donc très souvent négligé.

### 2.2.1.4/ MODÈLE COMPLET

Le modèle sténopé complet consiste donc à passer d'un point 3D  $Q(X, Y, Z)$  exprimé dans le repère monde  $R_M$  à un point image  $q(u, v)$  exprimé dans le repère image  $R_i$  en passant par les trois transformations vues précédemment. Pour un point  $q(u, v)$  on peut alors écrire :

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = T_3 T_2 T_1 \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}. \quad (2.5)$$

En développant (2.6), on obtient :

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{bmatrix} k_x & s & x_0 \\ 0 & k_y & y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R_{3 \times 3} & \begin{matrix} t_x \\ t_y \\ t_z \end{matrix} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}, \quad (2.6)$$

qui peut se mettre sous la forme suivante :

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{bmatrix} f_x & s & x_0 & 0 \\ 0 & f_y & y_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R_{3 \times 3} & \begin{matrix} t_x \\ t_y \\ t_z \end{matrix} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = K \begin{bmatrix} R_{3 \times 3} & \begin{matrix} t_x \\ t_y \\ t_z \end{matrix} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}. \quad (2.7)$$

La matrice  $K$  est composée exclusivement de paramètres propres à la caméra, on parle alors de paramètres intrinsèques. On trouve souvent le modèle sténopé écrit sous la forme simplifiée suivante :

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = K \begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = P_c \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}, \quad (2.8)$$

avec  $K$  la matrice des paramètres intrinsèques,  $[R|t]$  les paramètres extrinsèques. La matrice  $P_c$  qui englobe les deux matrices  $K$  et  $[R|t]$  est appelée matrice de projection. Un système de stéréovision étant composé de deux caméras, il convient de différencier les différents repères et matrices de projection afin de ne pas les confondre (figure 2.5).

Généralement, pour les distinguer, on ajoute un indice différent pour chaque caméra. Ainsi dans le cas de la première caméra, la projection du point  $Q(X, Y, Z)$  dans le repère image sera notée  $q_0$ , la projection du repère monde au repère image  $P_{c0}$  et les différents repères  $R_{c0}$ ,  $R_{r0}$  et  $R_{i0}$ . De la même façon, pour la seconde caméra, les repères seront notés  $R_{c1}$ ,  $R_{r1}$  et  $R_{i1}$ , la matrice de projection  $P_{c1}$  et la projection du point  $Q$  dans l'image

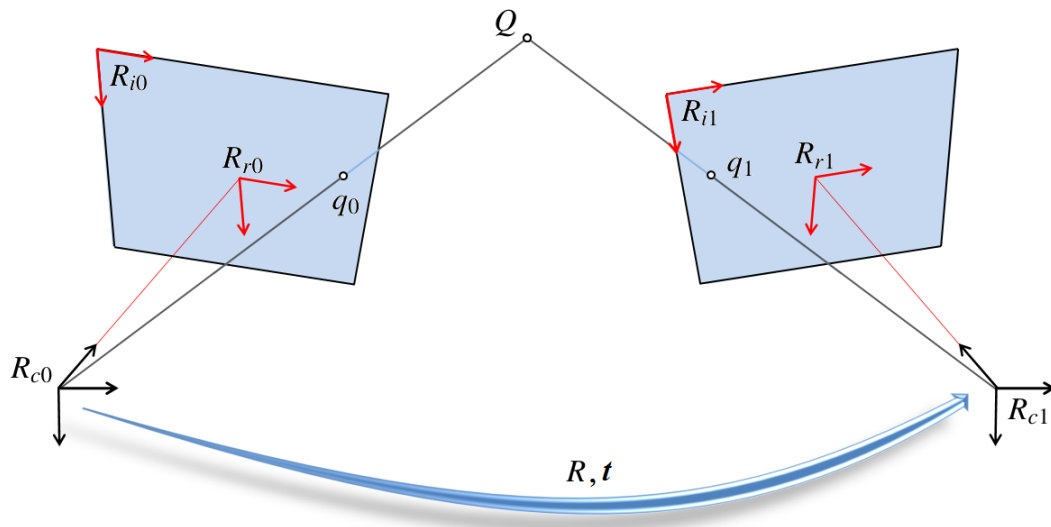


FIGURE 2.5 – Modélisation d'un système de stéréovision.

sera notée  $q_1$ . Afin de simplifier les notations, le repère monde  $R_M$  est généralement confondu avec le repère caméra  $R_{c0}$ . De ce fait, la matrice de projection  $P_{c0}$  ne contient plus que les paramètres intrinsèques de la première caméra, et la relation entre  $R_{c0}$  et  $R_{c1}$  correspond à une rotation et une translation notée  $[R|t]$  (à ne pas confondre avec les paramètres extrinsèques des caméras qui sont désormais notés  $[R_0|t_0]$  et  $[R_1|t_1]$ ).

#### 2.2.1.5/ LES DISTORSIONS

Dans tout système optique, les objectifs créent inévitablement des distorsions au niveau des images. Le modèle sténopé que nous venons de présenter modélise une caméra idéale et ne tient donc pas compte des distorsions géométriques induites par l'objectif. Cependant, Horst Beyer [33] a montré que dans le cas d'applications métrologiques (ce qui est le cas de la numérisation 3D), il est indispensable de prendre en compte les distorsions afin de les corriger.

Il existe plusieurs approches permettant de modéliser les distorsions d'une caméra. Dans la plupart des cas, l'approche utilisée est une approche paramétrique qui consiste à modéliser la distorsion par des termes supplémentaires. L'une de ces méthodes, présentée par Weng *et al.* [34], s'inspire des aberrations géométriques des systèmes centrés en ajoutant au modèle des termes correctifs correspondant à différents types de distorsions : radiales, prismatiques ou de décentrage.

Dans le modèle sténopé, la distorsion peut être modélisée par une quatrième transformation  $D$  qui consiste à corriger l'image en fonction des termes correctifs liés à la distorsion induite par l'objectif. Le modèle complet peut alors être schématisé sous la forme suivante :

$$Q(X, Y, Z) \xrightarrow{T_1} Q_c(X_c, Y_c, Z_c) \xrightarrow{T_2} q(x, y) \xrightarrow{D} q(x', y') \xrightarrow{T_3} q(u', v'),$$

où  $u'$  et  $v'$  représentent les coordonnées pixelliques corrigées du point  $q$ . Toujours dans le modèle présenté par Weng, la distorsion  $D$  correspond à la somme des distorsions radiales, prismatiques et de décentrage et peut s'écrire sous la forme suivante :

$$\delta x = \delta x_r + \delta x_p + \delta x_d, \quad \delta y = \delta y_r + \delta y_p + \delta y_d, \quad (2.9)$$

avec  $\delta x_r$  et  $\delta y_r$  la distorsion radiale selon les axes  $x$  et  $y$ ,  $\delta x_p$  et  $\delta y_p$  la distorsion prismatique, puis  $\delta x_d$  et  $\delta y_d$  la distorsion de décentrage. Dans la modélisation présentée par Weng, les différentes distorsions s'écrivent de la façon suivante :

$$\begin{cases} \delta x_r = k_1 x(x^2 + y^2) \\ \delta y_r = k_1 y(x^2 + y^2) \end{cases}, \quad (2.10)$$

$$\begin{cases} \delta x_p = s_1(x^2 + y^2) \\ \delta y_p = s_2(x^2 + y^2) \end{cases}, \quad (2.11)$$

$$\begin{cases} \delta x_d = p_1(3x^2 + y^2) + 2p_2xy \\ \delta y_d = 2p_1xy + p_2(x^2 + y^2) \end{cases}, \quad (2.12)$$

avec  $k_1$ ,  $d_1$ ,  $d_2$ ,  $p_1$  et  $p_2$  les paramètres de distorsions. Ces paramètres peuvent être estimés par une étape d'étalonnage de la caméra. Il est alors possible d'effectuer la transformation passant d'un point image distordu  $q(x, y)$  à un point image non-distordu  $q(x', y')$  et ainsi corriger la distorsion de l'image de la façon suivante :

$$x' = x + \delta x, \quad y' = y + \delta y. \quad (2.13)$$

Il est important de noter que ce modèle, bien que très utilisé n'est pas le seul modèle existant. Il est possible de trouver d'autres modélisations, notamment celles présentées par Tsai [35] et Faugeras [36].

### 2.2.1.6/ CALIBRAGE DU SYSTÈME

Quel que soit le système de numérisation utilisé, pour pouvoir effectuer une reconstruction 3D il faut avant tout connaître la matrice de projection  $P_c$ . Pour cela, il faut passer par une étape de calibrage, qui permet de déterminer les paramètres intrinsèques (y compris les paramètres de distorsion) et extrinsèques de la caméra. L'idée générale du calibrage d'une caméra consiste à calculer directement la matrice de projection  $P_c$  ou encore de calculer séparément les paramètres intrinsèques  $K$  et extrinsèques  $[R|t]$  en connaissant plusieurs points 3D  $Q_i(X_i, Y_i, Z_i)$  et leur observation dans l'image  $q_i(u'_i, v'_i)$ . Il existe actuellement de très nombreuses méthodes et toolbox [37, 38] permettant de calibrer une caméra. Salvi, Zhang et Remondino [39, 40, 41] propose un état de l'art ainsi qu'un comparatif des différentes techniques existantes.

### 2.2.2/ MISE EN CORRESPONDANCE

Une mesure tridimensionnelle avec un système de stéréovision s'établit en trois étapes : la calibration du système, la mise en correspondance des points entre les images et le calcul de la position du point 3D par triangulation. La mise en correspondance de pixels consiste à retrouver sur deux images d'une même scène, prises avec des points de vue différents, les deux pixels qui correspondent à la projection d'un point de la scène. La précision de la mesure est directement dépendante de la calibration et de la qualité de l'appariement des points entre les images. Cette dernière étape peut s'avérer compliquée car plusieurs difficultés peuvent entraîner des résultats erronés comme le bruit de l'image, les variations d'intensité ou encore les déformations perspectives entre les deux images.

#### 2.2.2.1/ RECTIFICATION DES IMAGES

Actuellement, il existe de nombreuses méthodes permettant de mettre des points en correspondance entre deux images. Cependant, la grande majorité des techniques vont, au préalable, rectifier les images afin de faciliter la mise en correspondance ; on parle de contrainte épipolaire. La notion de géométrie épipolaire est primordiale dans l'étape de mise en correspondance des points. Chercher le correspondant d'un point dans toute une image peut s'avérer extrêmement coûteux en temps de calcul si cette opération est effectuée pour tous les pixels. La contrainte épipolaire permet de simplifier le problème. En effet, Faugeras [36] a démontré que le correspondant d'un point  $q_0$  dans une image se trouve sur une ligne  $l_1$  (appelée ligne épipolaire) dans la seconde image. De ce fait, il est possible de réduire la zone de recherche du correspondant de  $q_0$  dans la seconde

image à une ligne plutôt qu'à l'image entière (figure 2.6). Les lignes épipolaires  $l_0$  et  $l_1$  sont obtenues par les relations suivantes :

$$l_0 = E \times P_{c0} \quad (2.14)$$

$$l_1 = E^T \times P_{c1}, \quad (2.15)$$

avec  $E$  la matrice essentielle, qui peut être obtenue lors de l'étape de calibrage. Le principe de la rectification d'images consiste à placer les épiholes  $e_0$  et  $e_1$  à l'infini, ce qui a pour effet de projeter les plans images dans un seul et même plan : on parle alors d'images rectifiées. L'appariement des points se fait toujours selon la ligne épipolaire, mais cette fois-ci cette dernière correspond à une seule et même ligne  $l$  dans les deux images, ce qui simplifie fortement les calculs et les algorithmes (figure 2.7). De plus, généralement les distorsions sont prises en compte lors de l'étape de rectification, permettant ainsi de revenir à un modèle sténopé classique sans distorsion, comme présenté précédemment.

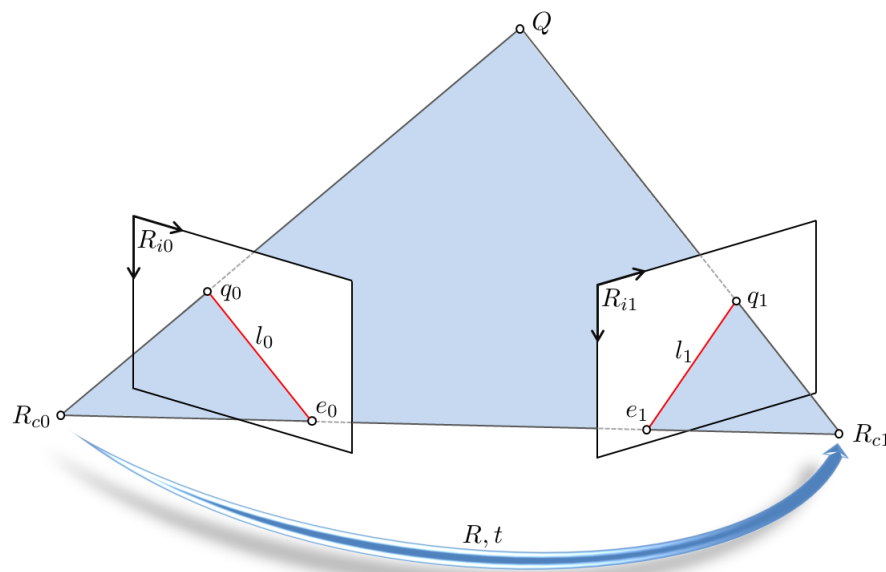


FIGURE 2.6 – Illustration de la géométrie épipolaire. En rouge, les lignes épipolaires et en bleu, le plan épipolaire.

La mise en correspondance stéréoscopique est un domaine qui a été fortement étudié ces dernières années et de nombreuses techniques ont été proposées. Il est possible de regrouper les différentes techniques d'appariement en deux grandes familles : les méthodes locales et les méthodes globales.

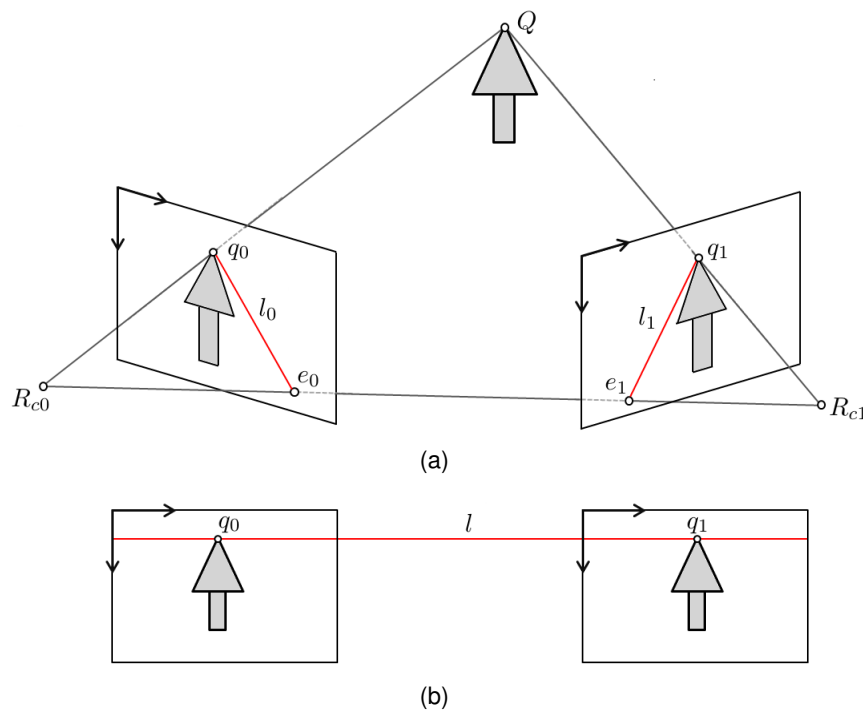


FIGURE 2.7 – Rectification des images. a) Images non rectifiées. b) Images rectifiées.

#### 2.2.2.2/ MÉTHODES LOCALES CLASSIQUES

Ces méthodes recherchent, par corrélation, les points dont le voisinage local est le plus ressemblant entre les deux images. Le candidat dont le score de corrélation est le plus élevé pour un point donné  $q_0$  est retenu comme correspondant, noté  $q_1$ . L'utilisation des images rectifiées permet de réduire la recherche du candidat à une seule ligne de l'image. Cependant, ces méthodes ayant toujours un candidat (le meilleur score), les erreurs peuvent être assez fréquentes. Afin de les réduire, la plupart des approches ajoutent des contraintes afin d'éliminer les candidats les moins fiables ne vérifiant pas certaines contraintes :

- La contrainte sur le score de corrélation : seuls les candidats ayant un score de corrélation supérieur à un seuil sont retenus.
- La contrainte d'ordre : l'ordre des pixels dans l'image de référence et des candidats dans la seconde image doit être le même.
- La contrainte d'unicité : un pixel ne peut être appairé qu'à un seul pixel dans la seconde image.
- La contrainte de symétrie : les appariements trouvés doivent être les mêmes quelle que soit l'image de référence.
- La contrainte du gradient de disparité : les disparités ne doivent pas varier brusquement entre deux pixels voisins.

Les résultats des méthodes locales classiques peuvent encore être affinés en adaptant la méthode de corrélation ainsi que la fenêtre de corrélation au contexte. Il existe beaucoup de mesures de corrélation différentes et chacune d'entre elles est plus ou moins robuste en fonction des difficultés rencontrées.

### 2.2.2.3/ MÉTHODES LOCALES PAR PROPAGATION DE GERMES

Ces méthodes sont un cas particulier des méthodes locales classiques où la recherche ne s'effectue plus sur toute la ligne épipolaire mais dans une zone encore plus réduite. En supposant que les pixels voisins ont des disparités proches, il est alors possible de rechercher le correspondant d'un point, appelé germe, dans un voisinage d'appariements fiables plutôt que sur toute la ligne épipolaire. Il s'agit de méthodes de propagation itératives où les points ainsi appairés servent à leur tour de germes lors de l'itération suivante. Il existe deux approches de propagation différentes :

- L'approche simultanée : cette approche consiste à propager un ensemble de germes à chaque itération. Les points appairés de cette façon servent alors de nouveaux germes pour l'itération suivante.
- L'approche séquentielle : à chaque itération, un seul germe sera propagé. Le germe est sélectionné par appariement de points d'intérêt. Cette approche a pour avantage de propager moins d'erreurs lorsqu'il existe un germe erroné, mais c'est une approche qui demande plus d'itération pour obtenir une reconstruction dense.

L'un des inconvénients des méthodes par propagation est le risque de propager des erreurs à cause de germes incorrects. Pour limiter au maximum cet effet, la plupart des méthodes ajoutent des contraintes supplémentaires qui empêchent la propagation si les correspondances sont jugées peu fiables. Ces contraintes sont la contrainte d'homogénéité, qui empêche la croissance de région dans les zones homogènes (sources d'erreurs importantes), et la contrainte de région qui suppose que les pixels voisins ayant des valeurs proches sont les projections de points d'une même surface et ont des disparités proches. Cette hypothèse permet de supposer que les discontinuités de profondeur coïncident avec les contours des régions.



#### 2.2.2.4/ MÉTHODES GLOBALES

A l'inverse des méthodes locales, les méthodes globales ne tiennent plus compte du voisinage de chaque point, elles ont pour objectif de minimiser une fonction de coût représentant l'erreur induite par les appariements trouvés. Le principe de ces méthodes consiste donc à trouver la fonction de correspondance  $m$  minimisant l'erreur globale de mise en correspondance. Cette erreur est quantifiée par une fonction de coût qui intègre un terme vérifiant que les valeurs de deux pixels correspondants soient proches, et un terme de lissage assurant la contrainte de la limite du gradient de disparité. Une fois la fonction de coût définie, il est alors possible de minimiser cette dernière à l'aide de techniques d'optimisation mathématique.

#### 2.2.2.5/ TEXTURE ALÉATOIRE

Les méthodes locales étant basées sur une corrélation entre images, elles sont fortement dépendantes de la texture de l'objet. Dès que l'on souhaite obtenir une bonne précision, il est nécessaire que l'objet numérisé soit suffisamment texturé, ce qui est rarement le cas concernant les pièces industrielles. Pour pallier ce problème, il est possible de projeter une texture connue (aussi appelée mouchetis) à l'aide d'une source lumineuse sur l'objet à numériser. Le motif le plus connu et le plus utilisé est le bruit de Perlin [42, 43]. Le bruit de Perlin est une fonction de bruit basée sur des gradients pseudo aléatoires. Un gradient pseudo aléatoire est appliqué à une grille de points régulière (figure 2.8). Les valeurs des pixels sont ensuite obtenus par interpolation du produit scalaire des gradients des sommets voisins du pixel concerné (les gradients sont les vecteurs rouges sur la figure 2.8), ainsi que des vecteurs reliant le pixel concerné aux sommets proches (en vert).

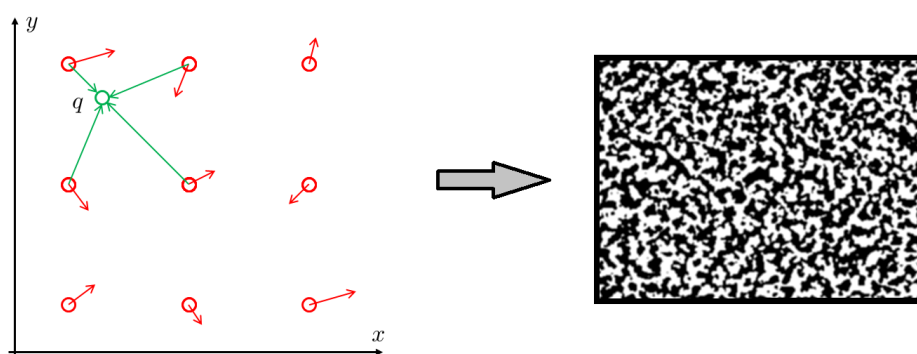


FIGURE 2.8 – Conception d'un bruit de Perlin en deux dimensions. La valeur de chaque pixel est déterminée à partir des gradients de chaque sommet voisin. L'image de droite est le résultat obtenu par cette méthode.

Un autre avantage à l'utilisation d'un bruit de Perlin réside dans le fait que ce dernier possède un ensemble de paramètres contrôlables (la taille des taches, le nombre de niveaux de gris, l'intensité, ...). Ces différents paramètres peuvent être ajustés selon les pièces à numériser afin de faciliter encore plus la mise en correspondance pour des pièces difficiles.

### 2.2.2.6/ SYNTHÈSE

Bien que le problème de la mise en correspondance stéréoscopique ait été étudié depuis de nombreuses années, il n'en reste pas moins une problématique encore d'actualité. De nouvelles techniques continuent d'apparaître permettant un appariement des pixels de plus en plus précis et de plus en plus robuste. Scharstein *et al.* [44] présentent une étude évaluant et comparant les différentes méthodes de mise en correspondance dense. Si les méthodes de corrélation denses sont généralement destinées à appairer chaque pixel de l'image, il est tout de même possible de contrôler la densité de points que l'on souhaite obtenir en appariant moins de pixels (un pixel sur deux par exemple).

### 2.2.3/ TRIANGULATION DES POINTS

La triangulation est une technique utilisée depuis l'antiquité permettant de déterminer la position d'un point dans l'espace. Une fois les points appairés entre les deux images, l'étape suivante d'un scanner consiste à trianguler les points pour obtenir la position tridimensionnelle de chaque point  $Q$  en connaissant la position de deux autres points de référence  $q_0$  et  $q_1$  ainsi que les angles  $\alpha$  et  $\beta$  (figure 2.9).

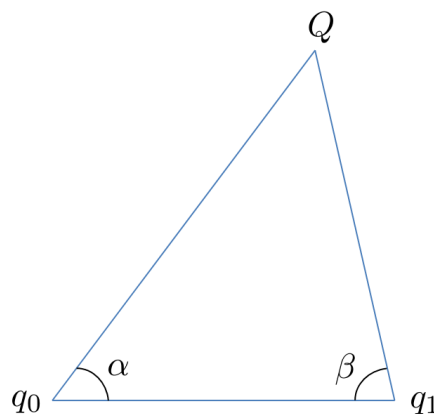


FIGURE 2.9 – Principe de la triangulation. Connaissant  $q_0$ ,  $q_1$ ,  $\alpha$  et  $\beta$ , il est possible de calculer la position du point  $Q$ .

Dans le cas nous concernant, les points de référence sont les pixels appairés  $q_0$  et  $q_1$  lors de l'étape de mise en correspondance et les angles sont pris en compte dans les matrices de projection  $P_{c0}$  et  $P_{c1}$ . De plus, les distorsions sont également prises en compte dans l'étape de rectification. Considérons un point  $q_0(u_0, v_0)$  dans le repère de l'image rectifiée de la première caméra, comme la projection d'un point  $Q(X, Y, Z)$ . On peut alors écrire :

$$\begin{pmatrix} u_0 \\ v_0 \\ 1 \end{pmatrix} = P_{c0} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}. \quad (2.16)$$

Supposons que  $q_1(u_1, v_1)$  soit le correspondant de  $q_0(u_0, v_0)$  et la projection du point  $Q(X, Y, Z)$  dans le repère l'image de la seconde caméra, on obtient alors :

$$\begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix} = P_{c1} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}. \quad (2.17)$$

Etant donné qu'il s'agit du même point  $Q$ , on peut alors écrire le système suivant en développant les équations (2.16) et (2.17) (pour simplifier les notations on notera  $P_{c0}$  et  $P_{c1}$ , respectivement  $P$  et  $P'$  dans cette partie) :

$$\begin{cases} u_0 = P_{11}X + P_{12}Y + P_{13}Z + P_{14} \\ v_0 = P_{21}X + P_{22}Y + P_{23}Z + P_{24} \\ 1 = P_{31}X + P_{32}Y + P_{33}Z + P_{34} \\ u_1 = P'_{11}X + P'_{12}Y + P'_{13}Z + P'_{14} \\ v_1 = P'_{21}X + P'_{22}Y + P'_{23}Z + P'_{24} \\ 1 = P'_{31}X + P'_{32}Y + P'_{33}Z + P'_{34} \end{cases}, \quad (2.18)$$

qui peut également se mettre sous la forme suivante :

$$\begin{cases} (P_{11} - u_0 P_{31})X + (P_{12} - u_0 P_{32})Y + (P_{13} - u_0 P_{33})Z + (P_{14} - u_0 P_{34}) = 0 \\ (P_{21} - v_0 P_{31})X + (P_{22} - v_0 P_{32})Y + (P_{23} - v_0 P_{33})Z + (P_{24} - v_0 P_{34}) = 0 \\ (P'_{11} - u_1 P'_{31})X + (P'_{12} - u_1 P'_{32})Y + (P'_{13} - u_1 P'_{33})Z + (P'_{14} - u_1 P'_{34}) = 0 \\ (P'_{21} - v_1 P'_{31})X + (P'_{22} - v_1 P'_{32})Y + (P'_{23} - v_1 P'_{33})Z + (P'_{24} - v_1 P'_{34}) = 0 \end{cases} \quad (2.19)$$

Les matrices de projection  $P$  et  $P'$  étant connues grâce à l'étape de calibrage et  $u_0, v_0, u_1, v_1$  grâce à l'étape de mise en correspondance, il est alors possible de résoudre

ce système par la méthode des moindres carrés. Ainsi, on obtient  $X, Y, Z$  et donc la position du point  $Q$  dans l'espace à partir des données de calibration et de la mise en correspondance des points.

## 2.3/ EXTRACTION D'INFORMATIONS COMPLÉMENTAIRES

Dans la chaîne 3D classique, l'étape suivant l'acquisition est l'étape d'extraction d'informations complémentaires aux points 3D. En effet, malgré l'évolution des technologies, la plupart du temps les scanners 3D se retrouvent limités en résolution à cause des caméras. Dans l'optique d'obtenir toujours plus d'informations, de nombreuses méthodes permettant d'extraire des informations complémentaires aux points 3D ont fait leur apparition au cours des dernières années. Les informations complémentaires les plus fréquemment calculées sont les normales à la surface et la courbure. Ces différentes informations permettent d'obtenir une plus grande précision sur le modèle 3D sans pour autant qu'il soit nécessaire d'ajouter des points. De plus, ces attributs sont de plus en plus fréquemment utilisés par les différents algorithmes permettant de simplifier les données.

### 2.3.1/ NORMALE À LA SURFACE

La normale à la surface décrit l'orientation locale de la surface. Il s'agit probablement de l'information complémentaire la plus importante et la plus utilisée, à tel point que désormais de plus en plus de fabricants de scanner 3D fournissent cette information dans le fichier de sortie du scanner. De ce fait, un grand nombre d'algorithmes de débruitage [45] ou de reconstruction de surface [46, 47] nécessitent en tous points un vecteur unitaire normal à la surface échantillonnée pour fonctionner. Il existe différentes approches permettant de calculer les normales à la surface, soit directement sur le nuage de points, après avoir maillé le nuage de points, ou encore directement depuis les images fournies par le scanner.

#### 2.3.1.1/ À PARTIR DU NUAGE DE POINTS

Hoppe *et al.* [48] présentent l'une des méthodes les plus connues et les plus utilisées pour estimer les normales à la surface à partir d'un nuage de points. Le principe de cette méthode consiste à ajuster des plans locaux centrés sur un point à partir de son voisinage. Considérons un point  $p$  du nuage de points dont on souhaite estimer la

normale, et  $\mathcal{N}(p)$  un ensemble de points voisins. Le voisinage  $\mathcal{N}(p)$  peut être défini soit en prenant les  $k$  plus proches voisins de  $p$  ou alors les points voisins contenus dans une sphère de rayon  $r$  centrée sur  $p$ . Si le voisinage de  $p$  est suffisamment petit et que la surface est supposée localement lisse, il est possible de l'approximer par un plan. La normale à la surface peut alors être calculée comme le vecteur normal au plan s'ajustant le mieux sur l'ensemble de points  $\mathcal{N}(p)$ .

L'ajustement de ce plan s'effectue par analyse en composantes principales (ACP). L'ACP [49, 50] est une méthode issue des statistiques permettant de calculer les axes qui décrivent le mieux les données. Concrètement, les axes se calculent comme étant les vecteurs propres de la matrice de covariance de  $\mathcal{N}(p)$ . La normale est associée au vecteur propre ayant la plus petite valeur propre. L'ACP fournit en plus une mesure de dispersion des points le long de chaque axe via les valeurs propres de la matrice de covariance, permettant ainsi d'évaluer la planéité du voisinage  $\mathcal{N}(p)$  et donc la fiabilité des normales calculées. Le problème majeur de cette approche réside dans le choix du voisinage. Si le voisinage est trop restreint, l'effet du bruit sur le nuage de points n'est pas compensé par la taille de la zone étudiée et ne permet donc pas un ajustement robuste du plan. A contrario, si ce dernier est trop grand, il permet effectivement de réduire l'effet du bruit mais il ne satisfait plus l'hypothèse selon laquelle le voisinage est assimilable à un plan. Pour résoudre ce problème, des méthodes adaptatives ont été développées, permettant de choisir localement la taille du voisinage la plus appropriée [51]. Ces techniques sont cependant plus coûteuses en temps de calcul.

Alexa *et al.* [52] présentent également une évolution de ces techniques consistant à pondérer les points du voisinage  $\mathcal{N}(p)$  de manière décroissante en fonction de leur distance à  $p$ . Les points les plus éloignés se voient ainsi attribuer une importance moindre dans l'évaluation du plan tangent à  $\mathcal{N}(p)$ . Malgré tout, cette approche possède un inconvénient, elle favorise les points les plus proches sans tenir compte de leur pertinence réelle.

Les normales obtenues avec ces méthodes souffrent d'un effet de lissage, induit par l'ajustement des plans sur l'ensemble du voisinage. Ainsi, les normales de deux points voisins ne peuvent pas être radicalement différentes étant donné qu'elles partagent un voisinage commun important (les plans moyens seront donc proches). De ce fait, ces techniques vont avoir tendance à lisser les arêtes saillantes. Berger *et al.* [53] présentent un état de l'art des différentes méthodes permettant de calculer les normales à la surface directement sur le nuage de points.

### 2.3.1.2/ A PARTIR D'UN MAILLAGE

Un maillage 3D est la modélisation d'un objet dans un espace tridimensionnel, il est formé d'un ensemble d'éléments. Un maillage est composé de points (il s'agit des points du nuage de points), d'arêtes (c'est un segment qui relie deux points) et de faces (c'est un polygone composé de plusieurs arêtes). Dans la plupart des cas, les faces sont triangulaires (figure 2.10) mais il existe également d'autres représentations (quadrangles, etc). Dans le cas d'un maillage triangulaire, la surface du polygone est très souvent modélisée par un plan, bien qu'il soit possible de la modéliser par une surface plus complexe comme des splines par exemple [54].

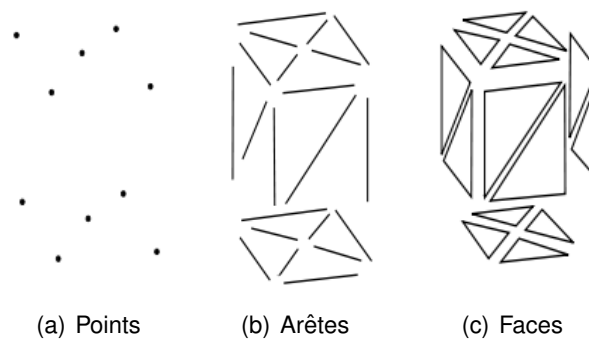


FIGURE 2.10 – Éléments qui composent un maillage 3D.

Lors des étapes de post-traitement, la plupart des techniques existantes sont appliquées sur des maillages plutôt que sur le nuage de points. En effet, un maillage possède une information de voisinage ainsi qu'une représentation surfacique à l'aide des surfaces qui le composent. De ce fait, on trouve de plus en plus de scanners commerciaux qui fournissent des données maillées à la sortie de l'acquisition. Ces maillages étant très souvent irréguliers (irrégularité du voisinage), ils sont généralement remaillés lors d'une étape supplémentaire. Roudet *et al.* [55] présentent un état de l'art des différentes méthodes de remaillage existantes pour des maillages triangulaires. A partir d'un maillage triangulaire régulier, il est également possible de calculer la normale à la surface. Chaque point du nuage permet de modéliser plusieurs triangles (plans). La normale à ce point est alors estimée comme la moyenne des normales des plans voisins (plans formés par ce même point).

### 2.3.1.3/ A PARTIR DES IMAGES

Généralement, les images fournies par le scanner sont utilisées uniquement pour mettre les points en correspondance et trianguler la position des points 3D correspondants. Cependant, certaines méthodes permettant de calculer la surface à la normale à partir des images sont apparues ces dernières années. Song *et al.* [56] présentent une méthode permettant d'extraire les normales à partir des images en utilisant un système projecteur-caméra à lumière structurée. Le principe de la méthode de Song consiste à étudier la variation des tangentes locales à chaque point. Pour ce faire, le projecteur illumine l'objet avec une grille régulière composée de lignes et de points (les points étant l'intersection des lignes). Les lignes vont alors se déformer en fonction de la forme 3D de l'objet. En étudiant ces déformations, capturées par la caméra, il est alors possible de remonter à la normale à la surface uniquement à partir des images et sans avoir à calculer la position des points 3D.

### 2.3.2/ COURBURE

La courbure est, au même titre que la normale, une information complémentaire très utilisée. Cette information permet de mesurer de manière quantitative si la surface est plus ou moins courbée. Il existe différents types de courbures, cependant les deux types de courbures les plus utilisés sont les courbures gaussienne et moyenne. Le principe de calcul est le même pour ces deux types de courbure.

#### 2.3.2.1/ DÉFINITION DE LA COURBURE

Considérons un plan tournant autour du vecteur normal à un point  $M$  (figure 2.11). Ce plan intersecte la surface en une multitude de courbes  $C$ . Pour chaque courbe est associée sa courbure en  $M$ . Les valeurs minimales et maximales de ces courbes sont appelées courbures principales et sont notées respectivement  $\gamma_{min}$  et  $\gamma_{max}$ . Ce sont ces valeurs qui sont utilisées pour déterminer les courbures gaussienne  $\gamma_g$  et moyenne  $\gamma_m$ , de la façon suivante :

$$\gamma_g = \gamma_{max} \cdot \gamma_{min}, \quad (2.20)$$

$$\gamma_m = \frac{\gamma_{max} + \gamma_{min}}{2}. \quad (2.21)$$

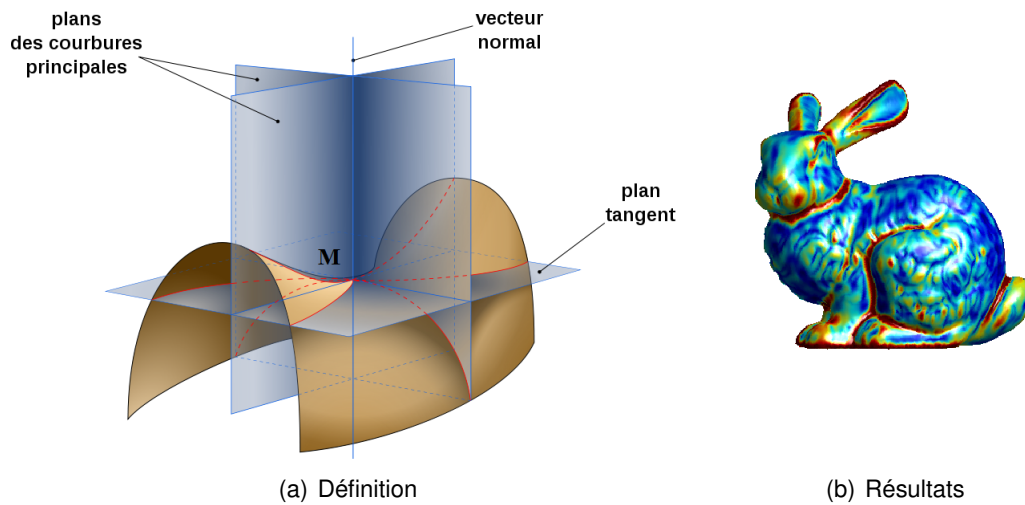


FIGURE 2.11 – Courbure d'un objet 3D. a) Principe de mesure de la courbure. b) Courbure moyenne obtenue sur un objet 3D.

### 2.3.2.2/ MÉTHODES DE CALCUL

Il existe aujourd'hui de nombreuses méthodes permettant d'estimer la courbure à partir d'un maillage triangulaire. Magid [57] et Gatzke [58] présentent un état de l'art des différentes méthodes permettant d'extraire une courbure pour chaque sommet du maillage, ainsi qu'une comparaison de celles-ci. Les résultats qu'ils présentent montrent que selon le type de maillage étudié, ce ne sont pas les mêmes méthodes qui sont les plus efficaces. Les auteurs présentent également une classification des différentes méthodes en trois catégories : les méthodes se basant sur une approximation par une quadrique, les méthodes utilisant les angles entre les triangles contenant le sommet et enfin les méthodes où l'on cherche à calculer un tenseur de courbure.

Douros *et al.* [59] présentent une méthode permettant de calculer les courbures par une approximation de quadrique. Une quadrique est une surface de l'espace euclidien de dimension 3 vérifiant une équation cartésienne de degré 2 :

$$Ax^2 + By^2 + Cz^2 + 2Dyz + 2Exz + 2Fxy + Gx + Hy + Iz + J = 0. \quad (2.22)$$

Le principe de la méthode consiste à estimer une quadrique locale sur le voisinage du point considéré par minimisation. Une fois la quadrique estimée, il est alors possible de déduire les informations de la courbure gaussienne et moyenne de l'équation implicite de la quadrique.



Dong *et al.* [60] présentent, quant à eux, une méthode où la courbure est calculée en étudiant les normales des triangles adjacents à chaque sommet. Pour un point  $P$  du maillage, il est possible de calculer les courbures discrètes grâce à l'équation suivante :

$$k_n(t) = \frac{\langle P_i - P, N_i - N \rangle}{\|P_i - P\|^2}, \quad (2.23)$$

où  $P_i$ , associé à sa normale à la surface  $N_i$  correspond à un point appartenant au voisinage de  $P$ , ayant pour normale  $N$ .  $t$  représente la projection du vecteur  $(P_i, P)$  sur le plan tangent à  $P$ . Il est ainsi possible de calculer une approximation de la courbure dans une direction. Les courbures principales sont ensuite obtenues à partir des courbures approximées pour chaque voisin.

Enfin, Gabriel Taubin [61] propose d'estimer le tenseur de courbure puis d'en extraire les courbures et les directions principales. Les directions principales et les courbures principales sont calculées en calculant les vecteurs et valeurs propres d'une matrice de symétrie, définie par des formules intégrales, qui est très proche de la représentation matricielle du tenseur de courbure.

Il existe de nombreuses autres méthodes dont le principe consiste à estimer la courbure depuis un maillage. Cependant, aucune approche n'est efficace sur tous les types de maillage : chaque méthode est plus ou moins adaptée à un type particulier. Bien que la courbure soit généralement estimée sur le maillage, on retrouve aujourd'hui plusieurs approches différentes permettant de calculer la courbure directement depuis le nuage de points. Parmi ces méthodes, citons entre autre la méthode présentée par Lange *et al.* [62]. Les auteurs estiment des courbures directionnelles à partir du voisinage d'un point, puis se servent de ces informations pour calculer les directions et courbures principales. Berkmann *et al.* [63] proposent également une approche dans laquelle ils calculent la matrice de covariance des vecteurs normaux pour approximer un opérateur de surface et ainsi obtenir la courbure à la surface.

### 2.3.3/ SEGMENTATION 3D

Bien que la segmentation 3D ne soit pas une information complémentaire aux points 3D, contrairement à la normale à la surface et à la courbure, elle n'en reste pas moins une information importante pour de très nombreuses applications, comme l'identification de primitives par exemple. La segmentation consiste à identifier et séparer des parties qui possèdent des caractéristiques communes, en utilisant un critère de segmentation

comme la courbure ou les normales à la surface par exemple. Bien que la segmentation de maillage soit une thématique assez récente, il existe déjà énormément de méthodes de segmentation, dont les plus connues sont les méthodes de segmentation en patches surfaciques. La segmentation en patches surfaciques consiste à créer des petites régions, appelés patches, qui peuvent obéir à certaines propriétés géométriques telles que la planéité, la convexité, la taille ou bien qui sont délimitées par des frontières de forte courbure.

#### 2.3.3.1/ CROISSANCE DE RÉGION

Les méthodes par croissance de région sont les méthodes les plus intuitives pour segmenter un maillage ou un nuage de points. Le principe est le même que dans le cas d'une croissance de région 2D : un point est pris aléatoirement pour initialiser la zone, puis tous les éléments voisins compatibles (qui répondent à un critère fixé) sont ajoutés à cette zone. Le critère utilisé pour stopper la croissance varie selon les différents algorithmes utilisés. Par exemple, Zhang *et al.* [64] utilisent la courbure gaussienne comme critère d'arrêt.

#### 2.3.3.2/ LIGNE DE PARTAGE DES EAUX

Une autre méthode, bien connue en traitement d'image, est la ligne de partage des eaux. Le maillage est considéré comme un relief où chaque point est positionné à une certaine hauteur. Ce relief est ensuite plongé progressivement dans de « l'eau » et des inondations sont simulées au niveau des minima formant ainsi des bassins d'eau. Lorsque deux bassins se rencontrent, une ligne de partage des eaux est créée. Mangan *et al.* [65] proposent deux stratégies pour segmenter un maillage en utilisant la ligne de partage des eaux, une approche ascendante et une approche descendante. L'approche ascendante consiste à inonder les minima jusqu'à ce que les bassins voisins se rencontrent alors que l'approche descendante consiste à faire ruisseler une goutte d'eau le long de la pente la plus forte pour atteindre un minimum. Le nombre de régions obtenues par les deux méthodes précédentes dépend fortement du nombre de graines pour la croissance de région et des minima pour la ligne de partage des eaux. En cas de sur-segmentation, il est toujours possible d'utiliser des procédés de fusion afin de regrouper plusieurs régions.

### 2.3.3.3/ PARTITIONNEMENT DE GRAPHES

La segmentation de maillage s'approche des méthodes de partitionnement de graphes planaires. Il existe de nombreux outils de partitionnement de graphe, dont l'analyse spectrale, et leur application à des maillages permet d'obtenir des résultats intéressants. Liu *et al.* [66] présentent une approche dans laquelle ils segmentent un maillage 3D à partir d'un partitionnement de l'espace spectral.

### 2.3.3.4/ SPHÈRE GAUSSIENNE

Yu *et al.* [67] proposent une méthode qui repose sur l'étude de la sphère de Gauss, qui est une représentation particulière basée sur les normales à la surface, supposées unitaires. La représentation de la sphère de Gauss consiste à placer les normales à la surface au centre d'une sphère de rayon unité. L'étude de la trace laissée par les normales sur la sphère de Gauss permet de déduire la nature de la surface (figure 2.12). Pour segmenter les données à partir de la sphère de Gauss, les auteurs proposent d'isoler les différentes traces présentes sur la sphère à l'aide d'un algorithme de clustering de type Mean-Shift. Il existe bien d'autres méthodes permettant de segmenter un maillage. Delest *et al.* [68] proposent un état de l'art assez fourni sur les différentes méthodes de segmentation.

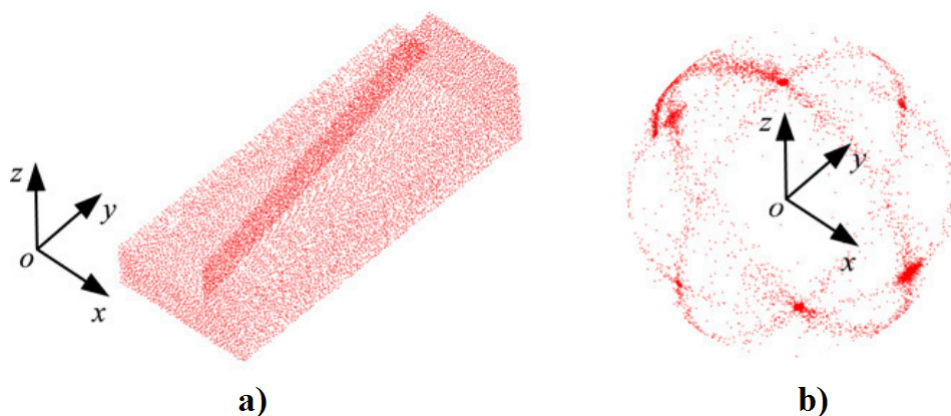


FIGURE 2.12 – Image gaussienne (b) du nuage de points présenté en (a). Extrait de [67].

### 2.3.4/ SYNTHÈSE

A cause de la limitation physique des scanners commerciaux (résolution des caméras principalement) et afin d'obtenir un modèle 3D toujours plus proche de l'original, les méthodes d'extraction d'informations complémentaires n'ont pas cessé de croître ces dernières années. Les différentes méthodes présentées dans cette partie ne sont que

quelques unes de la liste exhaustive des principales méthodes d'extraction d'informations. Il existe encore d'autres attributs qui peuvent être extraits sur le nuage de points ou le maillage, mais ils sont nettement moins utilisés que la normale et la courbure et c'est pourquoi il n'en est pas fait mention ici.

## 2.4/ SIMPLIFICATION DES DONNÉES

Comme cela a déjà été indiqué dans l'introduction, les technologies ne cessent d'évoluer et permettent d'obtenir de plus en plus de données avec une précision toujours plus grande. Si à première vue une telle quantité d'informations peut sembler être une bonne chose, dans certaines applications industrielles, elle s'avère redondante et augmente considérablement le temps de traitement et d'analyse des données. C'est pourquoi, dans la chaîne classique, il y a très souvent une phase de simplification qui permet de réduire la taille des données. Malgré tout, afin de conserver suffisamment d'informations sur l'objet, certaines méthodes de simplification vont chercher à préserver les arêtes et les points caractéristiques.

### 2.4.1/ SIMPLIFICATION D'UN NUAGE DE POINTS

Bien que la simplification des données tridimensionnelles se fasse généralement sur un maillage, on trouve aussi de nombreuses approches permettant de le faire directement depuis le nuage de points. Simplifier les données directement depuis le nuage de points permet d'éviter une étape de maillage voir de remaillage et permet ainsi d'accélérer le processus de la chaîne. On trouve de nombreux travaux dans l'état de l'art permettant de simplifier un nuage de points. Ces différentes approches peuvent être classées en trois catégories : les méthodes de clustering, les méthodes coarse-to-fine et les méthodes itératives.

#### 2.4.1.1/ MÉTHODES DE CLUSTERING

Les méthodes de clustering consistent à subdiviser le nuage de points en plusieurs patches surfaciques (sous-ensemble de points) pour ensuite les remplacer par leur point représentatif, comme le centroïde du patch par exemple. Pauly *et al.* [69] proposent deux stratégies différentes pour la construction des patches surfaciques. La première stratégie utilise un algorithme de croissance de région. La méthode est classique, à un point  $P$  est associé un cluster  $C_0$  qui va croître jusqu'à atteindre une certaine taille, ou que le critère d'arrêt soit atteint. La deuxième stratégie de construction des patches utilise un

partitionnement binaire de l'espace. Le nombre et la taille des patchs sont contrôlés en utilisant l'attribut de courbure. Les méthodes de clustering sont généralement rapides mais produisent des nuages de points avec une erreur quadratique importante.

#### 2.4.1.2/ MÉTHODES COARSE-TO-FINE

Les méthodes coarse-to-fine permettent d'extraire aléatoirement un sous-ensemble de points du nuage original, puis de faire appel à un diagramme de Voronoi 3D pour définir une fonction de distance. Par la suite, cette fonction de distance va être utilisée pour raffiner le nuage de points jusqu'à atteindre l'erreur tolérée requise par l'utilisateur. Moening *et al.* [70] présentent une approche utilisant ce principe. Dans cette approche, les auteurs raffinent le sous-ensemble de points en ajoutant les points qui se trouvent à une certaine distance définie par une fonction géodésique basée sur le diagramme de Voronoi calculé sur le nuage original. Cette méthode permet de contrôler la densité du nuage de points final de telle sorte que les zones planes soient caractérisées par une densité de points plus faible que dans les zones de fortes courbures. Malgré tout, les résultats de ces approches sont fortement dépendants de la distribution du sous-ensemble, choisi aléatoirement.

#### 2.4.1.3/ MÉTHODES ITÉRATIVES

Enfin, les algorithmes qui composent la troisième catégorie, les méthodes itératives, cherchent à réduire de manière itérative le nombre de points du nuage, à l'aide d'un opérateur de décimation. A chaque point est associé un poids qui quantifie son importance dans son voisinage. Ces algorithmes recherchent le point ayant le poids le plus faible dans un voisinage afin de le supprimer. Le voisinage est alors remis à jour ainsi que le poids attribué à chaque point. Song *et al.* [71] proposent une méthode faisant appel à ce principe, où l'importance d'un point est évaluée en vérifiant si ses voisins directs peuvent le représenter. Dans le cas positif, le point est jugé non pertinent et se voit alors supprimé. L'intérêt des méthodes itératives, contrairement aux méthodes précédentes, réside dans la conservation des points vifs et des arêtes. Cependant, l'inconvénient majeur de ces algorithmes est qu'ils sont coûteux en mémoire et en temps de calcul.

### 2.4.2/ SIMPLIFICATION D'UN MAILLAGE

Aujourd'hui, les maillages triangulaires sont la représentation surfacique la plus simple et la plus utilisée par les différents logiciels de visualisation 3D. Certains maillages, qui peuvent être composés de milliards de sommets, sont très lourds à visualiser et ne sont pas forcément nécessaires. De ce fait, la recherche pour simplifier les maillages triangulaires s'est intensifiée ces dernières années afin de réduire le nombre de sommets tout en essayant de conserver le maximum d'informations pertinentes comme les arêtes saillantes. Il existe de très nombreuses techniques permettant de simplifier un maillage triangulaire, voici les plus connues d'entre-elles.

#### 2.4.2.1/ REGROUPEMENT DE SOMMETS

Les techniques de regroupements de sommets, présentées par Rossignac *et al.* [72], cherchent à regrouper les sommets en clusters, pour ensuite les remplacer par un unique représentant. Pour ce faire, dans un premier temps, chaque sommet du maillage se voit attribué un poids selon un critère prédéfini (courbure, taille des faces, angle entre les triangles, ...) afin de donner plus d'importance aux points situés sur des zones courbes, qui seront donc conservés plus longtemps que les autres sommets. L'étape suivante consiste à partitionner le maillage en plusieurs cellules. Par la suite, tous les points appartenant à une cellule sont regroupés au sommet ayant le plus grand poids dans cette même cellule. Low *et al.* [73] présentent des améliorations permettant d'adapter localement la taille des cellules et ainsi obtenir de meilleurs résultats. Cette technique a l'avantage d'être très rapide et permet de fortement simplifier le maillage. Néanmoins, la topologie du modèle original n'étant pas préservée, les résultats visuels sont moins satisfaisants que ceux obtenus par d'autres méthodes. En effet, les techniques ne préservant pas la topologie du maillage ont tendance à lisser les arêtes, ce qui affecte la silhouette du modèle.

#### 2.4.2.2/ TECHNIQUE DE DÉCIMATION

Comme leur nom l'indique, les techniques de décimation cherchent à supprimer des sommets dans le maillage, comme le présentent Schroeder *et al.* [74]. Les trous ainsi créés sont remaillés en employant diverses méthodes, ce qui permet de réduire le nombre final de faces tout en préservant au maximum la topologie du modèle. L'algorithme procède en plusieurs passes en choisissant à chaque fois si un sommet peut être supprimé selon deux critères : le sommet sera enlevé si sa suppression préserve la topologie du maillage, et si la re-triangulation du trou qu'il crée donne des faces à une distance du maillage original inférieure à un certain seuil. En géométrie, on utilise souvent la caractéristique d'Euler

pour vérifier si une opération change la topologie de la forme modifiée. La caractéristique d'Euler est un invariant numérique, un nombre qui décrit un aspect d'une forme, qui est communément noté par  $\chi$  et qui dans le cas de maillage peut se mettre sous la forme suivante :

$$\chi = s - a + t, \quad (2.24)$$

avec  $s$  le nombre de sommets,  $a$  le nombre d'arêtes associés aux sommets, et  $t$  le nombre de triangles. Si la caractéristique d'Euler ne change pas, on considère que l'opération ne modifie pas la topologie du maillage. Il existe trois grandes méthodes pour simplifier un maillage par décimation : la suppression de sommets, la contraction d'arêtes et la contraction partielle d'arêtes (figure 2.13).

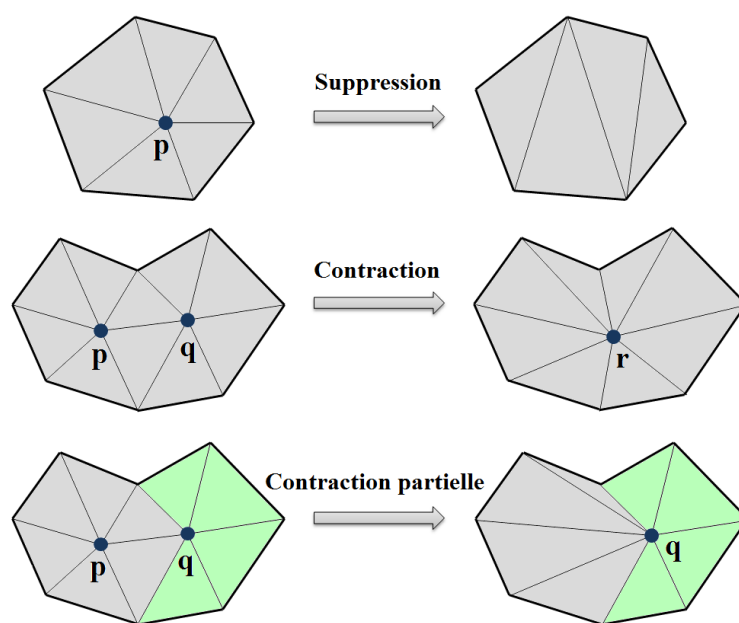


FIGURE 2.13 – Les trois différentes méthodes de décimation.

La première approche, la suppression de sommets, consiste à supprimer un point et à re-trianguler le trou. S'il y a  $k$  triangles associés à un point, il n'en restera plus que  $k - 2$  après cette opération (figure 2.13). Le nombre d'arêtes sera quant à lui de  $k - 3$  et le nombre de sommets sera réduit de 1. On constate alors qu'avec cette méthode, la caractéristique d'Euler est bien invariante. La seconde approche, la contraction d'arête ou « *edge collapse* » en anglais, consiste à prendre deux sommets voisins  $p$  et  $q$  et à contracter les arêtes communes en un seul et nouveau sommet  $r$ . Il s'agit également d'une opération d'Euler. La dernière approche (*half-edge collapse* en anglais) est un dérivé de la méthode précédente où l'on va déplacer le sommet  $p$  sur le sommet  $q$ . De plus, la qualité des modèles simplifiés peut être améliorée en déplaçant les sommets

créés pour qu'ils améliorent l'aspect visuel final, comme le présentent Garland *et al.* [75]. Néanmoins, d'une part le fait de préserver la topologie du maillage empêche une très grande simplification et d'autre part, ces algorithmes sont très coûteux en charge de calcul.

#### 2.4.2.3/ SURFACES ENVELOPPANTES

La simplification par surfaces enveloppantes [76] est une approche dérivée des méthodes de décimation. Le principe de cette méthode consiste à définir deux surfaces qui vont envelopper le maillage, puis de simplifier celui-ci à l'aide d'une méthode de décimation en forçant la déviation géométrique induite par la simplification dans l'espace fixé par les deux enveloppes. Les deux surfaces, l'une à l'intérieur et l'autre à l'extérieur de la surface du modèle, sont créées en déplaçant la surface originelle d'un petit intervalle  $\varepsilon$  le long des normales des sommets :  $+\varepsilon$  pour la surface externe et  $-\varepsilon$  pour la surface interne. Le processus de simplification est le même que la décimation de maillage : des sommets sont supprimés, mais les choix des sommets sont faits de façon à ce que le maillage résultant reste toujours compris entre les deux enveloppes. Cette méthode a l'avantage de conserver la géométrie du maillage d'origine. Néanmoins, cette approche est très coûteuse en temps de calcul et n'est pas forcément robuste pour tous les types de maillages.

#### 2.4.2.4/ MAILLAGE PROGRESSIF

Hoppe *et al.* [77] présentent une technique originale permettant de simplifier un maillage en minimisant une fonction d'énergie. Le principe de cette approche consiste à définir une fonction d'énergie qui permet de mesurer la qualité de chaque réduction du maillage. A chaque itération, le maillage va être réduit par une contraction d'arête (technique de décimation). L'arête qui sera supprimée sera celle qui induit la plus grande erreur sur la fonction d'énergie. Cette méthode a l'avantage de conserver la topologie de la surface et donne de bons résultats (figure 2.14) mais c'est une méthode très lourde en temps de calcul.

#### 2.4.2.5/ AUTRES MÉTHODES

Il existent encore d'autres méthodes permettant de simplifier un maillage, comme les méthodes basées sur une décomposition en ondelettes. Néanmoins, pour obtenir de bons résultats avec une méthode utilisant des ondelettes, le maillage doit être régulier



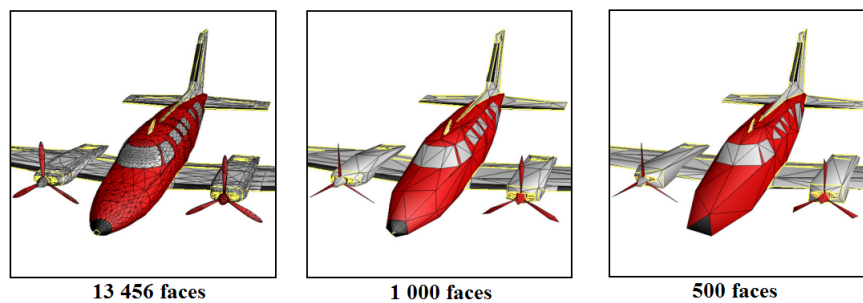


FIGURE 2.14 – *Simplification de maillage par la méthode présentée par Hoppe [77].*

ce qui implique d'ajouter une étape supplémentaire de re-maillage à la chaîne 3D. Certains travaux sont apparus ces dernières années où les auteurs présentent une méthode basée ondelette pour simplifier des maillages irréguliers [78]. Cependant, la plupart des travaux sur la simplification de maillage basée ondelette continue de se faire à partir de maillage régulier ou semi-régulier. Dans une autre approche, basée sur un échantillonnage en disque de poisson, les auteurs cherchent à préserver au maximum les arêtes saillantes tout au long du processus [79]. Pour plus d'informations, Cignoni *et al.* [80] présentent une étude comparative des différentes méthodes existantes permettant de simplifier un maillage 3D.

### 2.4.3/ PRIMITIVES 3D

Bien que l'extraction des primitives ne soit pas directement une méthode de simplification, elle peut s'en rapprocher. En effet, la plupart des objets industriels peuvent être modélisés par des primitives géométriques (plan, sphère, cylindre, cône), ce qui permet de remplacer des données par une forme géométrique simple permettant ainsi de réduire considérablement la taille des données. Ces formes géométriques sont très utiles lorsque l'on souhaite obtenir un modèle CAO d'une pièce manufacturée, mais elles servent également à bien d'autres applications comme la segmentation par exemple. Les primitives peuvent également être utilisées comme une information supplémentaire aux points 3D. Ces dernières années, de plus en plus de méthodes sont apparues permettant d'extraire des primitives à partir d'un nuage de points ou d'un maillage. Si certains algorithmes vont extraire des primitives à partir du nuage de points complet, la plupart des approches préfèrent d'abord passer par une étape de segmentation préalable.

### 2.4.3.1/ RANSAC

L'algorithme du Ransac (RANDOM SAmple Consensus), initialement proposé par Fischler [81], est une méthode stochastique permettant de calculer un modèle correspondant aux données observées. Le plus gros intérêt de cette approche réside dans sa robustesse aux points aberrants, qui sont le plus gros problème des méthodes de modélisation au sens des moindres carrés. Le Ransac permet d'identifier un modèle correspondant aux seules données pertinentes, sans tenir compte des données aberrantes, sous l'hypothèse que les données pertinentes soient suffisamment nombreuses. Généralement, ce type d'algorithme est utilisé sans segmentation préalable, dans les cas où l'objectif consiste à chercher un modèle dans l'ensemble du nuage de points.

On suppose qu'un modèle peut être construit à partir d'un ensemble de plusieurs données. Par exemple, Beder *et al.* [82] expliquent comment construire un cylindre à partir de 5 points, bien qu'il soit plus simple de le construire à partir de 6 ou 7 points. Le Ransac est un algorithme itératif qui va, à chaque itération, construire un candidat de  $k$  données choisies aléatoirement parmi l'ensemble des données. La pertinence de chaque candidat est évaluée à l'aide d'un score comptabilisant le nombre de données décrivant correctement le modèle parmi la totalité des données. Généralement, le critère permettant de définir si les données sont « correctes » ou non est la distance euclidienne entre un point et la forme du candidat. Si cette distance est inférieure à un seuil  $\varepsilon$  prédéfini, alors le point est considéré comme un point décrivant le modèle. A l'issue des  $N$  itérations de l'algorithme, seul le candidat ayant obtenu le meilleur score est conservé. Si le score maximal est supérieur à un seuil  $\tau$  prédéfini, le candidat associé à ce score est validé, sinon le candidat est rejeté et on considère que la modélisation a échoué. Généralement, lors des étapes de reconnaissance de primitives, on ne recherche pas qu'une seule primitive. De ce fait, l'algorithme sera amené à être répété plusieurs fois, jusqu'à ce qu'il échoue (seuil  $\tau$  non atteint).

Le paramètre le plus important lorsque l'on cherche à identifier des primitives est le nombre d'itérations  $N$ . Du fait que le procédé est stochastique, plus  $N$  sera grand et plus le test du Ransac sera robuste pour identifier la meilleure primitive. A titre d'exemple, Schnabel *et al.* [83] montrent qu'il faut plus de 30 milliards de tentatives pour garantir à plus de 95% la détection d'un cylindre (créé avec  $k = 5$  points) échantillonné par 10000 points au sein d'un nuage composé d'un million de points. Le nombre de points  $k$ , utilisés pour modéliser la primitive, joue également un rôle essentiel car plus il sera grand et plus il sera improbable de sélectionner les points sur une même primitive. C'est pour ces

raisons, qu'en règle générale les algorithmes Ransac sont plutôt utilisés pour identifier des plans, qui ne sont construits qu'avec 3 points ( $k = 3$ ). Bretar [84] propose d'ailleurs un tirage des points basé sur une similarité des normales à la surface pour permettre la génération de candidats plus pertinents.

#### 2.4.3.2/ TRANSFORMÉE DE HOUGH

Paul Hough [85] propose une méthode permettant la reconnaissance de lignes parmi des données 2D. Considérons qu'une droite est paramétrée par son angle vis-à-vis de l'axe des abscisses, l'ensemble des droites passant par un point correspond à une sinusoïde dans l'espace paramétrique des droites. Ainsi, un point de l'espace paramétrique où se rencontrent plusieurs sinusoïdes représente une droite passant par plusieurs points (figure 2.15).

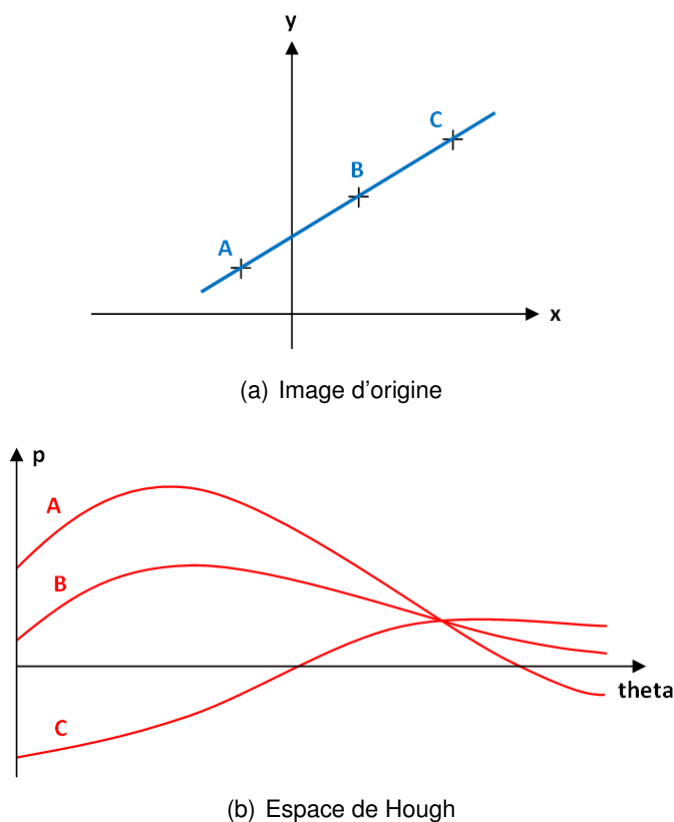


FIGURE 2.15 – Principe de la transformée de Hough. A chaque point  $A, B$  et  $C$  correspond une sinusoïde dans l'espace de Hough. Si les sinusoïdes se croisent en un point, il existe alors une droite qui passe par les trois points  $A, B$  et  $C$  dont l'orientation correspond à l'intersection des sinusoïdes dans l'espace de Hough.

Cette approche peut être généralisée pour permettre la détection de formes plus complexes dans les images et dans les données tridimensionnelles. Cependant, afin d'identifier une forme plus complexe, il faudra explorer l'espace paramétrique autant de

fois qu'il y a de paramètres utilisés pour définir la primitive. A titre d'exemple, dans le cas d'un cylindre ( $k = 5$ ) la recherche de celui-ci dans un nuage d'un million de points nécessite  $10^{14}$  opérations pour le calcul de la transformée de Hough et  $10^{10}$  pour détecter les intersections des sinusoides. C'est pourquoi, tout comme l'algorithme Ransac, la transformée de Hough est plutôt utilisée pour identifier les plans présents dans les données 3D.

Tahir Rabbani [86] présente des travaux particulièrement intéressants dans lesquels il propose une méthode permettant de détecter les cylindres dans un nuage de point en séparant le traitement des paramètres définissant le cylindre. Au lieu de calculer une transformée en 5 dimensions, les auteurs procèdent en deux temps en calculant une transformée en 2 dimensions puis en 3 dimensions. La première étape consiste à estimer l'axe du cylindre (2 paramètres) en utilisant la transformée de Hough sur la sphère gaussienne. La deuxième partie consiste à calculer le rayon et la position du cylindre (3 paramètres) en projetant les points du nuage dont la normale est approximativement orthogonale à l'axe, sur un plan perpendiculaire à ce même l'axe. Les points ainsi projetés décrivent alors un cercle, dont il est possible d'extraire la position et le rayon du cylindre.

#### 2.4.3.3/ MÉTHODE DE BENKŐ *et al.*

Les différentes méthodes que nous venons de voir ne nécessitent pas de segmentation préalable pour extraire des primitives. De ce fait, ces méthodes peuvent être particulièrement coûteuses en temps de calcul, à l'image de l'algorithme du Ransac. Benkő *et al.* [87] présentent une méthode permettant d'extraire des primitives à partir d'un maillage en vue d'une modélisation CAO par la suite. Pour ce faire, les auteurs ont choisi de segmenter les différents objets qui composent la scène, puis de les soumettre chacun leur tour à des tests afin de déterminer par quel type de primitives ceux-ci peuvent être modélisés. Ainsi, la première étape de la méthode de Benkő consiste à segmenter les différents objets en supprimant les triangles qui composent les arêtes saillantes et les congés (détectés à l'aide de l'attribut de la courbure). Les auteurs proposent ensuite de soumettre chaque objet segmenté à une série de tests successifs pour déterminer à quel type de primitive ils appartiennent. Voici les grandes lignes de l'algorithme :

- Test de planéité (plan, sphère).
- Test sur la direction de translation (cylindre, forme libre).
- Test s'il s'agit d'un cône (cône).
- Test sur l'axe de rotation (tore, forme libre).

Les auteurs donnent de plus amples détails sur les différents tests mis en place dans [87]. Le problème de cette approche est le nombre de tests effectués pour chaque objet, et de ce fait le nombre de seuils différents qui interviennent à chaque fois. Une fois les primitives identifiées, les auteurs construisent la topologie des objets en utilisant un graphe d'adjacence et d'intersections pour obtenir un modèle CAO, qu'ils complètent avec des congés entre les différentes formes en se basant sur la topologie. Cependant, cette approche effectue de nombreux tests, et de ce fait, utilise un nombre important de seuils, ce qui peut s'avérer contraignant dans certains cas.

#### 2.4.3.4/ MÉTHODE DE SUNIL *et al.*

La méthode présentée par Sunil *et al.* [88] est assez similaire à celle de Benkő [87] : l'idée est de segmenter les différents objets qui composent la scène puis d'attribuer un type de surface à chaque objet en vue d'une modélisation CAO. La différence entre les deux méthodes se fait tout d'abord au niveau de la segmentation. Sunil propose dans un premier temps de nettoyer le maillage pour obtenir un maillage uni-manifold (c'est à dire un maillage respectant certaines propriétés). La segmentation des différents objets se fait ensuite en fonction des arêtes spécifiques (déterminées grâce au fait que le maillage soit uni-manifold).

L'idée présentée par les auteurs [88] consiste à définir différents tests pour chaque objet afin d'identifier la primitive pouvant modéliser celui-ci. Dans le cas du plan par exemple, les normales à la surface sont toutes orientées dans la même direction. De ce fait, les auteurs proposent de déterminer les plans en utilisant une croissance de région avec comme critère d'arrêt l'angle entre deux normales voisines (tant que l'écart angulaire respecte une certaine tolérance, la zone continue de croître). La modélisation du cylindre se fait ici de la même façon que pour l'approche de Yu *et al.* [67], qui est basée sur l'utilisation de la sphère gaussienne. Les autres surfaces sont quant à elles identifiées à partir de la structure et la connectivité du maillage. L'identification du type de surface suit le principe suivant : en CAO, une surface peut être représentée par une densité de triangles plus ou moins faible en fonction de la nature de la surface. Dans un maillage uni-manifold, la structure et la connectivité de chaque surface sont particulières et peuvent être utilisées pour différencier le type de surface. L'inconvénient de cette approche réside dans la difficulté d'obtenir un maillage manifold, notamment sur des pièces industrielles où le maillage aura fréquemment des parties « ouvertes ».

## 2.4.3.5/ MÉTHODE DE BÉNIÈRE

Plus récemment, Roseline Bénière [89] a proposé une approche similaire à celles de Benkō et Sunil pour extraire des primitives. Le principe peut se résumer en trois points :

- Définir la forme locale de chaque sommet en utilisant l'attribut de courbure.
- Extraire des zones (plusieurs sommets) correspondant à un même type de primitive.
- Estimer les paramètres des primitives pour les modéliser.

Dans ces travaux, les auteurs considèrent que les primitives géométriques peuvent être caractérisées par la courbure. Comme l'illustre la figure 2.16, chaque type de primitive possède des propriétés spécifiques que les auteurs utilisent pour caractériser chaque sommet et identifier leur appartenance à un type de primitive.

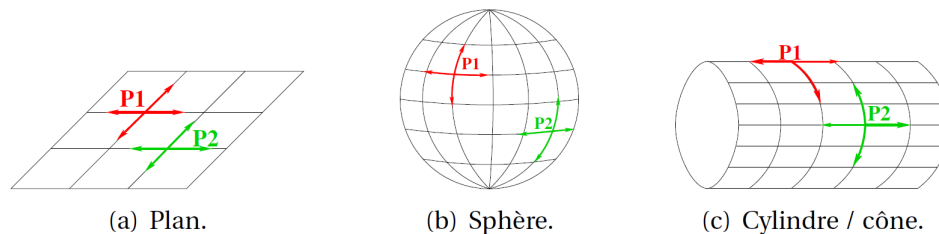


FIGURE 2.16 – *Propriétés de courbure de différentes primitives simples. Les flèches correspondent aux directions principales. Extrait de [89].*

L'étape suivante consiste à extraire des zones de points ayant des caractéristiques communes en s'appuyant sur l'étape précédente. En étudiant le voisinage de chaque point, il est possible de faire croître une zone si les primitives associées aux sommets voisins sont similaires. Le critère d'appartenance à la zone est différent pour chaque type de primitive. Enfin, une fois les zones de points extraites, il ne reste plus qu'à modéliser la primitive. Pour cela, Bénière utilise deux approches : par moyenne des valeurs de courbure et par régression sur le nuage de points. Cependant, comme mentionné par les auteurs, la discrétisation influence beaucoup la qualité des résultats. Si l'on veut obtenir de meilleurs résultats, ce genre d'approche nécessite une étape de re-maillage au préalable.

## 2.4.3.6/ SYNTHÈSE

Les différentes méthodes présentées ci-dessus permettent toute d'extraire des primitives, soit directement depuis le nuage de points, soit depuis le maillage. Les techniques basées sur le nuage de points ont tendance à être assez coûteuses en temps de calcul, ce qui les rend assez contraignantes au niveau de leur utilisation. Récemment,

Yu [67] a présenté une méthode originale permettant d'obtenir une segmentation et une identification des primitives directement depuis le nuage de points à partir de la sphère gaussienne. Cependant, cette approche ne permet d'identifier que les plans, les cylindres, les formes ellipsoïdales et les formes coniques. Les méthodes basées sur le maillage, quant à elles, sont plus rapides et plus robustes mais elles nécessitent d'avantage d'informations pour fonctionner correctement (courbure locale, maillage régulier, etc.).

Que ce soit sur le nuage de points, sur le maillage ou par modélisation de primitives, la simplification des données est une thématique qui a été et qui est toujours très étudiée. Ceci met bien en avant le besoin de réduire la taille des données tout en préservant le maximum d'informations. Ce qui amène la question que l'on se pose dans ce document : pourquoi chercher à calculer un nuage très dense lors de l'acquisition si les données sont destinées à être simplifiées par la suite ?

## EXTRACTION D'INFORMATIONS DEPUIS LES IMAGES

Dans ce mémoire nous cherchons à développer un système de numérisation dynamique permettant d'adapter l'acquisition à la forme de l'objet, en scannant grossièrement la pièce à numériser puis en raffinant les zones jugées importantes par la suite (on appelle ce type de numérisation : « *coarse to fine* »). L'intérêt de cette approche est de simplifier les données finales tout en évitant de passer par la chaîne 3D classique (figure 3.1), que nous venons de décrire étape par étape dans la partie précédente. Ce travail peut se décomposer en deux parties : la première partie consiste à extraire un maximum d'informations complémentaires aux points 3D à partir des images fournies par le scanner, et la seconde partie consiste à utiliser ces informations pour juger quelles sont les parties pertinentes et ainsi raffiner l'acquisition en utilisant les images du scanner 3D.

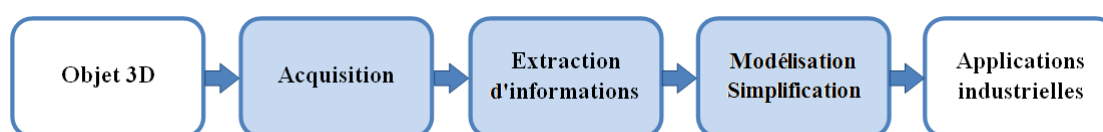


FIGURE 3.1 – *La chaîne 3D classique : de l'acquisition à la modélisation.*

Dans ce chapitre, nous traitons la première partie de ce travail, à savoir l'extraction d'informations complémentaires à partir des images fournies par le scanner 3D de la société Noomeo. Ce scanner (figure 3.2-a) fonctionne sur le principe d'un système de stéréovision, comme présenté dans l'état de l'art, en s'aidant d'une projection de lumière structurée (figure 3.2-b) pour appairer les points entre les deux images. Les caractéristiques techniques du scanner sont les suivantes :



- Nombre de points : 10.000 à 500.000 par acquisition.
- Champ de vue :  $150 \times 165\text{mm}$ .
- Résolution spatiale :  $300\mu\text{m}$
- Précision :  $\pm 100\mu\text{m}$ .
- Taille de l'objet : de  $10\text{cm}^3$  à  $1,2\text{m}^3$ .
- Résolution des caméras :  $1024 \times 728$ .

Si l'extraction d'informations tridimensionnelles a déjà été très étudiée sur le nuage de points et les maillages, elle l'est beaucoup moins sur les images. L'objectif du système étant d'adapter l'acquisition à l'objet, tout doit se faire en utilisant au maximum les informations du scanner (images, paramètres de calibration, ...) et donc en utilisant au minimum les informations 3D.

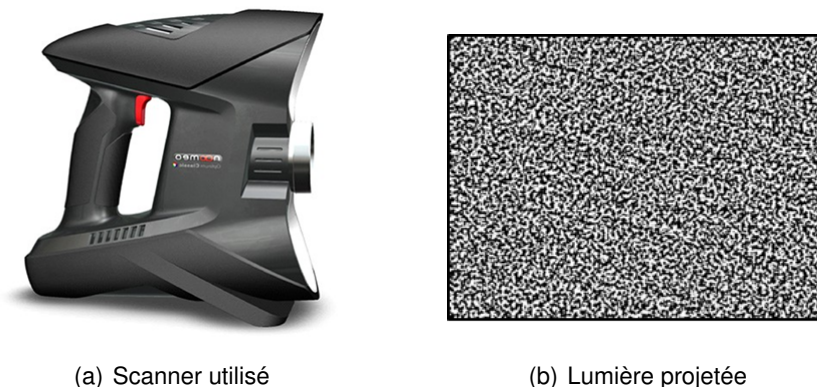


FIGURE 3.2 – Scanner de la société Noomeo.

La question qui se pose est donc la suivante : quelles informations peut-on extraire à partir des images et des données de calibration fournies par le scanner ? Nous avons vu qu'il existe déjà des méthodes pour calculer les normales à la surface à partir des images et des paramètres de calibration. Nous avons également vu que la segmentation et l'extraction de primitives était une étape importante dans la grande majorité des méthodes de simplification. Dans notre cas nous ne cherchons pas à simplifier les données, mais plutôt à adapter l'acquisition par un procédé de « *coarse to fine* ». Pour cela, il est possible de calculer des informations complémentaires aux points 3D depuis les images : les normales 3D à la surface, les discontinuités et les primitives géométriques (plans, sphère, cylindre et cône).

### 3.1/ CALCUL DES NORMALES

Comme mentionné dans l'état de l'art, il existe déjà des approches permettant de calculer les normales à la surface à partir d'images. La méthode la plus récente, et qui semble donner les meilleurs résultats, est la méthode proposée par Song *et al.* [56] qui utilise un système projecteur-caméra avec projection de lumière structurée.

#### 3.1.1/ ADAPTATION DE LA MÉTHODE DE SONG

La méthode [56] présentée par les auteurs permet d'obtenir la normale à partir de la projection d'une grille régulière, qui n'est autre que le motif de lumière structurée, sur l'objet à numériser. L'objectif de cette partie consiste à adapter la méthode présentée par Song à notre système de numérisation.

La figure 3.3 illustre le principe de cette approche. Prenons un point de la grille du projecteur  $A_p$ . La projection de ce point sur la surface d'un objet 3D forme un point  $A$ . La capture du point  $A$  par la caméra nous donne un point  $A_c$  dans l'image caméra. Notons également les tangentes aux lignes de la grille au point  $A_p$  ;  $t_{px}$  et  $t_{py}$ , et les tangentes correspondant aux lignes de la grille au point  $A_c$  (caméra) ;  $t_{cx}$  et  $t_{cy}$ . Notons également le plan d'éclairage du projecteur  $\Pi(A_p, t_{px})$  formé par la tangente  $t_{px}$  et le point  $A_p$  (ainsi que le centre de projection du projecteur  $O_p$ ). Ce plan de lumière est réfléchi par la surface de l'objet au point  $A$  et devient le plan de projection  $\Pi(A_c, t_{cx})$  pour le plan image. L'intersection des deux plans  $\Pi(A_p, t_{px})$  et  $\Pi(A_c, t_{cx})$  définit en réalité une tangente  $t_x$  à la surface de l'objet au point  $A$ .  $A_p$  et  $t_{px}$  sont tout deux connus car ils font partie du motif projeté, de même que  $A_c$  et  $t_{cx}$  qui sont observables à partir de l'image. Connaissant toutes les données, il est alors possible de reconstruire les deux plans  $\Pi(A_p, t_{px})$  et  $\Pi(A_c, t_{cx})$ . On peut alors facilement déterminer leur intersection  $t_x$ .

De la même façon, il est possible de déterminer une autre tangente  $t_y$  au point  $A$  à partir de deux autres plans  $\Pi(A_p, t_{py})$  et  $\Pi(A_c, t_{cy})$  qui sont tous deux connus (à partir du modèle projeté et de son observation dans l'image). En d'autres termes, la méthode proposée par les auteurs de [56], permet de déterminer l'orientation de la surface  $n(x, y)$  de la surface de l'objet associé au point 3D  $A$ , en prenant une image de la surface de l'objet éclairé par un modèle approprié. Les tangentes 3D  $t_x, t_y$  peuvent être déterminées à partir des données extraites de la caméra et du projecteur, et la normale à la surface est tout simplement donnée par la relation suivante :

$$n(x, y) = t_x \times t_y. \quad (3.1)$$

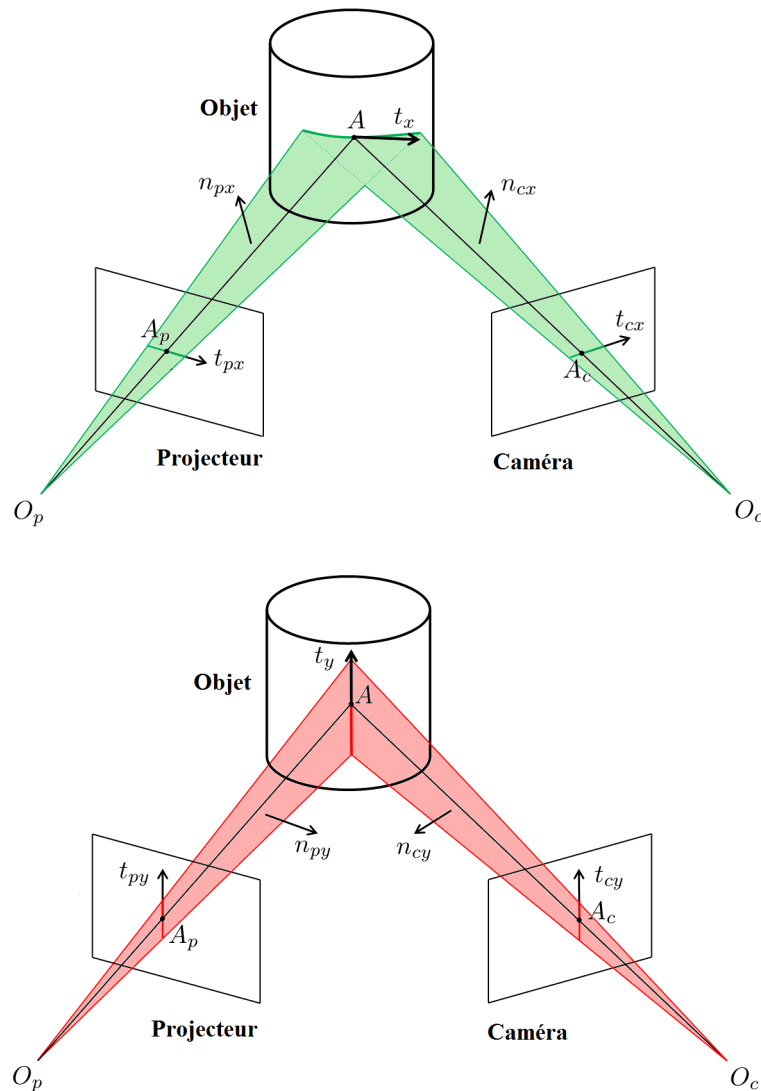


FIGURE 3.3 – La géométrie permettant de déterminer l'orientation de la surface à un point de la grille à partir des lignes de cette grille.

L'intérêt majeur de cette technique repose sur la possibilité d'extraire l'orientation de la surface en utilisant uniquement les images et les paramètres de calibration, ce qui permet de fortement réduire le temps de calcul (les données 2D sont plus simples à traiter que les données 3D et l'information de voisinage est fournie par l'image).

Dans le cas nous concernant, nous n'utilisons pas un système projecteur-caméra mais un système de stéréo-vision. La méthode de mise en correspondance utilisée est une méthode locale par propagation de germe. Ce que nous proposons ici est d'adapter la méthode de Song, prévue à la base pour une grille régulière, au système développé. Nous avons vu dans l'état de l'art sur la mise en correspondance que les méthodes locales par propagation de germes pouvaient suivre deux différentes façons de faire : l'ap-

proche simultanée et l'approche séquentielle. L'approche séquentielle ne nous intéresse pas car elle se base sur les points d'intérêts présents dans l'image. A contrario l'approche simultanée consiste à propager un ensemble de germes. En choisissant ces germes de manière régulière, il est possible de générer une grille de germes et ainsi revenir au cas présenté par Song, en assimilant la caméra de référence du scanner 3D au projecteur utilisé dans sa méthode (figure 3.4).

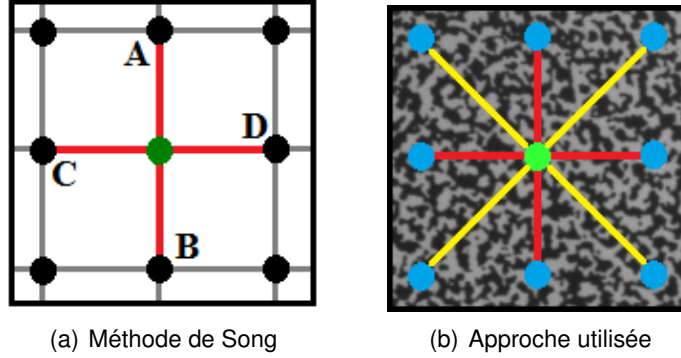


FIGURE 3.4 – *Adaptation de la méthode de Song [56] au scanner utilisé. a) Grille utilisée dans la méthode présentée par Song. b) Grille de germes (en vert) utilisée pour adapter la méthode de Song.*

Dans cette approche, la surface est supposée localement plane entre deux points successifs. Cette supposition permet de calculer les tangentes comme une simple différence finie entre deux points voisins. La figure 3.4 illustre la méthode. Pour un point  $P$  (en vert), les deux tangentes  $t_{px}$  et  $t_{py}$  sont estimées de la façon suivante :

$$t_{px} = \begin{bmatrix} x_A \\ y_A \end{bmatrix} - \begin{bmatrix} x_B \\ y_B \end{bmatrix}, \quad t_{py} = \begin{bmatrix} x_C \\ y_C \end{bmatrix} - \begin{bmatrix} x_D \\ y_D \end{bmatrix}. \quad (3.2)$$

Dans la méthode originale présentée par Song, la normale 3D à chaque point est calculée à partir de deux tangentes (symbolisées en rouge sur la figure 3.4). Dans ce cas précis, le moindre bruit sur les tangentes peut affecter les résultats. Afin d'améliorer la robustesse de cette méthode, il est possible d'utiliser plus de tangentes en prenant en compte un voisinage plus important toujours extrait à partir de la grille, en l'occurrence les diagonales (en jaune sur la figure 3.4-b). La figure 3.5 présente les résultats obtenus avec la méthode de Song adaptée au motif du scanner à notre disposition.

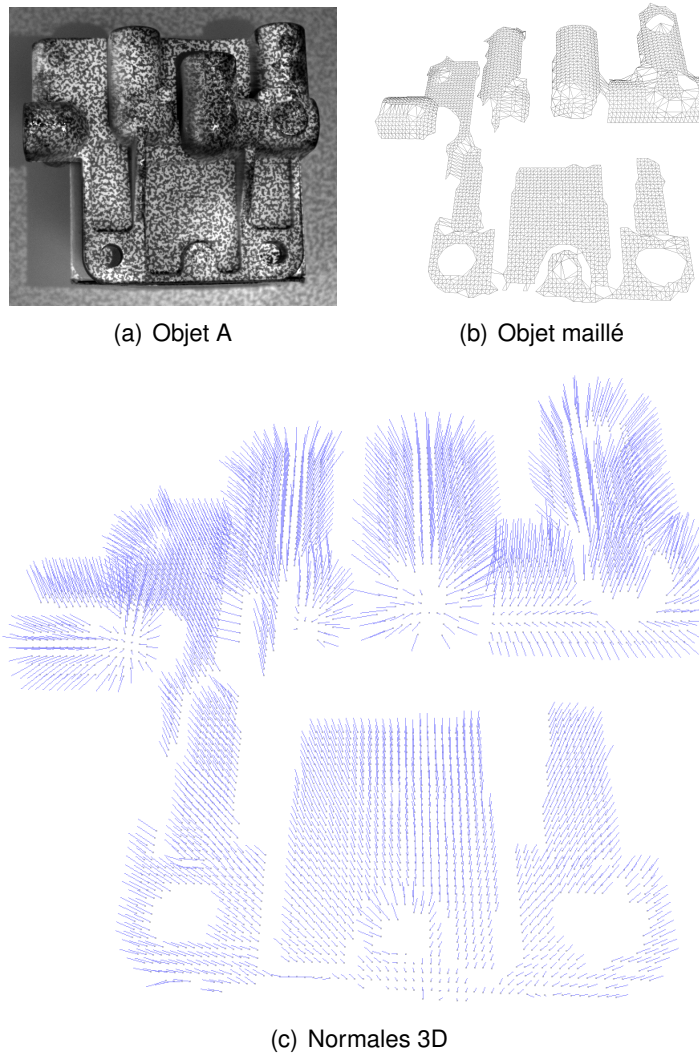


FIGURE 3.5 – Résultats obtenus en adaptant la méthode de Song avec le scanner 3D utilisé.

### 3.1.2/ INFLUENCE DE LA DENSITÉ DE POINTS

Afin de vérifier la qualité des résultats obtenus, nous présentons une comparaison de notre méthode, dérivée de [56], et utilisant deux tangentes puis quatre tangentes pour calculer les normales à la surface. La comparaison des résultats est faite en calculant l'écart angulaire entre les normales de référence (estimées sur le nuage de points avec 30 voisins par la méthode de Hoppe *et al.* [48]) et celles estimées par notre méthode avec deux puis quatre tangentes, pour différentes densités de points sur trois objets bien différents présentés en figure 3.6. Le premier objet est un plan dont la surface est légèrement irrégulière, ce qui est conforme à la supposition de départ qui stipule que la surface est localement plane. Le deuxième objet est composé de nombreuses zones planes ainsi que de plusieurs arêtes saillantes. Enfin, le dernier objet est quant à lui plus

complexe que les autres, la surface varie beaucoup et n'est pas composée de zones planes. Il s'agit de l'objet respectant le moins la supposition sur laquelle repose le principe de la méthode. La figure 3.6 représente l'erreur moyenne (la moyenne des écarts angulaires) entre les deux approches (avec deux puis quatre tangentes) et la méthode de Hoppe, en fonction du nombre de points total utilisé pour représenter l'objet.

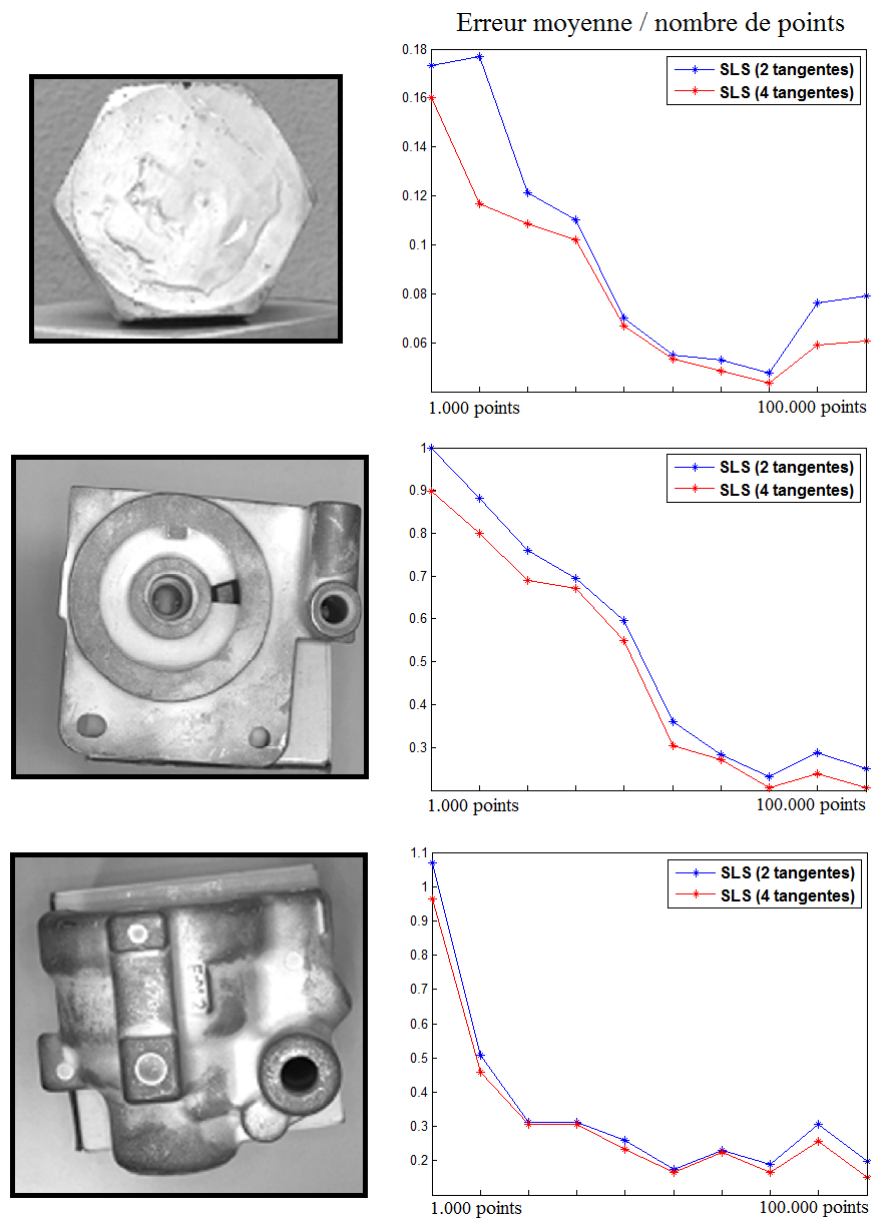


FIGURE 3.6 — Etude de l'influence de la densité de points et du nombre de tangentes utilisées.

Ces courbes montrent que le fait d'utiliser plus de tangentes pour estimer la normale donne globalement de meilleurs résultats quelle que soit le nombre de points. Le gain en terme de précision est plus ou moins important en fonction de l'objet utilisé. On remarque que dans le cas du troisième objet, où la supposition d'une surface localement plane est la

moins vérifiée, le fait d'utiliser plus de tangentes n'apporte qu'une très faible amélioration. Globalement les erreurs sur les normales restent faibles, de  $0^\circ$  à  $1^\circ$  d'écart angulaire entre les normales de référence et celles calculées en adaptant la méthode de Song au système.

### 3.1.3/ CARACTÉRISATION DU BRUIT SUR LES NORMALES

Comme dans tout système de numérisation, il existe du bruit dans les images et les paramètres de calibration sont légèrement incorrects. De ce fait, les normales calculées en adaptant la méthode de Song sont elles aussi bruitées. Dans cette partie, nous avons cherché à estimer le bruit sur les normales, à partir d'un plan étalon, comme l'écart entre les normales calculées avec la méthode décrite précédemment et la normale théorique au plan de référence (figure 3.7).

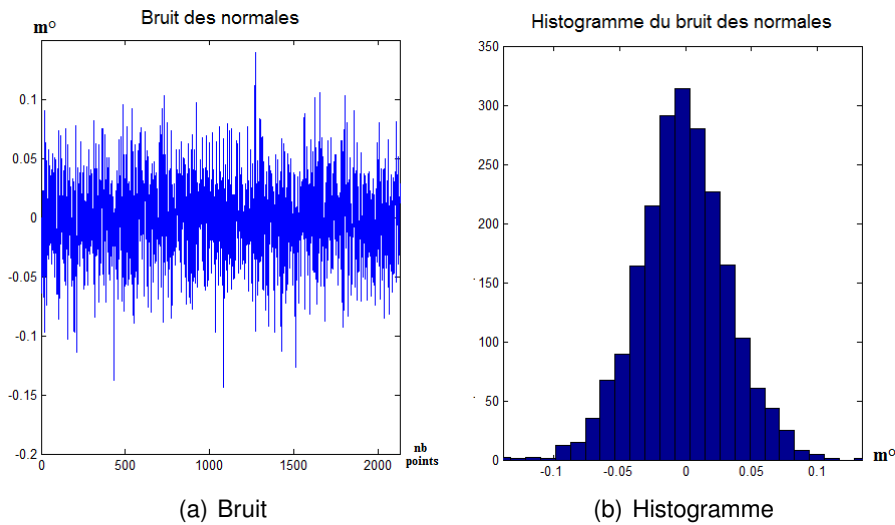


FIGURE 3.7 – *Caractérisation du bruit des normales.*

A partir de l'histogramme (figure 3.7), le bruit semble suivre une loi normale. La loi normale théorique peut-être calculée directement à partir des données simplement en estimant l'écart-type  $\sigma$  et la moyenne  $x_0$  des données. Pour information, la moyenne du bruit  $x_0$  est de  $-1,56 \cdot 10^{-3}^\circ$  et la variance est de  $33,7 \cdot 10^{-3}^\circ$ . La loi normale s'écrit alors :

$$f(x) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{1}{2} \left( \frac{x-x_0}{\sigma} \right)^2}. \quad (3.3)$$

Sur la figure 3.8, on peut distinguer l'histogramme ainsi que la loi normale obtenue à partir des données (courbe rouge). Il semble que le bruit suit bien une loi normale. Afin de le vérifier, nous proposons d'effectuer le test du  $\chi^2$ .

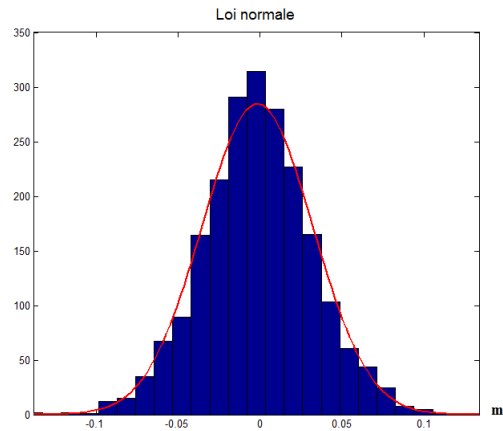


FIGURE 3.8 – Superposition de l'histogramme et de la loi normale estimée à partir de la variance et la moyenne du bruit (courbe rouge).

Le test du  $\chi^2$  est un test statistique permettant de tester l'adéquation entre des données et une loi de probabilité. Pour ce faire, on suppose une hypothèse appelée hypothèse nulle  $H_0$  qui est la suivante dans notre cas : les données suivent une loi de probabilité, ici la loi normale. Le but du test consiste à tester la validité de cette hypothèse. Pour ce faire, on calcule le  $\chi^2$  qui s'écrit :

$$\chi^2 = \sum_{i=1}^J \frac{(Nobs_i - Nth_i)^2}{Nth_i}, \quad (3.4)$$

avec  $J$  le nombre de classes,  $Nobs_i$  et  $Nth_i$  respectivement le nombre de données observées et le nombre de données théoriques pour chaque classe de l'histogramme. Sous l'hypothèse nulle, cette statistique suit une loi du  $\chi^2$  à  $(J - 1 - \nu)$  degrés de liberté, avec  $\nu$  le nombre de paramètres estimés (dans notre cas, deux paramètres :  $x_0$  et  $\sigma$ ). L'histogramme comportant 25 bins, nous avons donc 22 degrés de libertés. La figure 3.9 illustre la loi du  $\chi^2$  pour 22 degrés de libertés. En autorisant une erreur  $\alpha$  de 5% (standard pris dans la plupart des cas), la valeur du  $\chi^2$  ne doit pas dépasser 33.9244 pour vérifier l'hypothèse nulle, à savoir que les données suivent la loi gaussienne. Dans notre cas, le  $\chi^2$  vaut 22.1825, ce qui signifie que l'hypothèse nulle n'est pas rejetée. On peut donc supposer que les données suivent bien une loi normale d'écart type  $\sigma = -1,56.10^{-3}^\circ$  et de moyenne  $x_0 = 33,7.10^{-3}^\circ$ .



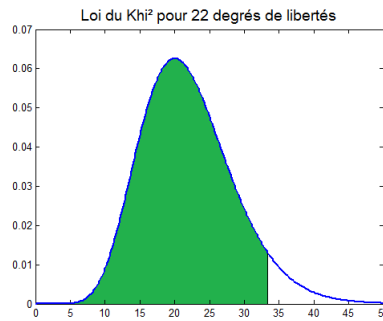


FIGURE 3.9 – Loi du  $\chi^2$  pour 22 degrés de liberté. Avec une erreur  $\alpha = 5\%$  autorisée, le  $\chi^2$  doit être inférieur à 33.9244 pour que l'hypothèse nulle soit respectée.

### 3.2/ DÉTECTION D'ARÊTES SAILLANTES PAR IMAGERIE

Comme présenté dans l'état de l'art, les deux principaux attributs complémentaires aux points 3D sont la normale à la surface et la courbure. Comme nous venons de le présenter, l'extraction des normales est possible. Malheureusement à notre connaissance, il n'existe pas de méthode permettant de calculer une courbure 3D en chaque point à partir d'une paire d'images. Pourtant, l'attribut de courbure est utilisé dans beaucoup d'algorithmes et principalement dans les étapes de segmentation qui permettent de fortement simplifier les étapes de simplification et de recherche de primitives. Pour cette raison, nous avons dans un premier temps essayé d'extraire un attribut supplémentaire assez proche de la courbure à partir des images fournies par le scanner : un rayon de courbure local.

#### 3.2.1/ ESTIMATION DU RAYON DE COURBURE

Dans cette partie, nous cherchons à estimer un rayon de courbure directionnel à partir des images. Le principe de la méthode est illustré en figure 3.10. Pour bien comprendre le principe, prenons un cas particulier où l'objet est un cylindre. En projetant une ligne à l'aide d'un projecteur, on génère un plan lumineux  $\Pi_m$  qui va intersecter l'objet 3D. Cette intersection forme un portion de cercle dans ce même plan (dans le cas où le plan lumineux est perpendiculaire à l'axe du cylindre). La visualisation de la scène par la caméra forme une portion ellipse dans le plan de l'image caméra  $\Pi_i$ . Ce problème peut être vu comme la transformation projective d'une conique (qui est la forme générale d'une ellipse) entre le plan  $\Pi_m$  et  $\Pi_i$ .

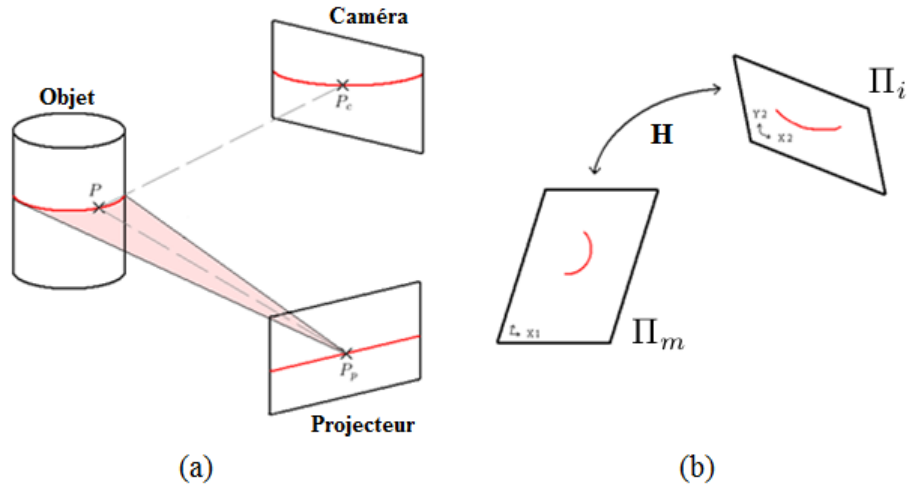


FIGURE 3.10 – Schématisation de la problématique. (a) Principe utilisé pour déterminer la courbure. (b) Relation entre le plan d'éclairage (demi-cercle) et le plan de la caméra (ellipse).

La relation entre deux espaces de même dimension, en l'occurrence le plan  $\Pi_m$  et le plan  $\Pi_i$ , est une matrice de transfert communément appelé homographie. De manière générale, pour deux points  $X_i$  et  $X'_i$  appartenant respectivement à  $\Pi_m$  et  $\Pi_i$ , on a :

$$X'_i = HX_i. \quad (3.5)$$

Comme expliqué précédemment, ce problème peut être vu comme une transformation projective de la conique  $A_Q$  (estimée depuis la portion de l'ellipse visible dans la caméra) entre le plan  $\Pi_i$  et le plan  $\Pi_m$ . On notera la nouvelle conique résultante dans le plan  $\Pi_m$  :  $A'_Q$ . Prenons un point  $q = (x, y, 1)^T$  appartenant au plan  $\Pi_i$  et sa projection  $q' = (x', y', 1)^T$  appartenant au plan  $\Pi_m$ . D'après la définition d'une conique, on peut écrire :

$$q^T A_Q q = 0, \quad (3.6)$$

on sait que

$$q' = Hq, \quad (3.7)$$

et on cherche

$$q'^T A'_Q q' = 0. \quad (3.8)$$

De l'équation (1.6) vient :

$$(H^{-1}Hq)^T A_Q H^{-1}Hq = 0, \quad (3.9)$$

$$(H^{-1}q')^T A_Q H^{-1}q' = 0, \quad (3.10)$$

$$q'^T H^{-T} A_Q H^{-1} q' = 0. \quad (3.11)$$

De (1.8) et (1.11) on en déduit :

$$A'_Q = H^{-T} \cdot A_Q \cdot H^{-1}. \quad (3.12)$$

$A_Q$  est connue étant donné qu'il s'agit d'une observation de l'image. Si l'on connaît  $H$ , à l'aide d'une étape de calibrage supplémentaire, il est alors possible de déterminer la conique  $A'_Q$  dans le plan  $\Pi_m$ , qui peut s'écrire de la façon suivante :

$$A'_Q = \begin{bmatrix} A' & B'/2 & D'/2 \\ B'/2 & C' & E'/2 \\ D'/2 & E'/2 & F' \end{bmatrix}. \quad (3.13)$$

De cette matrice, il est possible d'extraire différents paramètres comme le centre de la conique ou encore le rayon de courbure de la portion du cercle dans le plan  $\Pi_m$ . Afin de vérifier le bon fonctionnement et la robustesse de cette méthode, nous avons comparé le rayon du cylindre ainsi obtenu avec le rayon mesuré au pied à coulisse.

## RÉSULTATS

Le tableau 3.1 montre un exemple de courbure calculée avec cette approche sur différents cylindres de différents diamètres, mesurés à l'aide d'un pied coulisse. Pour ces trois exemples, nous obtenons une erreur maximale de 3% du le rayon réel de l'objet. Cependant, la qualité des résultats obtenus dépend fortement du nombre de points utilisés pour estimer la courbure de l'objet comme le montre le tableau 3.2.

Objet test	Rayon mesuré au pied à coulisse	Rayon estimé par notre méthode
Cylindre 1	4,50 mm	4,37 mm
Cylindre 2	3,45 mm	3,41 mm
Cylindre 3	2,82 mm	2,75 mm

TABLE 3.1 – Résultats obtenus sur trois cylindres différents.

Rayon réel	Rayon avec 100 points	Rayon avec 50 points	Rayon avec 20 points
4,50 mm	4,37 mm	4,85 mm	6,2 mm
3,45 mm	3,41 mm	3,65 mm	4,52 mm
2,82 mm	2,75 mm	2,74 mm	3,80 mm

TABLE 3.2 – Résultats obtenus en fonction du nombre de points utilisés pour déterminer la conique  $A_Q$ .

On constate que cette approche reste très dépendante du nombre de points utilisés pour estimer la conique  $A_Q$  depuis l'image, et ne fournit donc pas de résultats suffisamment précis pour être utilisé sur un objet complexe. C'est pourquoi, nous avons cherché à estimer un autre paramètre, différent de la courbure, mais qui peut être utilisé de manière similaire à l'attribut de la courbure 3D pour segmenter les objets présents dans la scène : les discontinuités.

### 3.2.2/ DISCONTINUITÉS

Généralement, on parle de discontinuité lorsqu'il y a rupture d'une forme continue, comme une arête saillante. Cependant, limiter les discontinuités aux simples arêtes saillantes d'un objet serait trop réducteur par rapport aux mesures de courbure 3D qui, quant à elles, fournissent une information quantifiée pour chaque point du nuage. Ici, l'idée est donc de définir un coefficient de discontinuité pour chaque point du nuage afin de segmenter les différents objets qui composent la scène. Dans le cas d'une surface plane, il n'y a pas de discontinuité et de ce fait le coefficient de discontinuité de chaque point est égal à 0. A contrario, plus la forme sera discontinue et plus le coefficient sera important.

### 3.2.3/ PRINCIPE

Le principe utilisé pour estimer les discontinuités est présenté en figure 3.11. Si on éclaire l'objet avec une ligne (issu d'un laser ou d'un projecteur), celle-ci va se déformer en fonction de la forme de l'objet. La caméra va capturer cette déformation qui sera alors visible dans l'image. Un peu à l'image du calcul des normales, où l'on estime la normale à la surface à partir de vecteurs 2D, l'idée est d'étudier la variation des normales à la ligne vue dans l'image. En utilisant la même grille de germe que dans la méthode de calcul des normales, il est possible de définir une ligne dans l'image de référence, et de visualiser la déformation de celle-ci dans l'image de la seconde caméra.

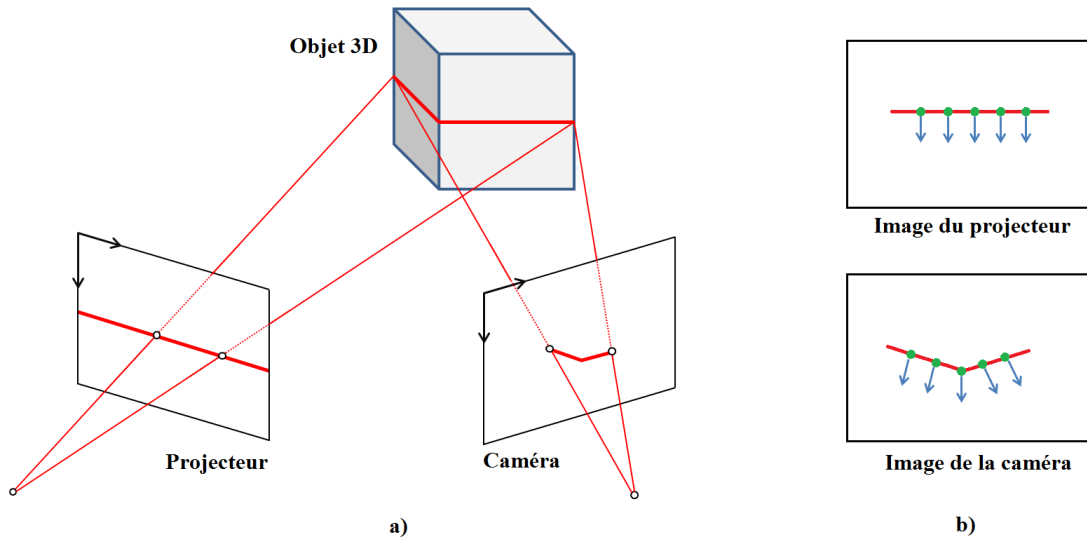


FIGURE 3.11 — Principe utilisé pour calculer les coefficients de discontinuité. a) La ligne projetée va se déformer en fonction de la forme de l'objet. b) L'orientation des normales à la ligne varie en fonction de la forme 3D de l'objet.

A chaque point de la grille de germe, est associé un vecteur normal à la ligne noté  $n$ . Le coefficient de discontinuité est alors calculé de la façon suivante pour un point  $p_i$  de normale  $n_i$  :

$$c_i = \left| \frac{\vec{n}_i \cdot \vec{n}_{i-1} + \vec{n}_i \cdot \vec{n}_{i+1}}{2} \right|, \quad (3.14)$$

avec  $n = [n_x, n_y]$ . Ainsi, pour chaque point de la grille, il est possible d'obtenir un coefficient de discontinuité pour une direction donnée. Cette étape est ensuite répétée pour d'autres directions ce qui permet d'obtenir des coefficients de discontinuité pour chaque point dans quatre directions différentes (une ligne horizontale  $c_{i1}$ , une ligne verticale  $c_{i2}$  et deux lignes diagonales  $c_{i3}$  et  $c_{i4}$ ). Le coefficient final au point  $p_i$ , noté  $C_i$ , est la moyenne des quatre coefficients  $c_{i1}$ ,  $c_{i2}$ ,  $c_{i3}$  et  $c_{i4}$ .

Le calcul des coefficients de discontinuité se rapproche de celui utilisé dans la méthode de Dong *et al.* [60], permettant de calculer la courbure sur un maillage à partir des normales à la surface prises dans un voisinage de  $k$  points. La différence majeure entre les deux approches réside dans le fait que nous ne faisons pas appel au nuage de points pour estimer les coefficients de discontinuité, mais seulement aux coordonnées image, ce qui est nettement moins lourd en temps de calcul. De plus, les algorithmes de courbure s'utilisent généralement sur des maillages complets et fermés, ce qui est rarement le cas lorsque l'on ne prend qu'une seule acquisition. Enfin, cette méthode est également applicable à différentes densités de la grille. Dans le cas d'un faible échantillonnage (grille

de points espacés) les coefficients des discontinuité fourniront une information plutôt globale, permettant de détecter des variations brusques, comme les arêtes. A contrario, si la grille de points est serrée, il est possible de quantifier les variations locales de la surface.

### 3.2.4/ RÉSULTATS

Pour chaque point appairé, entre les deux images du scanner, est calculé un coefficient de discontinuité. La figure 3.12 présente les résultats que l'on obtient avec cette méthode sur un objet composé de plusieurs formes homogènes (trois plans, un cylindre et une sphère) et présentant de fortes discontinuités entre les différentes formes. Les zones à forte courbure sont bien représentées par un fort coefficient de discontinuité, alors que les points qui composent les zones homogènes ont un faible coefficient.

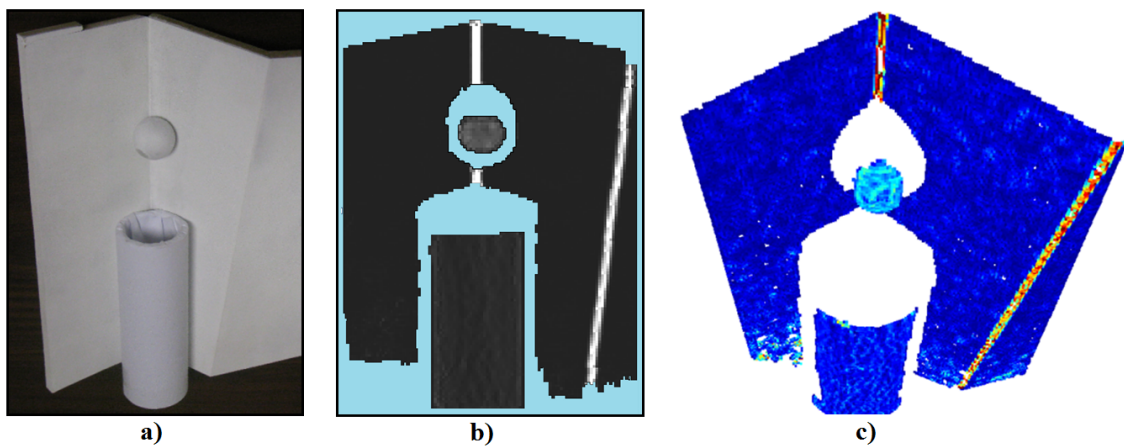


FIGURE 3.12 – *Calcul des coefficients de discontinuité. a) Objet test. b) Coefficients de discontinuité calculés pour chaque point image (la zone en bleu représente les points n'ayant pas de coefficients de discontinuité). c) Résultats plaqués sur le nuage de points en fausses couleurs pour un meilleur rendu visuel.*

Ces résultats mettent en avant la ressemblance visuelle entre cette méthode et un calcul de courbure classique sur un nuage de points, à la différence que le calcul s'effectue uniquement à partir des images du scanner ce qui est beaucoup moins coûteux en ressources.

### 3.3/ PRIMITIVES

Comme nous l'avons vu dans l'état de l'art, les primitives constituent une information importante que ce soit pour la modélisation, la simplification de données, ou bien encore simplement comme information complémentaire aux points 3D. Si ces informations sont généralement extraites depuis le nuage de points ou le maillage, nous avons pu voir dans la méthode présentée par Roseline Bénére [89] qu'il était possible d'extraire un cylindre uniquement en utilisant les normales 3D (sans utiliser les points 3D) à l'aide de la représentation des points sur la sphère gaussienne. Nous proposons dans ce qui suit de généraliser cette approche.

#### 3.3.1/ LA SPHÈRE GAUSSIENNE

La sphère gaussienne est l'extension de la carte de Gauss [90] (ou *Gauss Map* en anglais) en 2D. Cette dernière permet de représenter une courbe à partir des normales à la surface en plaçant celles-ci au centre d'un disque de rayon unitaire, en supposant que les normales à la surface sont normalisées (figure 3.13).

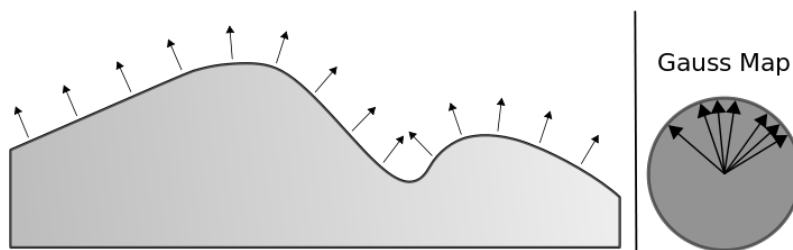


FIGURE 3.13 – Représentation de la carte de Gauss. Toutes les normales à la surface sont placées au centre d'un disque unitaire.

La sphère de Gauss ou sphère gaussienne est la représentation de la carte gaussienne en 3D. Les normales  $(n_x, n_y, n_z)$  sont cette fois-ci placées au centre d'une sphère unitaire. L'intérêt de cette représentation apparaît clairement quand on considère des formes géométriques simples comme les primitives. En effet, certaines primitives (plans, cylindres, cônes et sphères) possèdent une représentation particulière dans la sphère gaussienne qui peut être exploitée pour les identifier (figure 3.14). Par exemple, pour un plan parfait toutes les normales à la surface sont identiques, ce qui implique que sa représentation sur la sphère gaussienne sera un point (figure 3.14-a).

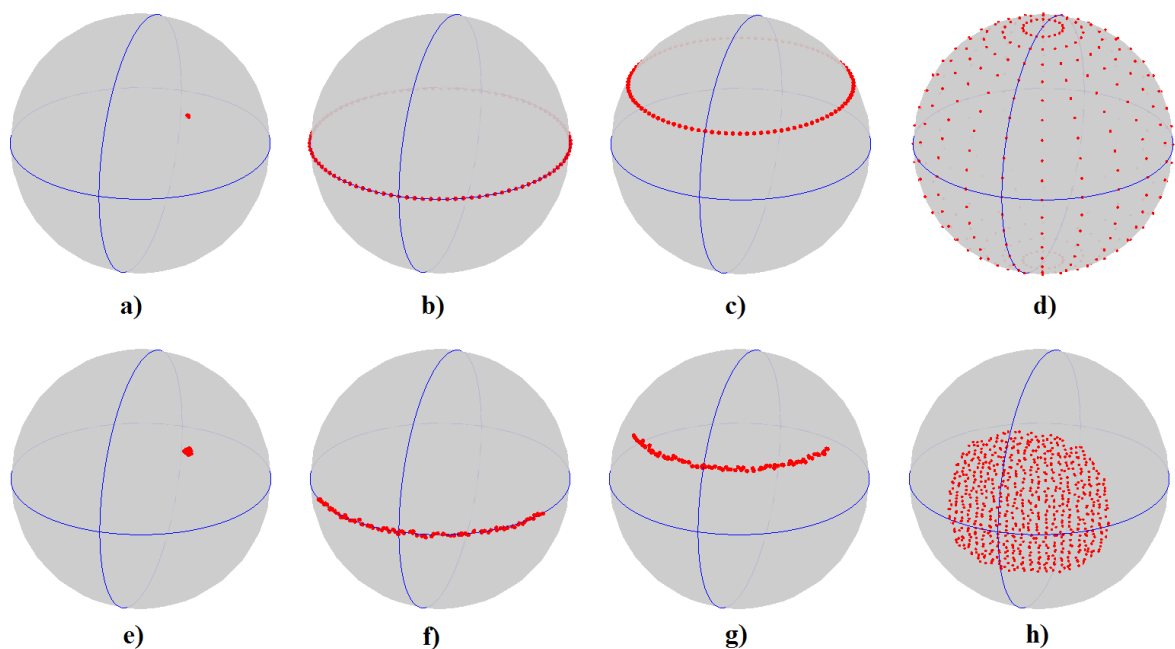


FIGURE 3.14 – Représentation de différentes primitives géométriques sur la sphère gaussienne. a-d) Représentation théorique pour un plan, un cylindre, un cône puis une sphère. e-h) Représentation pratique pour les mêmes primitives avec notre système. La représentation est partielle car l'acquisition ne fournit qu'une portion de la primitive.

Sur la figure 3.14, on remarque que les représentations des différentes primitives sont très différentes les unes des autres. Ici, l'idée consiste à utiliser les informations calculées à partir des images (normales 3D et coefficient de discontinuité) afin de déterminer si un objet présent dans l'image correspond ou non à une primitive. Comme nous l'avons présenté précédemment, les normales à la surface calculées par notre méthode sont légèrement bruitées, ce qui donne de légères variations au niveau de la représentation sur la sphère de Gauss. De plus, en pratique la représentation sur la sphère gaussienne est en général partielle car seule une partie de la primitive est visible avec un système de stéréovision (3.14-e à h). Cependant, la répartition générale des points sur la sphère gaussienne reste très spécifique à chaque type de primitive.

### 3.3.2/ ANALYSE EN COMPOSANTES PRINCIPALES

Pour tenir compte de la nature bruitée des données nous proposons d'utiliser une méthode statistique pour détecter et identifier les primitives à partir de la représentation sur la sphère de Gauss. L'analyse en composantes principales (ACP) est une méthode d'analyse statistique des données qui permet de rechercher les axes qui décorrèlent le mieux les données, autrement dit les axes qui « expliquent » le mieux la répartition des



points. Concrètement, ces axes se calculent comme étant les vecteurs propres de la matrice de covariance des données. Dans le cas de la sphère gaussienne, les données sont les normales à la surface de l'objet  $n_i = (n_{1i}, n_{2i}, n_{3i})^T$  que l'on peut rassembler dans une matrice de la façon suivante :

$$N = \begin{bmatrix} n_{11} & \dots & n_{1i} & \dots & n_{1K} \\ n_{21} & \dots & n_{2i} & \dots & n_{2K} \\ n_{31} & \dots & n_{3i} & \dots & n_{3K} \end{bmatrix}. \quad (3.15)$$

Généralement, dans le calcul de l'ACP les données sont centrées par rapport à la moyenne suivant chaque composante :

$$\bar{n}_1 = \frac{\sum_{i=1}^K n_{1i}}{K}, \quad (3.16)$$

ce qui donne :

$$N' = \begin{bmatrix} n_{11} - \bar{n}_1 & \dots & n_{1i} - \bar{n}_1 & \dots & n_{1K} - \bar{n}_1 \\ n_{21} - \bar{n}_2 & \dots & n_{2i} - \bar{n}_2 & \dots & n_{2K} - \bar{n}_2 \\ n_{31} - \bar{n}_3 & \dots & n_{3i} - \bar{n}_3 & \dots & n_{3K} - \bar{n}_3 \end{bmatrix}. \quad (3.17)$$

La matrice de covariance s'écrit alors :

$$C = N'N'^T. \quad (3.18)$$

L'analyse en composante principale permet donc de déterminer les vecteurs propres  $\lambda$  et les trois valeurs propres associées  $X$  de la matrice de covariance. De part la grande différence qui existe entre les différents types de primitives, nous verrons que les vecteurs et valeurs propres de la matrice de covariance sont très différents d'une primitive à l'autre. Ce que nous proposons dans cette partie peut se diviser en deux parties : une partie de classification par les valeurs propres  $X$  et une partie de modélisation en utilisant les vecteurs propres  $\lambda$ .

### 3.3.2.1/ IDENTIFICATION DES PRIMITIVES

Afin de classer les différentes primitives, nous proposons une approche par apprentissage. Pour ce faire, nous avons modélisé différentes primitives géométriques, puis avons ajouté un bruit gaussien sur les normales, ce bruit possédant les mêmes propriétés que celui caractérisé précédemment en section 3.1.3. Pour chaque primitive modélisée de

cette façon, nous avons alors calculé les valeurs propres de la matrice de covariance de la représentation gaussienne. Le scanner utilisé ne pouvant pas calculer de points sur une surface ayant une inclinaison de plus de 60°(donnée constructeur) par rapport au point de vue de la caméra de référence, nous avons limité la modélisation des primitives à un champ de vue de 120°(de -60° à +60 °). Pour chaque type de primitives, nous avons généré dix primitives différentes en modifiant les différents paramètres permettant de les modéliser :

- Plan : Un plan est généralement modélisé par un point (position du plan) et un vecteur normal à la surface du plan (orientation).
- Cylindre : Le cylindre peut être défini par un rayon, un vecteur (orientation de l'axe) et un point 3D situé sur l'axe du cylindre (position).
- Cône : Le cône est un cylindre particulier. Contrairement au cylindre, les normales à la surface du cône ne sont pas perpendiculaires à l'axe de celui-ci mais forment un angle fixe avec l'axe.
- Sphère : La sphère est définie, quand à elle, par un centre et un rayon.

Le tableau 3.3 présente les valeurs propres  $\lambda_1$ ,  $\lambda_2$  et  $\lambda_3$  obtenues pour chaque type de primitives. La valeur moyenne correspond à la moyenne des 10 valeurs propres obtenues pour chaque type de primitives. A partir de ce tableau, nous montrons qu'il est possible de répartir les différents type de primitives en trois classes : une première classe pour les plans où les valeurs propres sont très faibles, une deuxième classe pour les cylindres et les cônes et une troisième classe pour les sphères.

Type de primitives		Plan	Cylindre	Cône	Sphère
Valeur moyenne	$\lambda_1$	$1,52.10^{-4}$	$2,01.10^{-1}$	$1,98.10^{-1}$	$1,52.10^{-1}$
	$\lambda_2$	$1,49.10^{-4}$	$1,23.10^{-2}$	$1,42.10^{-2}$	$1,24.10^{-1}$
	$\lambda_3$	$1,38.10^{-4}$	$2,08.10^{-4}$	$2,12.10^{-4}$	$1,42.10^{-3}$
Valeur maximale	$\lambda_1$	$2,01.10^{-4}$	$2,30.10^{-1}$	$2,25.10^{-1}$	$1,95.10^{-1}$
	$\lambda_2$	$1,87.10^{-4}$	$2,15.10^{-2}$	$1,91.10^{-2}$	$1,78.10^{-1}$
	$\lambda_3$	$1,70.10^{-4}$	$2,71.10^{-4}$	$3,23.10^{-4}$	$1,97.10^{-3}$
Valeur minimale	$\lambda_1$	$1,21.10^{-4}$	$1,80.10^{-1}$	$1,72.10^{-1}$	$1,27.10^{-1}$
	$\lambda_2$	$1,32.10^{-4}$	$5,20.10^{-3}$	$4,81.10^{-3}$	$1,12.10^{-1}$
	$\lambda_3$	$1,19.10^{-4}$	$1,21.10^{-4}$	$1,09.10^{-4}$	$7,54.10^{-4}$

TABLE 3.3 – Valeurs propres de la matrice de covariance pour les différentes primitives modélisées.

Tout objet présent dans la scène peut être classé en trois catégories de primitives, simplement à partir des valeurs propres de la matrice de covariance. Si les valeurs propres d'un objet sont comprises dans l'intervalle  $[\lambda_{min}, \lambda_{max}]$ , défini par les valeurs minimales et maximales de chaque type de primitives, alors l'objet est considéré comme appartenant potentiellement à cette classe, sinon il n'est pas identifié comme une primitive. Cette méthode d'identification a la particularité de classer rapidement les primitives en trois catégories à partir d'une simple analyse des normales 3D, elles mêmes extraites directement depuis les images. Cependant, bien que classer les différentes primitives à partir des valeurs propres de la matrice de covariance permet de faire un premier tri, ce n'est pas suffisant pour affirmer qu'il s'agit bel et bien d'une primitive. C'est pourquoi l'étape suivante consiste à modéliser la primitive afin de vérifier que les points 3D appartiennent bien à la primitive supposée.

### 3.3.2.2/ MODÉLISATION DES PRIMITIVES

L'étape de modélisation de la primitive est une étape de vérification essentielle pour éviter de commettre une erreur. En effet, les valeurs propres étant très différentes d'une primitive à l'autre, il est peu probable de commettre une confusion à ce niveau. Malgré tout, il est possible qu'un objet qui ne soit pas une primitive puisse être identifié comme telle. À la différence des méthodes présentées précédemment permettant d'extraire des données, l'étape de modélisation de primitives requiert des informations 3D.

Dans l'état de l'art, il existe déjà différentes approches permettant de modéliser des primitives à partir d'un nuage de points, comme la méthode présentée par Lukács [91] dont le principe est d'approximer des primitives par une régression sur les moindres carrés, la méthode de Schnabel *et al.* [83] permettant d'améliorer l'algorithme Ransac, ou encore Bey *et al.* [92] qui proposent une méthode reposant sur des informations *a priori* couplées à une méthode statistique pour approximer des cylindres. Cependant, ces approches sont basées sur l'étude du nuage de points et sont donc généralement coûteuses en mémoire et en temps de calcul. Il existe d'autres méthodes pour modéliser une primitive sur un nuage de points, qui ont la particularité de ne pas utiliser directement le nuage de points [93, 89]. Ces méthodes sont basées sur une utilisation la sphère de Gauss pour identifier et modéliser des cylindres. En nous appuyant sur les travaux déjà réalisés pour les cylindres, nous proposons une méthode permettant de modéliser les différentes primitives en utilisant les résultats fournis par l'ACP sur les données de la sphère gaussienne. Nous avons vu dans la partie précédente que la matrice de covariance pouvait s'écrire :

$$C = N'N'^T, \quad (3.19)$$

avec  $N'$  la matrice des données centrées. La matrice de covariance est par construction symétrique, ce qui implique que ses vecteurs propres forment une base orthonormale. Supposons une surface de révolution d'axe unitaire  $U = (u_1, u_2, u_3)^T$ . Notons  $\alpha$  l'angle entre les normales à la surface  $n_i$  et l'axe de la surface  $U$ , de telle sorte que le produit scalaire entre la normale et le vecteur directeur  $U$  soit :

$$n_i^T U = U^T n_i = \cos \alpha_i. \quad (3.20)$$

On cherche à vérifier que  $U$  est toujours un des vecteurs propres de la matrice de covariance, ce qui nous permettrait d'utiliser le plan formé par les deux autres vecteurs propres pour modéliser nos primitives.  $U$  est un vecteur propre s'il respecte l'équation suivante :

$$CU = \lambda U. \quad (3.21)$$

Afin de ne pas nuire à la lisibilité du document, la démonstration est effectuée en Annexe A. Dans le cas de données bruitées, nous sommes parvenus à démontrer que l'axe  $U$  n'est pas un vecteur propre, mais qu'il est très proche de celui-ci et peut être supposé comme tel. De ce fait, en utilisant les vecteurs propres de la matrice de covariance il est possible de modéliser une primitive approximative.

## PLAN

Un plan peut être modélisé à partir d'un point et de sa normale au plan. En supposant que  $U$  est un vecteur propre de la matrice de covariance, il correspond alors à la normale au plan. Les deux autres vecteurs propres forment un plan tangent à la sphère, passant par le barycentre des normales projetées sur la sphère. Connaissant la normale au plan (vecteur  $U$  extrait de la matrice de covariance) ainsi que son point 3D correspondant (le barycentre des normales), il est alors possible de modéliser un plan 3D en utilisant un minimum d'informations 3D.

## CYLINDRE

Un cylindre peut se modéliser par un rayon, un axe de révolution et un point sur cet axe. Dans [93], les auteurs proposent une méthode pour approximer un cylindre sur un nuage de points en deux temps, en utilisant la sphère gaussienne puis l'approximation d'un cercle. En effet, la représentation d'un cylindre parfait sur la sphère gaussienne représente un cercle dont le centre est le centre de la sphère, et dont le vecteur normal au plan qui comprend ce cercle représente l'axe du cylindre. Les auteurs proposent donc d'extraire l'axe du cylindre depuis la sphère gaussienne, puis de projeter les points 3D dans un plan perpendiculaire à cet axe pour déterminer le rayon (rayon du cercle) et un point de l'axe (centre du cercle).

Dans notre cas, toujours en supposant que  $U$  est un vecteur propre de la matrice de covariance, l'axe du cylindre peut être assimilé au vecteur  $U$ . Les deux autres vecteurs propres forment ainsi un plan perpendiculaire à l'axe étant donné que les vecteurs propres forment une base orthogonale. La seconde partie de notre approche est identique à celle présentée par Thomas Chaperon [93], nous projetons les points 3D dans le plan obtenu avec l'ACP, ce qui forme un arc de cercle. En ajustant un cercle sur ces points par la méthode des moindres carrés, il est ainsi possible d'extraire un point sur l'axe (centre du cercle) et le rayon, et donc de modéliser un cylindre.

## CÔNE

Le cas du cône est très similaire à celui du cylindre, la seule différence se situe au niveau de l'angle entre les normales à la surface et l'axe de la primitive. De ce fait, la représentation du cône au niveau de la sphère gaussienne est différente de celle du cylindre, le cercle ne passe plus par le centre de la sphère. Cette distance est importante car elle est directement liée à l'angle du cône (figure 3.15).

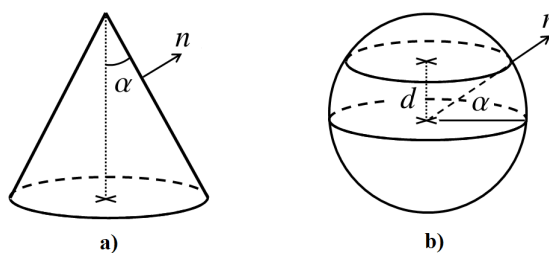


FIGURE 3.15 — Calcul de l'angle du cône. a) Normale à la surface et angle du cône. b) Représentation sur la sphère gaussienne et distance entre le centre du cercle et de la sphère.

Dans la partie précédente, nous avons vu qu'il était possible d'identifier trois catégories de primitives à partir des valeurs propres de la matrice de covariance ; les plans, les sphères et les cylindres/cônes. C'est également cette distance entre le plan, formé par les vecteurs propres, et le centre de la sphère qui nous permet de différencier les cylindres des cônes. Dans sa thèse, Thomas Chaperon [94] propose une méthode pour modéliser un cône. La méthode est identique à celle du cylindre, avec en plus un calcul de l'angle. Comme le montre la figure 3.15, l'angle du cône peut être calculé par la relation suivante :

$$\alpha = \tan^{-1} \left( \frac{d}{r} \right), \quad (3.22)$$

avec  $r$  le rayon de la sphère (ici  $r = 1$  car la sphère est unitaire) et  $d$  la distance entre le centre du cercle et le centre de la sphère. En appliquant cette méthode, il est possible d'obtenir l'angle du cône et ainsi modéliser la primitive.

## SPHÈRE

La sphère se modélise à partir d'un point (le centre de la sphère) et d'un rayon. Contrairement aux autres primitives, la représentation gaussienne d'une sphère n'apporte aucune information par rapport à la position des points. En effet, la représentation gaussienne d'une sphère est une sphère. De ce fait, la méthodologie utilisée pour estimer le centre et le rayon de la sphère reste très conventionnelle. Par définition, toutes les normales à la surface d'une sphère se coupent au centre de celle-ci. Les données étant bruitées, les normales ne se coupent pas en un point unique. Pour remédier à ce problème, Slabaugh *et al.* [95] présentent une méthode dont le principe consiste à déterminer le point d'intersection optimal de plusieurs lignes 3D en utilisant la méthode des moindres carrés, permettant ainsi d'estimer le centre optimal de la sphère. Le rayon  $r$  de la sphère peut alors être obtenu par la relation suivante :

$$r = \frac{\sum_{i=1}^N \left( (p_{xi} - c_x)^2 + (p_{yi} - c_y)^2 + (p_{zi} - c_z)^2 \right)}{N}, \quad (3.23)$$

avec  $p_{xi}$ ,  $p_{yi}$ ,  $p_{zi}$  la position d'un point  $p_i$  dans l'espace 3D et  $c_x$ ,  $c_y$  et  $c_z$  la position du centre de la sphère. L'avantage d'utiliser l'ACP sur les données de la sphère gaussienne réside dans le fait de pouvoir modéliser des primitives en utilisant un minimum d'information 3D et les méthodes sont donc très rapides et peu coûteuses en mémoire.

### 3.3.2.3/ VÉRIFICATION

Une fois les primitives modélisées, la dernière étape consiste à vérifier que les points 3D appartiennent effectivement à ces primitives. Pour cela, la méthode la plus répandue consiste à calculer la distance entre la surface modélisée et le nuage de points. Il existe différents critères de distance, bien que le plus utilisé soit la distance entre un point et son projeté orthogonale sur la surface. Ainsi, si la distance entre les points et la surface est inférieure à un certain seuil de tolérance, l'objet identifié pourra être retenu comme appartenant à un type de primitive.

### 3.3.2.4/ RÉSULTATS

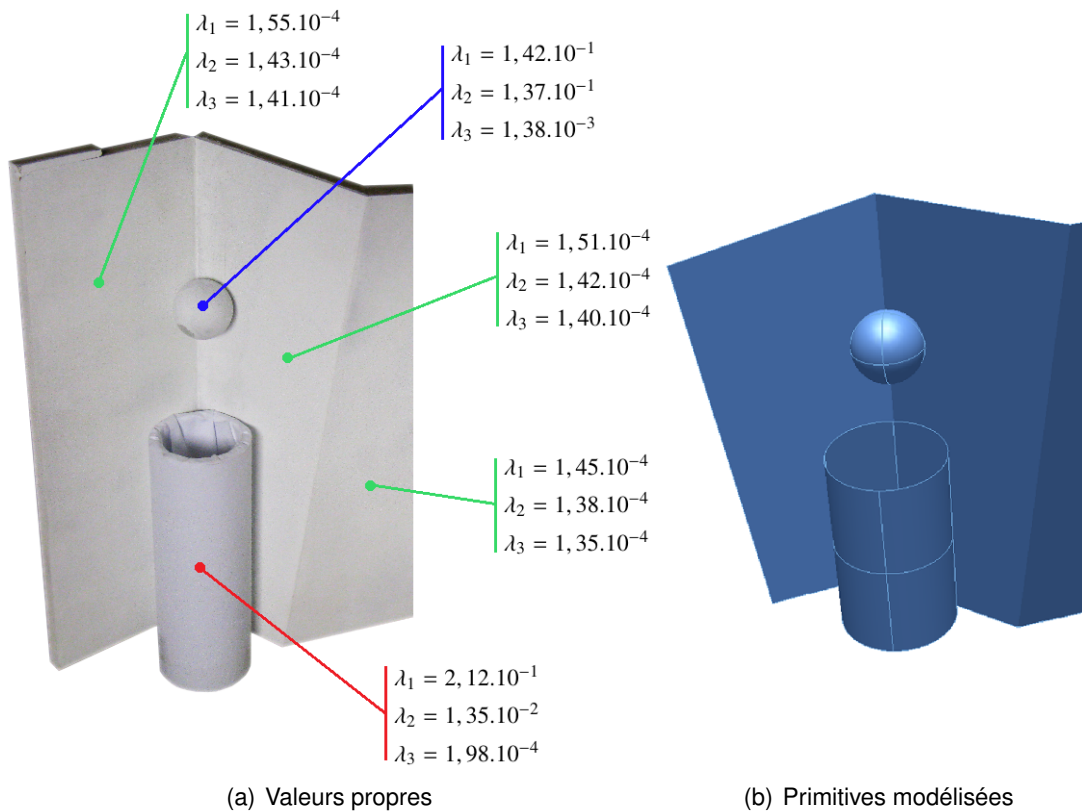


FIGURE 3.16 – Résultats sur une pièce test. a) Valeurs propres obtenues sur les différents objets. Les valeurs propres qui correspondent à un plan sont représentées en vert, en bleu pour une sphère et en rouge pour un cylindre. b) Primitives modélisées à partir des vecteurs propres.

La figure 3.16 présente les résultats obtenus sur une pièce test composée de plusieurs type de primitives. Dans cet exemple, on constate que les valeurs et vecteurs propres obtenus à partir de la représentation gaussienne de l'objet permettent d'identifier les différentes primitives, mais aussi de les modéliser.

Les figures 3.17 et 3.18 présentent également les résultats que l'on obtient sur des primitives qui ne sont pas parfaites. Malgré la présence de petits défauts (présence d'un méplat sur le cylindre et de rainures sur le cône), il est tout de même possible d'identifier et de modéliser des primitives sur ces deux objets.

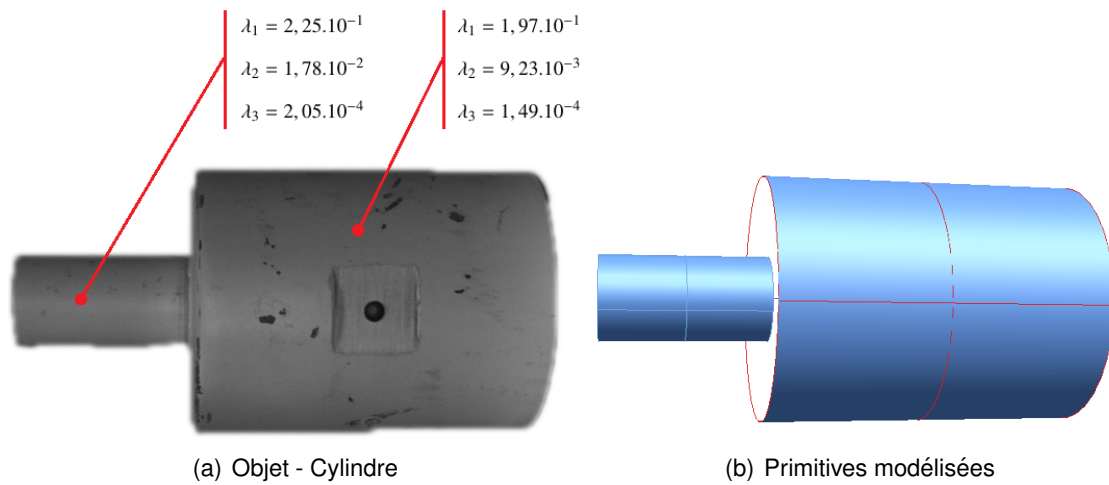


FIGURE 3.17 – Résultats sur un objet test composé de deux cylindres et d'un méplat. a) Valeurs propres obtenues pour les deux cylindres. b) Primitives modélisées à partir des vecteurs propres.

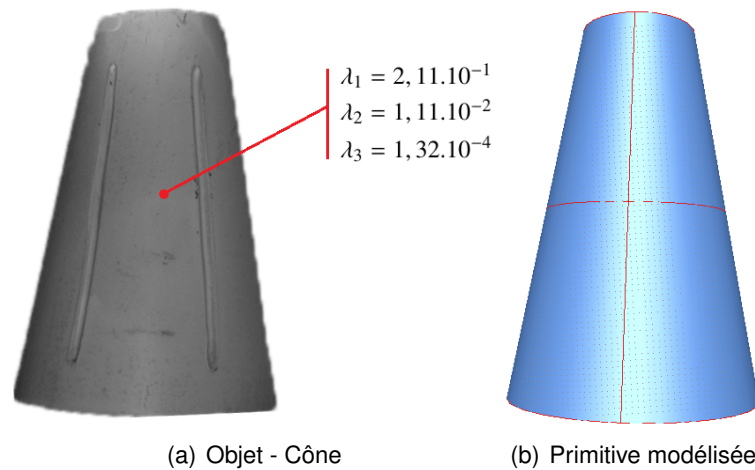


FIGURE 3.18 – Résultats sur un cône. a) Valeurs propres obtenues. b) Primitive modélisée à partir des vecteurs propres.



### 3.4/ CONCLUSION

Dans cette partie, nous avons présenté plusieurs approches permettant d'extraire différentes informations à partir des images du scanner : les normales 3D, les discontinuités et des primitives géométriques. L'approche utilisée pour extraire les normales 3D est inspirée d'une technique existante présentée par [56] et adaptée au scanner utilisé. En s'inspirant de l'état de l'art sur la chaîne 3D, nous avons cherché à extraire un attribut proche de la courbure : le rayon de courbure. Cependant cette approche étant soumise à de trop nombreuses contraintes, il a fallu extraire un autre attribut permettant de segmenter les différents objets présents dans la scène (principale utilisation de la courbure 3D dans la chaîne classique) : les discontinuités. Enfin, nous avons développé une approche basée sur le calcul de l'ACP de la représentation des données sur la sphère gaussienne, permettant d'extraire des primitives géométriques sans avoir à utiliser le nuage de points. Les différentes primitives ont été identifiées et classées à l'aide d'une méthode d'apprentissage basée sur les valeurs propres de la matrice de covariance obtenue par le calcul de l'ACP. Afin de vérifier et valider cette classification, nous avons proposé une deuxième approche, inspirée de méthodes existantes, pour modéliser les primitives en utilisant les vecteurs propres de la matrice de covariance. L'intérêt de cette approche réside dans une utilisation limitée des données 3D, ce qui permet d'obtenir une classification et une modélisation rapide des primitives.

Dans la chaîne 3D classique, ces différents attributs sont utilisés pour segmenter les objets, identifier des primitives, aider à la reconstruction de surface, etc. Dans notre cas, les attributs extraits dans cette partie vont servir dans de nombreuses applications du système de numérisation dynamique présenté dans la partie suivante.

# SYSTÈME DE NUMÉRISATION DYNAMIQUE

Nous proposons d'adapter l'acquisition à l'objet à numériser afin de se rapprocher des résultats que l'on pourrait obtenir en passant par les différentes étapes de la chaîne 3D classique présentée dans l'introduction, sans pour autant avoir à passer par ces étapes. Pour ce faire, nous proposons un système de numérisation dynamique qui cherche à adapter la densité de points calculés en fonction des informations qui peuvent être extraites depuis les images. Dans la partie précédente, nous avons présenté différentes approches permettant d'extraire des informations à partir des images qui peuvent être utilisées pour réaliser notre système de numérisation dynamique. Nous allons maintenant décrire le fonctionnement global du système et en décrire les différentes étapes.

## 4.1/ FONCTIONNEMENT DU SYSTÈME

Dans l'état de l'art, nous avons présenté le schéma classique de la chaîne 3D, allant de l'acquisition à la simplification des données. Comme mentionné dans l'introduction, ces étapes sont généralement indépendantes les unes des autres, et traitées par des opérateurs différents (notamment la partie acquisition et la partie traitement des données). Le principe que nous proposons ici, consiste à obtenir directement à la sortie du scanner un nuage de points avec une densité adaptée à la forme de la pièce à numériser.

Comme mentionné dans la partie 2.2 de l'état de l'art, un scanner à stéréovision n'a besoin que de deux images et des paramètres de calibration pour calculer les points 3D. À partir de celles-ci, il est possible de trianguler les points 3D en appariant des points communs entre les deux images. Dans le cas du scanner que l'on utilise l'appariement se fait

par une méthode locale par propagation de germe (section 2.2.2.3), et plus précisément une approche simultanée. Cette approche consiste à propager un ensemble de germes pour trouver les correspondants. Dans notre cas, ces germes ont été choisis de façon à former une grille régulière où chaque germe est espacé des autres d'une même distance  $i$ . Cette distance  $i$  est directement liée à la densité de points que l'on obtiendra à la sortie du scanner. En effet, si on choisit une distance  $i$  faible, la grille sera composée de nombreux germes et le nuage de points en sortie sera dense. A partir de maintenant et pour la suite du document, nous nommerons la densité  $d$ , qui sera exprimée de la façon suivante :

$$d = \frac{1}{i}, \quad (4.1)$$

avec  $i$  la distance en pixels entre deux germes dans une direction donnée. Ainsi une densité de  $d = \frac{1}{3}$  signifie un germe (et donc un point appairé entre les deux images) tous les 3 pixels selon les lignes et les colonnes. (figure 4.1).

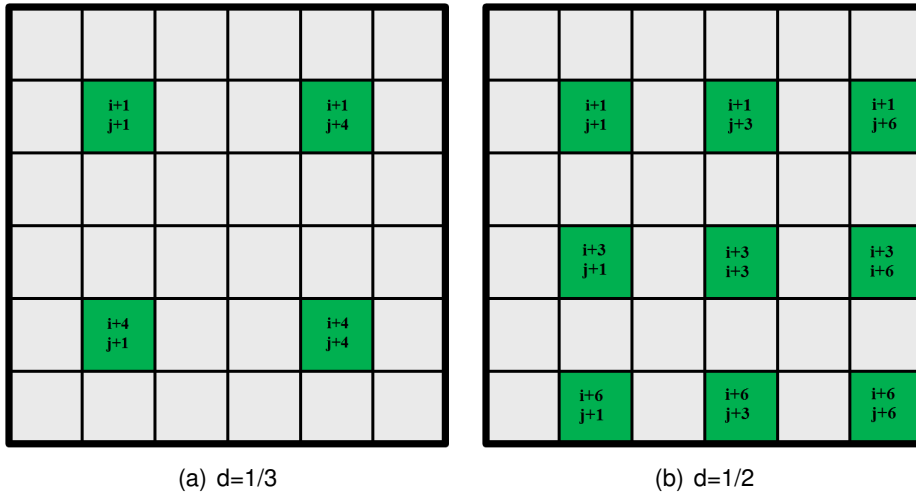


FIGURE 4.1 – Exemple de positionnement des germes (en vert) dans l'image de référence dans le cas où  $d = \frac{1}{3}$  et  $d = \frac{1}{2}$ .

Le schéma présenté en figure 4.2 décrit les différentes étapes du système de façon simplifiée (voir figure 4.3 pour le schéma détaillé). Le principe du système consiste donc, dans un premier temps, à calculer les points 3D pour une densité  $d = \frac{1}{i}$  fixée. A ce moment, la question qui se pose est la suivante : quel critère peut-on utiliser pour déterminer les zones qui ont besoin d'être raffinées ? Pour répondre à cette question, nous nous sommes inspirés du fonctionnement de la chaîne 3D classique, où la plupart des méthodes de simplification d'un nuage de points cherchent à préserver un maximum de points dans les zones à fortes courbures et un minimum de points dans les zones

identifiées comme des primitives. Dans la partie précédente, nous avons montré qu'il était possible de déterminer si un objet appartient ou non à une primitive en utilisant la sphère gaussienne et les données extraites depuis les images.

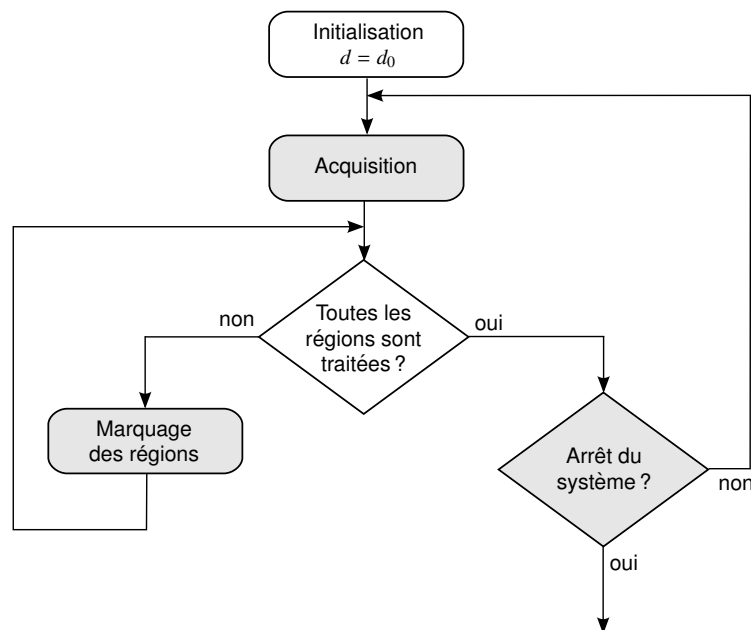


FIGURE 4.2 – *Principe de notre système de numérisation dynamique. L'idée est d'acquérir un nuage de points peu dense puis d'extraire différents attributs afin de sélectionner les zones qui ont besoin ou non d'être raffinées. Le système étant itératif, il permet d'adapter la densité de points à la complexité de la pièce à numériser.*

En s'inspirant des méthodes traditionnelles existantes dans l'état de l'art, nous proposons donc un système qui va, dans un premier temps, calculer des points 3D avec une assez faible densité de points et chercher à extraire des primitives afin de définir les zones qui ont besoin, ou non, d'être raffinées.

### ACQUISITION

Dans un premier temps, il est nécessaire de calculer les points 3D pour une densité  $d = 1/i$  fixée. L'objectif étant de partir d'un nuage de points grossier pour le raffiner par la suite, on commencera avec une densité assez faible. A partir des images fournies par le scanner, et en utilisant la grille de germes obtenue avec la densité  $d$ , il est alors possible de calculer les informations complémentaires suivantes : normales à la surface et coefficients de discontinuité. En utilisant ces informations il est alors possible de segmenter la scène en plusieurs régions homogènes.

### MARQUAGE DES RÉGIONS

Une fois la scène segmentée, il est alors possible d'identifier les différentes primitives, en utilisant la méthode que nous avons présentée en partie 3.3. Les primitives géométriques étant des formes simples, lorsque une région est identifiée comme telle, on suppose qu'il n'est pas nécessaire de calculer d'avantage de points dans cette région. Dans la suite, nous appellerons ces zones, des zones « marquées ». Cependant, si aucune zone présente dans la scène ne peut être identifiée par une primitive, le système ne marquera aucune zone et continuera de raffiner les données jusqu'à obtenir un scan de densité maximum. Pour anticiper ce problème, nous proposons d'ajuster des surfaces paramétriques sur les points 3D. Si la surface évolue entre deux itérations, il est pertinent de calculer de nouveaux points. A contrario, si la surface n'évolue pas, ajouter de l'information est inutile. Dans ce cas les zones sont également marquées. Le but de cette étape consiste donc à marquer les régions identifiées comme appartenant à une primitive, ainsi que les régions dont la surface évolue peu entre deux itérations du système.

### INCRÉMENTATION DU SYSTÈME

Le système va ensuite raffiner l'acquisition avec une densité plus importante  $d = 1/(i - 1)$  sur toutes les zones non marquées lors de l'étape précédente. Ainsi, seules les zones marquées seront conservées avec une faible densité de points. Les itérations se poursuivent ainsi jusqu'à atteindre la densité maximale du scanner, ou lorsqu'il ne reste plus d'objets non marqués dans la scène, comme le présente le schéma détaillé du système (figure 4.3).

## 4.2/ SEGMENTATION

Les premières étapes de notre système de numérisation consistent à calculer les normales et les discontinuités avec les méthodes présentées dans le chapitre précédent. Une fois ces étapes effectuées, la scène est alors segmentée en régions homogènes. Comme cela a été mentionné dans l'état de l'art, la plupart des algorithmes d'extraction de primitives cherchent d'abord à segmenter les différents objets présents dans la scène en utilisant des informations 3D telles que la courbure. Notre système procède de manière similaire en cherchant à segmenter les différents objets directement depuis les images.

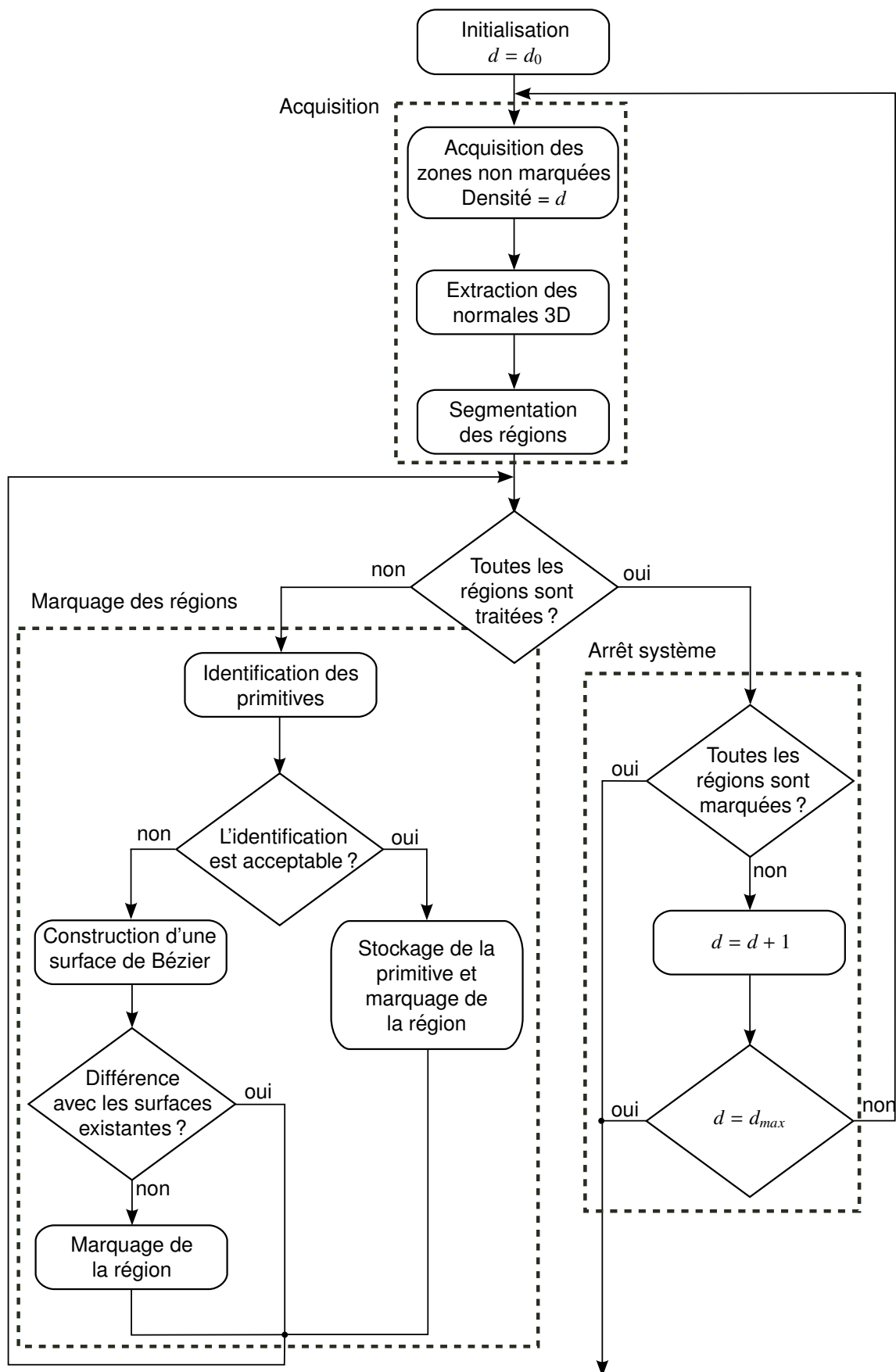


FIGURE 4.3 – Principe détaillé du système de numérisation dynamique.

La segmentation des images est une problématique qui a été, et qui est encore, très étudiée. La segmentation consiste à partitionner une image en plusieurs régions semblables. Pour cela, il est possible d'utiliser différents critères comme la texture, l'intensité, etc. Bien qu'il existe de nombreuses méthodes, aucune méthode ne permet de segmenter tous les types d'images. Les différentes méthodes de segmentation d'image peuvent se classer en plusieurs catégories (figure 4.4).

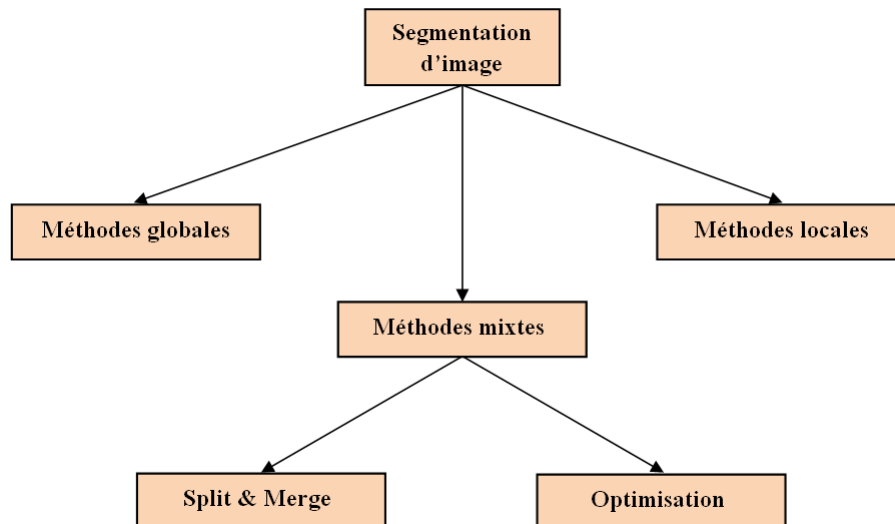


FIGURE 4.4 – Classement par catégories des différentes méthodes de segmentation d'images.

#### 4.2.1/ MÉTHODES GLOBALES

La méthode de segmentation globale la plus connue est la segmentation basée sur l'histogramme de l'image. Cette méthode est considérée comme globale au sens où la décision d'appartenance d'un pixel à une région se fait à partir de l'image entière. Le principe, présenté par Ohlander *et al.* [96], consiste à sélectionner itérativement le pic le plus significatif de l'histogramme et à conserver uniquement les points correspondant à cette zone. Les régions ainsi obtenues sont ensuite supprimées de l'image. L'algorithme continue ainsi jusqu'à ce que tous les objets de la scène soient segmentés ou considérés comme trop petits. Cette approche est particulièrement rapide et efficace sur des images couleurs ou des images contrastées.

### 4.2.2/ MÉTHODES MIXTES

Les méthodes mixtes sont des méthodes plus récentes qui ne fonctionnent pas de manière globale, ni même locale. Les techniques mixtes les plus connues sont le « Split and Merge » [97] et les méthodes d'optimisation type champs de Markov [98].

#### 4.2.2.1/ SPLIT AND MERGE

L'idée de ces méthodes est de partitionner l'image en petite région puis de les regrouper par la suite pour segmenter les différents objets. Le principe, illustré en figure 4.5, consiste à diviser récursivement l'image en parties égales tant qu'un critère d'homogénéité n'est pas satisfait. Cette étape permet de générer un graphe qui sera ensuite utilisé dans un second temps pour regrouper les zones qui se ressemblent. Ces méthodes permettent d'obtenir de bons résultats sur des formes géométriques, mais à cause du découpage en quad-tree, les régions ainsi obtenues sont généralement de formes carrées.

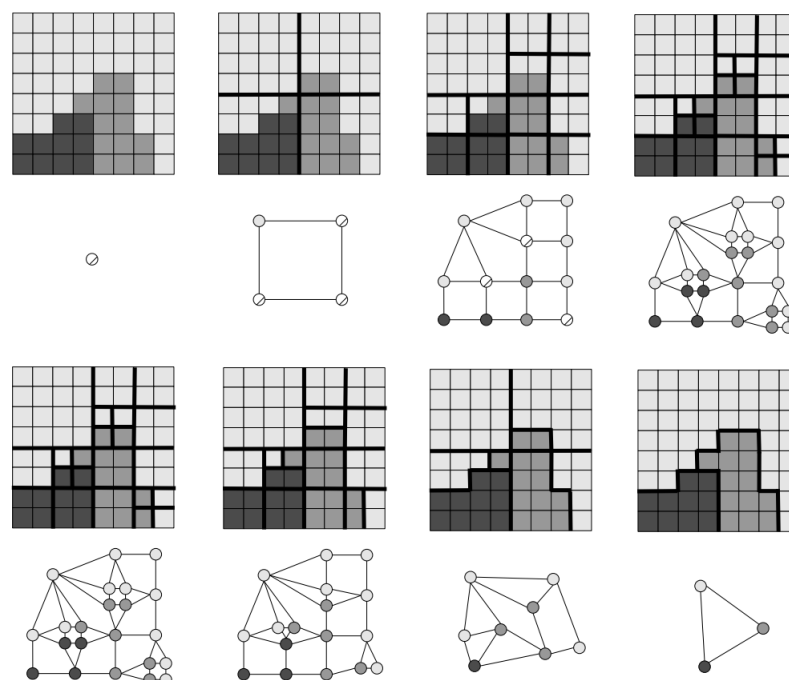


FIGURE 4.5 – *Principe du Split and Merge. La première étape (en haut) consiste à diviser l'image en plusieurs parties égales tant que la zone n'est pas homogène. La seconde partie (en bas) consiste à regrouper les graphes obtenus lors de la première étape afin de regrouper les zones qui se ressemblent.*



#### 4.2.2.2/ OPTIMISATION

Le principe des méthodes d'optimisation repose sur l'estimation d'une fonction  $f$  bidimensionnelle qui respecte certaines propriétés (constante par morceaux, régulière, ...) tout en restant proche de l'image analysée. Le but est de trouver un compromis entre ces différentes propriétés en minimisant une fonctionnelle de coût  $K$  qui dépend de l'image analysée, des contours, de la segmentation calculée et de la fonction  $f$  recherchée. L'une des fonctionnelles les plus connues est la fonctionnelle de Mumford-Shah [99] composée d'un terme de ressemblance à l'image d'origine, un terme de régularité sur chaque région et un terme de simplicité géométrique (longueur des frontières). On peut distinguer deux catégories de méthodes permettant de minimiser efficacement cette fonction de coût : les méthodes variationnelles (qui s'apparentent à des techniques de contours actifs) et les méthodes Markoviennes.

#### 4.2.3/ MÉTHODES LOCALES

Les méthodes locales les plus répandues en segmentation d'images sont sans conteste les méthodes basées sur la croissance de région. Le principe est identique à celui présenté en section 2.3.3.1. Une zone est initialisée à partir d'un point, puis tous les voisins respectant un certain critère sont ajoutés à la zone. Le choix du point, ainsi que l'ordre dans lequel les régions sont construites peuvent fortement influencer les résultats. Ces méthodes sont généralement très rapides et très efficaces sur les images qui présentent des zones homogènes (en intensité).

#### 4.2.4/ APPROCHE UTILISÉE

Dans notre cas, les images fournies par le scanner (figure 4.6-a) sont difficilement exploitables quelle que soit l'approche utilisée à cause de la projection de la texture aléatoire. Pour pallier ce problème, nous proposons d'utiliser la carte des discontinuités obtenue avec notre méthode présentée en partie 3.2 plutôt que les images fournies par le scanner (figure 4.6-b). L'image ainsi obtenue est composée de zones plus ou moins homogènes, ce qui permet d'utiliser les différentes méthodes présentées précédemment.

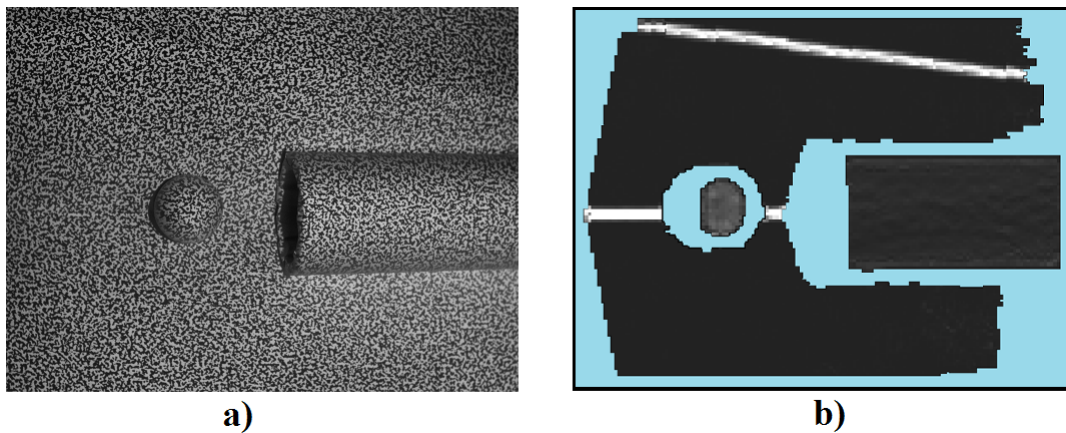


FIGURE 4.6 – *Calcul des coefficients de discontinuité depuis les images du scanner. a) Image de la caméra de référence. b) Coefficients de discontinuité.*

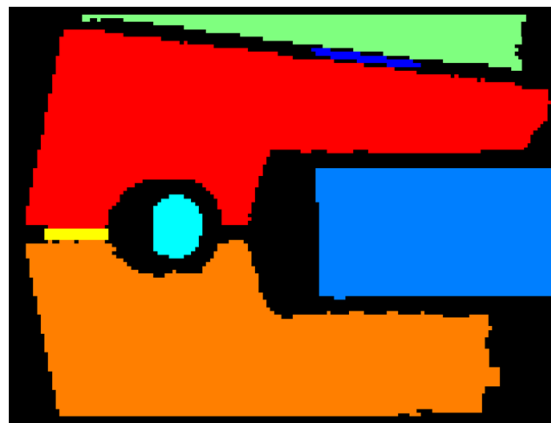


FIGURE 4.7 – *Segmentation des différents objets qui composent la scène à partir des coefficients de discontinuité.*

Les différentes régions étant très homogènes, la méthode la plus intuitive est l'approche basée croissance de région. Le seul problème de cette méthode est le choix des graines permettant d'initialiser la zone à faire croître. N'ayant pas de connaissances *a priori* sur la scène, nous proposons une méthode récursive assez basique. Le premier pixel non nul rencontré est choisi comme point d'initialisation. Une fois la croissance de région effectuée, la région obtenue est supprimée de l'image (tous les pixels formant cette région prennent une valeur nulle) et l'algorithme recommence avec une nouvelle graine. Le procédé s'arrête quand il ne reste plus que des pixels nuls dans l'image. En ne conservant que les zones de taille suffisante, nous sommes ainsi capables de segmenter les différents objets qui composent la scène. La figure 4.7 présente les résultats que nous obtenons avec une densité  $d = 1/3$  sur un objet test à partir des coefficients de discontinuité (figure 4.6). Chaque région ainsi obtenue est représentée par une couleur différente.

### 4.3/ EXTRACTION DES PRIMITIVES

Une fois la scène segmentée en plusieurs sous-parties, notre système cherche à identifier les primitives potentielles présentes dans la scène pour définir les zones à marquer, qui ne seront pas raffinées lors de l'itération suivante. L'identification des primitives se fait à partir de l'image gaussienne, comme présenté en partie 3.3. Dans la même partie, nous avons montré qu'il était possible d'identifier mais aussi de modéliser les primitives directement à partir des valeurs et vecteurs propres.

En pratique, la segmentation que nous avons effectuée depuis l'image à partir des coefficients de discontinuité ne sera jamais parfaite. La figure 4.8 illustre ce problème. Dans ce cas particulier, la déformation de la surface du cylindre (sur les bords) évolue lentement. De ce fait, l'algorithme de segmentation basée sur l'évolution des coefficients de discontinuité ne va pas segmenter correctement la primitive. C'est pourquoi nous proposons d'effectuer une nouvelle segmentation à partir de la primitive modélisée.

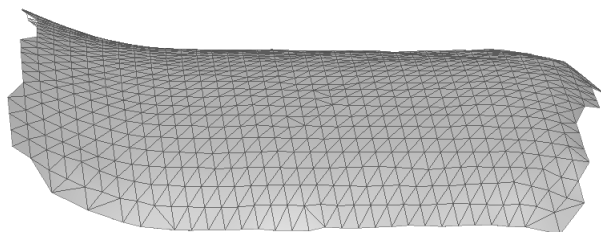


FIGURE 4.8 – *Partie d'un cylindre avec une variation de la surface localisée sur les bords.*

Les différents paramètres servant à modéliser la primitive ont été obtenus à partir des valeurs et vecteurs propres de la matrice de covariance, en supposant que le vecteur propre associé à la plus petite valeur propre était l'axe du cylindre. Cependant, nous avons précédemment démontré que l'axe du cylindre n'est pas tout à fait un vecteur propre de la matrice de covariance, ce qui induit inévitablement une légère erreur entre la primitive modélisée de cette façon et la primitive théorique. Afin de segmenter correctement, la primitive modélisée doit être optimale. C'est pourquoi nous proposons dans un premier temps d'ajuster la primitive obtenue sur les points 3D, c'est à dire minimiser la distance quadratique entre la primitive et les points, puis dans un second temps d'affiner la segmentation.

## 4.3.1/ AJUSTEMENT DES PRIMITIVES

A cause de l'écart qui existe entre l'axe de révolution de la primitive et le vecteur propre de la matrice de covariance, la primitive modélisée à partir de la sphère gaussienne est légèrement biaisée. Ce que nous proposons ici consiste à ajuster la primitive sur les points en prenant comme initialisation le modèle estimé depuis la sphère gaussienne.

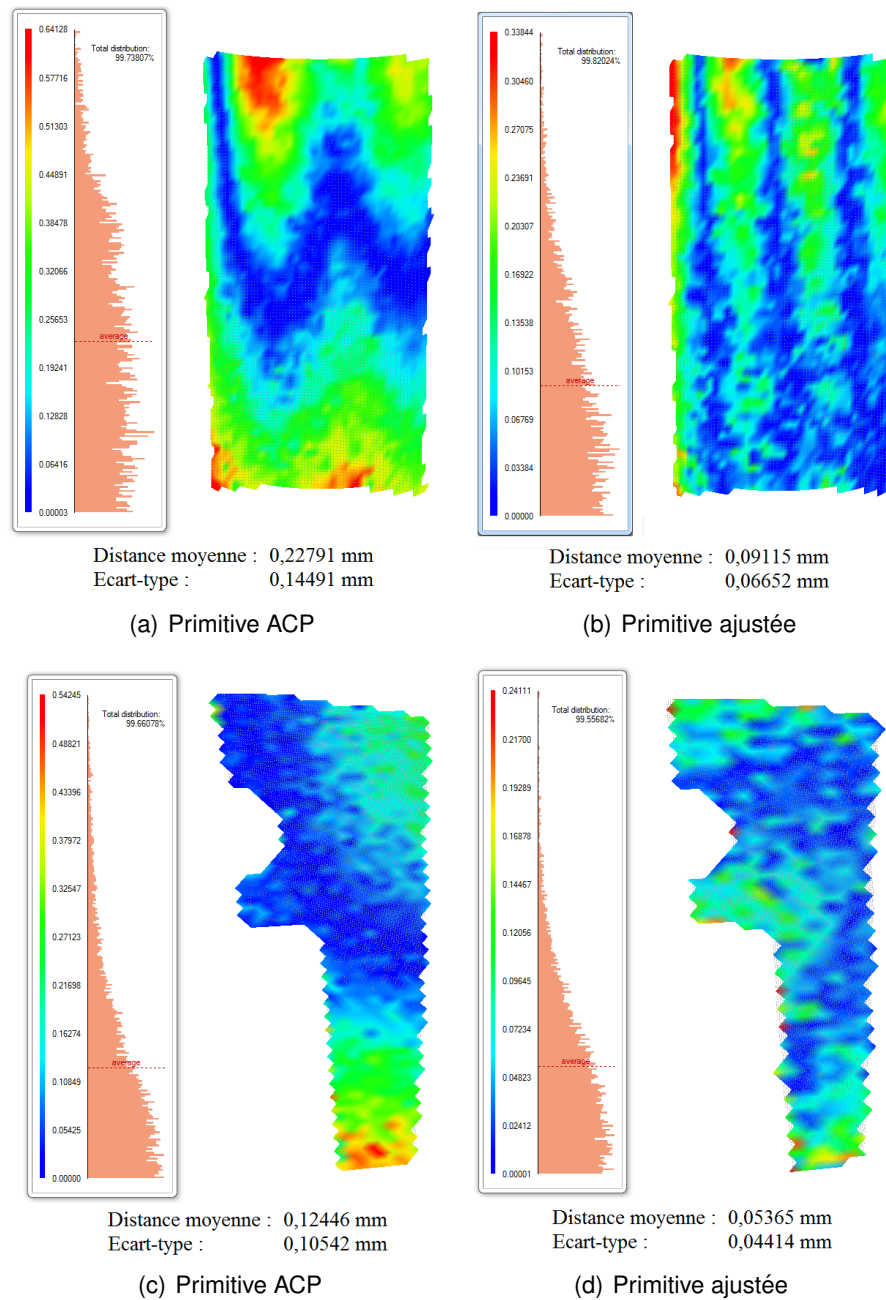


FIGURE 4.9 – Représentation de la distance entre la surface modélisée et les points 3D. a-b) Distance primitive/points 3D dans le cas d'un cylindre avant et après ajustement de la primitive sur les points. c-d) Distance primitive/points 3D dans le cas d'un plan.

Alistair Forbes [100] propose une méthode permettant d'ajuster une forme géométrique sur des données tridimensionnelles par la méthode de Gauss-Newton [90]. L'algorithme de Gauss-Newton étant une méthode de résolution des problèmes de moindres carrés non linéaire, le temps de calcul pour converger vers une solution peut parfois s'avérer assez long, or la contrainte du temps de calcul est un critère important dans un système dynamique. Dans le but de réduire le temps de calcul, nous proposons d'utiliser la primitive obtenue avec l'ACP sur les données de la sphère gaussienne comme forme géométrique initiale de l'algorithme. La primitive modélisée étant très proche de celle recherchée, il est possible de converger vers le résultat optimal en quelques itérations seulement. La figure 4.9 illustre l'évolution de la distance orthogonale entre les points 3D et la surface de la primitive modélisée avant et après ajustement par la méthode présentée par Alistair Forbes [100]. On peut constater une véritable amélioration suite à l'ajustement de la primitive.

#### 4.3.2/ AFFINAGE DE LA SEGMENTATION

L'identification de la primitive se fait à partir d'une analyse statistique sur des objets préalablement segmentés. De ce fait, même si la segmentation n'est pas parfaite, il est malgré tout possible d'extraire et de modéliser une primitive à partir du moment où celle-ci est représentée par une grande majorité des points. La figure 4.10 illustre deux cas où il est possible d'extraire une primitive malgré la présence d'éléments mal segmentés n'appartenant pas à la primitive (les bordures dans le cas du cylindre et la présence de rivets à tête ronde sur un plan). On constate également sur cette même figure que la distance entre la primitive modélisée et les points 3D devient importante dans les zones mal segmentées.

Nous proposons de segmenter les objets pour lesquels la distance entre la surface et les points 3D est supérieure à une certaine tolérance. Ainsi, ne seront conservés que les points 3D appartenant effectivement à la primitive. Les zones ainsi re-segmentées formeront alors de nouvelles régions qui seront raffinées lors de l'itération suivante du système.

Toutefois, un problème persiste avec cette approche. En effet, à aucun moment nous ne vérifions que la primitive estimée sur les données est correcte. Supposons un sous-ensemble de points  $\mathcal{N}(p)$  identifié comme appartenant à une primitive. Après ajustement de la primitive, tous les points dont la distance à la surface est supérieure à une tolérance fixée sont alors rejetés de la primitive. Cependant, dans le cas où le ratio entre le nombre de points rejetés et le nombre de points initiaux est important, il est alors judicieux de penser que la primitive identifiée ne correspond pas réellement aux données. Ainsi, pour

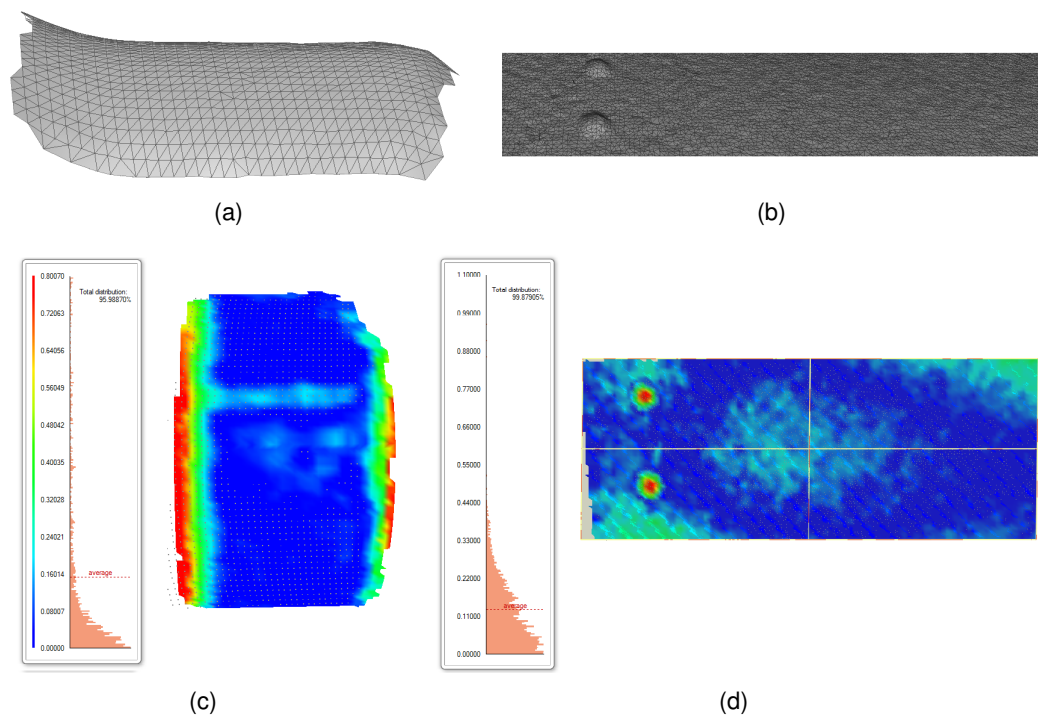


FIGURE 4.10 – Distance entre les points 3D et la primitive modélisée sur des objets mal segmentés. a-b) Objets mal segmentés. c-d) Distance entre la surface de la primitive et les points 3D respectivement pour les objets a) et b).

résoudre ce problème nous proposons d'ajouter un test qui permettra de vérifier si la primitive supposée est correcte ou non. Ce test sera effectué sur le ratio  $r$  :

$$r = \frac{N_r}{N_i}, \quad (4.2)$$

avec  $N_r$  le nombre de points rejetés (points dont la distance à la surface est supérieure à une certaine tolérance) et  $N_i$  le nombre de points qui composent la primitive. La question qui se pose alors est la suivante : quel ratio choisir pour décider si une primitive est correcte ou non ?

- Dans le cas où le ratio  $r$  est important (on autorise peu d'outliers), en cas de mauvaise segmentation (figure 4.10) il y a de fortes chances que la primitive soit rejetée. De ce fait, l'algorithme ne marquera pas la zone qui sera alors raffinée. Autrement dit, un seuil trop important rend notre système de numérisation moins performant en cas de mauvaise segmentation.
- A contrario, si le ratio est faible (beaucoup d'outliers) il y a un risque d'identifier une région par une primitive qui ne correspond pas aux données et du coup de sur-segmenter cette zone. Un seuil trop faible implique donc une erreur potentielle sur la primitive ainsi qu'une sur-segmentation de l'objet.



Le choix du ratio  $r$  dépend fortement de l'application visée. Si l'application finale consiste à obtenir un modèle précis, il faut éviter au maximum les erreurs sur l'identification des primitives. Inversement, si l'objectif consiste à obtenir un nuage peu dense ou de modéliser l'objet par un maximum de primitives, il faudra adapter le ratio  $r$  afin de ne pas rejeter systématiquement les primitives.

#### 4.3.3/ SYNTHÈSE

Le système de numérisation dynamique présenté jusqu'à présent procède de la façon suivante. Il calcule des points 3D depuis les images du scanner pour une densité  $d = 1/i$ , puis extrait ensuite des informations complémentaires afin d'identifier des primitives. S'il existe des primitives, celles-ci sont alors ajustées afin de vérifier que la primitive identifiée correspond bien aux données. Dans ce cas, une nouvelle segmentation est effectuée si besoin afin de ne conserver que les points correspondant réellement à la primitive identifiée. Le système est ensuite re-bouclé et calcule de nouveau des points 3D pour une densité  $d = 1/(i-1)$  excepté dans les zones marquées comme appartenant à une primitive lors de l'itération précédente. Le système suit ce procédé récursif jusqu'à atteindre la densité maximale  $d = 1/1$  ou jusqu'à ce qu'une condition d'arrêt soit atteinte.

#### 4.4/ SURFACES PARAMÉTRIQUES

Bien que le système s'applique avant tout à des pièces manufacturées, celles-ci ne sont pas systématiquement composées de primitives géométriques simples. Dans ce cas, le système ne sera pas en mesure d'identifier des primitives et ne marquera alors aucune zone, rendant ainsi notre système inefficace. Comme mentionné précédemment, pour résoudre ce problème et ne pas limiter notre système aux seules primitives géométriques, il convient d'en introduire de nouvelles. N'ayant pas d'*a priori* sur la forme de l'objet, nous avons choisi d'étudier des primitives pouvant s'adapter à toutes les formes : les surfaces paramétriques.

Le but est d'approximer des surfaces paramétriques sur les régions n'étant pas associées à une primitive puis de comparer l'évolution de ces surfaces entre deux itérations successives de notre système. Si la surface a évolué entre deux itérations, le fait de calculer de nouveaux points ayant permis de raffiner la région, il est pertinent de continuer à raffiner la zone concernée. A l'inverse, si la surface d'une région n'évolue plus, calculer d'avantage de points est inutile. Dans ce cas, la zone est alors marquée et ne sera plus traitée par la suite.

## 4.4.1/ SURFACES DE BÉZIER

Il existe différentes surfaces paramétriques souvent utilisées en 3D. Les plus connues sont les surfaces de Bézier [101], les B-splines et les NURBS [54]. Notre choix s'est porté sur les surfaces de Bézier, qui sont souvent utilisées en conception assistée par ordinateur (CAO). A noter qu'il est tout à fait possible de remplacer les surfaces de Bézier par des B-splines ou des NURBS. Cependant, dans notre application les surfaces de Bézier étant largement suffisantes nous avons opté pour celles-ci.

## 4.4.1.1/ COURBES DE BÉZIER

Les surfaces de Bézier ont été inventées par Pierre Bézier [102, 103] afin de modéliser des pièces de carrosseries automobiles. Ces surfaces sont composées de courbes de Bézier (figure 4.11), qui sont des courbes polynomiales paramétriques permettant de faire une approximation entre  $n$  points.

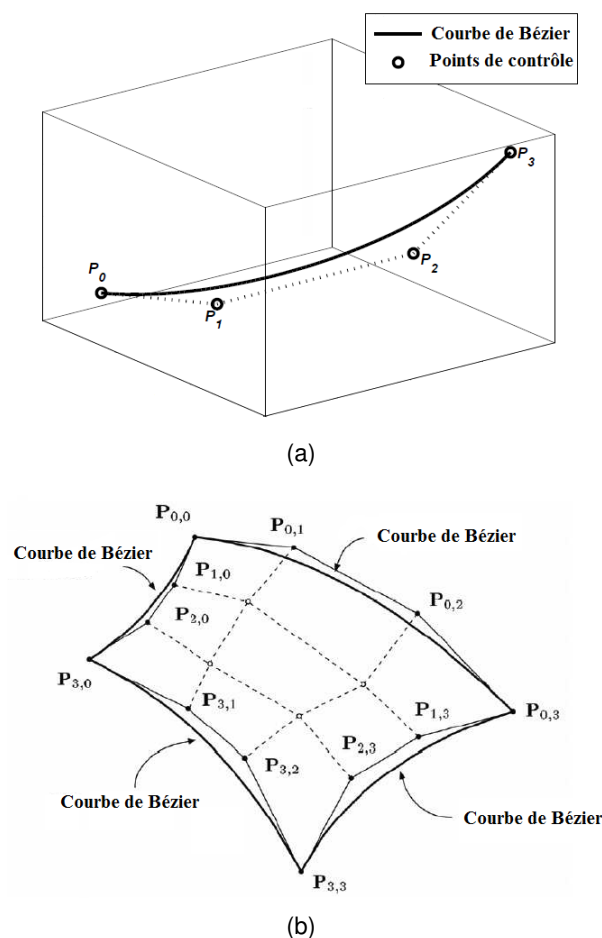


FIGURE 4.11 – Représentation tridimensionnelle des courbes et carreaux de Bézier. a) Courbe de Bézier. b) Carreau de Bézier cubique.



Les courbes de Bézier d'ordre  $n$  sont définies par  $n + 1$  points de contrôle de la façon suivante :

$$C(t) = \sum_{i=0}^n B_i^n(t) \cdot P(i), \quad (4.3)$$

avec  $t \in [0, 1]$  et où les  $B_i^n$  sont les polynômes de Bernstein. Dans la plupart des cas industriels, les courbes de Bézier sont d'ordre 3, cubiques ( $n = 3$ ). Dans ce cas, on peut écrire la courbe sous la forme paramétrique suivante :

$$P(t) = P_0(1 - t)^3 + 3P_1t(1 - t)^2 + 3P_2t^2(1 - t) + P_3t^3 \quad (4.4)$$

Une surface de Bézier cubique (ordre 3) est composée d'un ou plusieurs carreaux de Bézier, qui sont eux mêmes obtenus à l'aide des courbes de Bézier. Un carreau de Bézier cubique (figure 4.11) est composé de  $4 \times 4$  points de contrôle  $P_{i,j}$ , avec  $0 \leq i \leq 3$  et  $0 \leq j \leq 3$ . Il est défini par l'équation suivante :

$$Q(u, v) = \sum_{j=0}^3 \sum_{i=0}^3 P_{i,j}(u) \cdot B_{i,3}(u) \cdot B_{j,3}(v), \quad (4.5)$$

avec  $u$  et  $v \in [0, 1]$ ,  $B_{i,3}$  et  $B_{j,3}$  les polynômes de Bernstein d'ordre 3 et  $P_{i,j}$  les points de contrôle. On constate que si l'on connaît les points de contrôle, on est alors capable de résoudre l'équation (4.5) et ainsi de déterminer un carreau de Bézier entre ces différents points.

#### 4.4.1.2/ ESTIMATION DES POINTS DE CONTRÔLE

En général, l'approche classique pour construire un carreau de Bézier consiste à utiliser des points 3D comme points de contrôle [104]. Cependant, en procédant de cette façon, on privilégie certains points par rapport à d'autres étant donné que les courbes de Bézier ne passent pas par les points de contrôle. Pour pallier ce problème, on retrouve parfois une méthode 2D, dérivée de l'algorithme de De Casteljau [105], permettant d'estimer des points de contrôle à partir de seulement deux points et de leur normale à la surface associée à chaque point. La figure 4.12 schématise la méthode permettant d'obtenir les points de contrôle  $P_1$  et  $P_2$  à partir des points  $P_0$  et  $P_3$  ainsi que des normales  $n_0$  et  $n_3$  pour une courbe quelconque  $C$ .

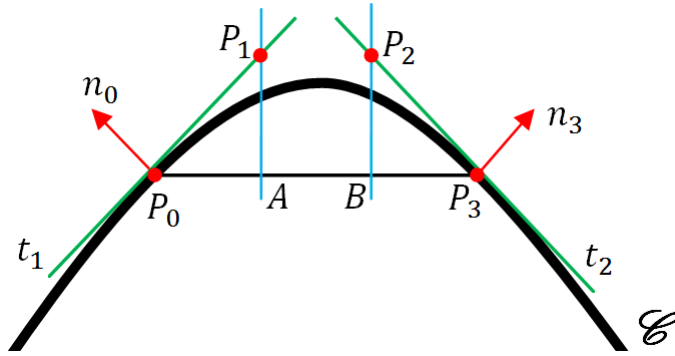


FIGURE 4.12 – Estimation des points de contrôle d'une courbe de Bézier d'ordre 3.

Connaissant les couples  $\{P_0, n_0\}$  et  $\{P_3, n_3\}$  on peut calculer les tangentes  $t_1$  et  $t_2$  (en vert sur la figure 4.12) aux points  $P_0$  et  $P_3$ . De même, on peut calculer le segment  $[P_0, P_3]$  et déterminer les points  $A$  et  $B$  espacés régulièrement sur le segment  $[P_0, P_3]$  de la façon suivante :

$$A(x, y) = P_0(x, y) + \frac{P_3(x, y) - P_0(x, y)}{3}, \quad (4.6)$$

$$B(x, y) = P_0(x, y) + \frac{2 \times P_3(x, y) - P_0(x, y)}{3}. \quad (4.7)$$

Il est alors possible d'estimer le point de contrôle  $P_1$  comme l'intersection de la tangente  $t_1$  et de la droite perpendiculaire à  $[P_0, P_3]$  au point  $A$  (en cyan sur la figure 4.12). La même méthode est applicable pour le point  $P_2$  à partir de la tangente  $t_2$  et de la perpendiculaire à  $[P_0, P_3]$  au point  $B$ . Par analogie, nous avons implémenté cette méthode en l'adaptant à la 3D. La figure 4.13 illustre la méthodologie.

- On cherche les points de contrôle permettant d'approximer la courbe en rouge (figure 4.13-a) à partir de deux points  $P_0$  et  $P_3$ , ainsi que de leur normale à la surface associée  $n_0$  et  $n_3$  (en noir). Connaissant les points et les normales, on peut alors définir deux plans tangents à la surface  $\Pi_1$  et  $\Pi_2$  (plans verts).
- L'étape suivante consiste à déterminer la position des points  $A$  et  $B$  (figure 4.13-b). Pour cela, on procède de la même façon qu'en 2D, en prenant deux points à équidistance sur le segment  $(P_0, P_3)$  (symbolisés par les cercles rouges). Il est alors possible de définir deux nouveaux plans perpendiculaires au segment  $(P_0, P_3)$  et passant respectivement par les points  $A$  et  $B$  (plans bleus).

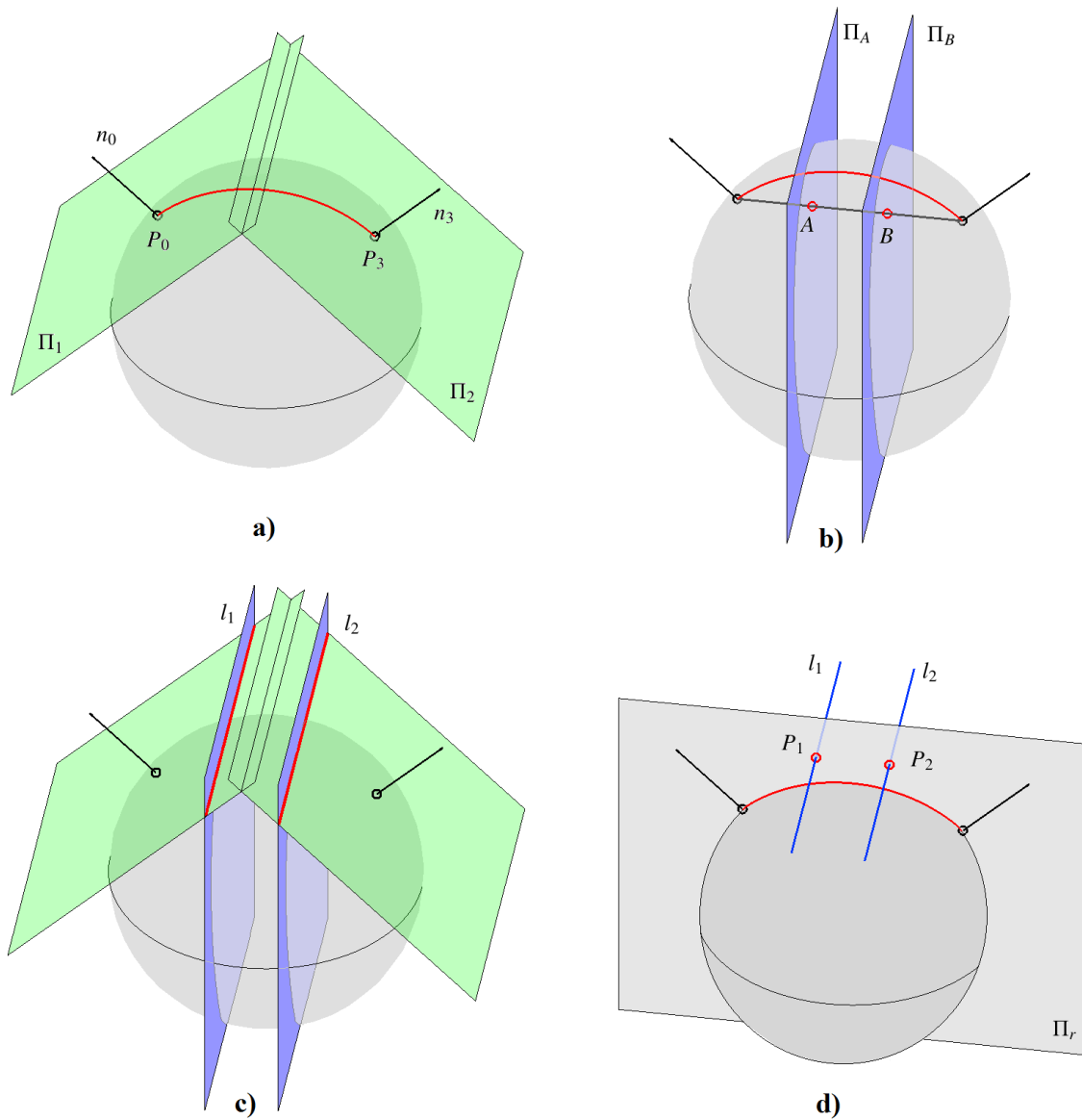


FIGURE 4.13 – Estimation des points de contrôle en 3D. a) Plans tangents à la surface aux points  $P_0$  et  $P_3$ . b) Plans orthogonaux ( $P_0, P_3$ ) et passant par les points  $A$  et  $B$ . c) Les intersections des plans forment deux lignes  $l_1$  et  $l_2$ . d) Les intersections des lignes avec le plan  $\Pi_r$  forment les points de contrôle  $P_1$  et  $P_2$ .

- Une fois ces quatre plans connus, il est alors possible de calculer les intersections de ces plans (figure 4.13-c). On obtient ainsi deux lignes  $l_1$  et  $l_2$  (en rouges), l'une pour l'intersection des plans  $\Pi_1$  et  $\Pi_A$  et l'autre pour l'intersection des plans  $\Pi_2$  et  $\Pi_B$ .
- Les points de contrôle  $P_1$  et  $P_2$  (les cercles rouges sur la figure. 4.13-d) sont finalement obtenus en calculant l'intersection entre les lignes  $l_1$  et  $l_2$  et le plan  $\Pi_r$  (plan gris) orthogonal à  $\Pi_A$  et  $\Pi_B$  et passant par  $P_0, A, B$  et  $P_3$ .

Avec cette approche, il est ainsi possible de définir deux points de contrôle uniquement à partir de données connues : les points 3D et leur normale à la surface. Il est ainsi possible d'approximer un carreau de Bézier à partir de quatre points et de leur normale.

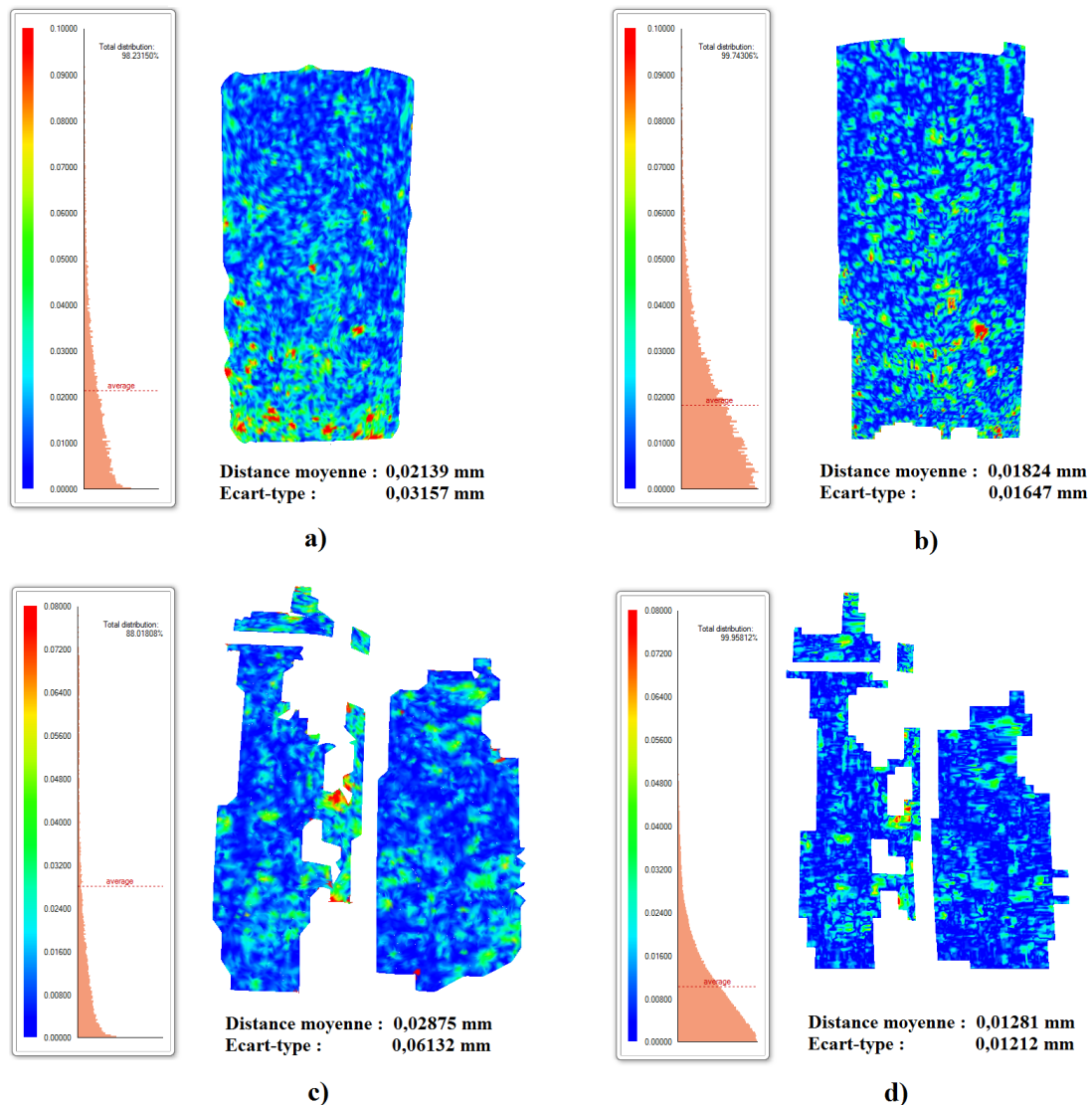


FIGURE 4.14 – Représentation de l'erreur entre un nuage de points dense et des surfaces paramétriques obtenues depuis un nuage de points de densité  $d = 1/5$ . a) Surface estimée sur un cylindre avec le logiciel RapidForm. b) Surface de Bézier avec prise en compte des normales sur le même cylindre. c) Surface RapidForm sur des objets non identifiés. d) Surface de Bézier sur ces mêmes objets.

L'intérêt des surfaces de Bézier réside dans l'utilisation de points de contrôle qui peuvent être calculés à partir des normales 3D, ce qui permet d'obtenir une surface plus fidèle qu'en utilisant d'autres méthodes ne prenant pas en compte les normales, comme le montre la figure 4.14. Un second intérêt, toujours lié à l'utilisation des normales pour

définir les points de contrôle, se situe au niveau de la continuité entre plusieurs carreaux. Du fait que l'on utilise les normales pour calculer les points de contrôle, la surface ainsi obtenue sera continue, ce qui n'est pas toujours le cas autrement.

#### 4.4.1.3/ DÉCOMPOSITION DES OBJETS

Nous venons de voir qu'en connaissant quatre points 3D et leurs normales à la surface, il est possible de calculer un carreau de Bézier. Pour représenter un objet en intégralité avec des carreaux de Bézier, il faut donc décomposer celui-ci en plusieurs quadrilatères. Ce type de décomposition peut parfois s'avérer assez complexe suivant la forme des régions. Roseline Bènière [106] propose une approche permettant d'obtenir un maillage quadrangulaire à partir d'un maillage triangulaire, ce qui permet de calculer des surfaces paramétriques à partir du maillage ainsi obtenu. Cette méthode pourrait fonctionner dans notre cas, mais elle impliquerait dans un premier temps de mailler le nuage de points puis de le re-mailler pour obtenir un maillage quadrangulaire. De plus, cette approche serait trop coûteuse en temps de calcul car on aurait alors énormément de carreaux. La solution que nous proposons s'inspire d'une décomposition de l'objet en maillage quadrangulaire. Nous proposons de définir une grille connue dans l'image et de l'adapter aux données (figure 4.15). Dans l'image, en adaptant la grille aux points 2D les plus proches, il est possible de décomposer un objet quelconque en plusieurs quadrilatères et ainsi approximer une surface de Bézier sur l'objet étudié.

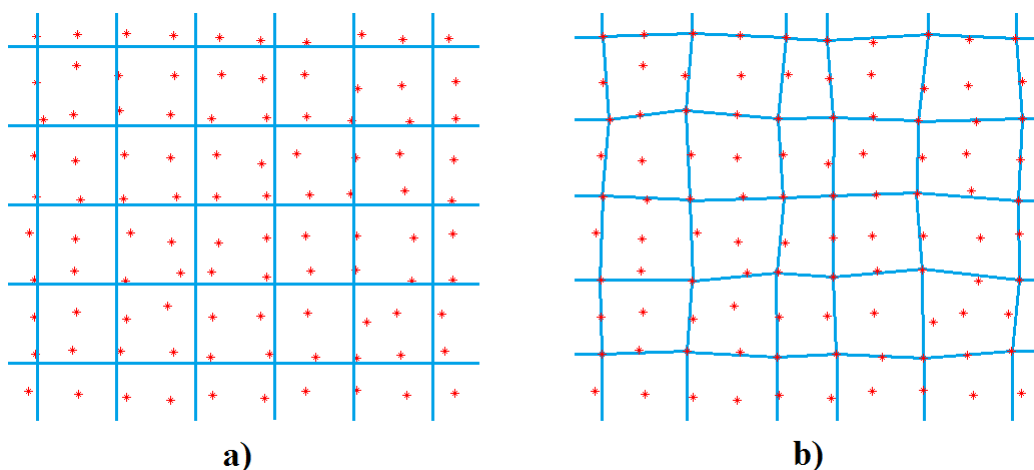


FIGURE 4.15 – Grille utilisée pour décomposer les objets en quadrilatères. a) Grille régulière générée dans l'image. b) Déformation de la grille en fonction des données.

#### 4.4.2/ TEST DES SURFACES

Comme nous l'avons présenté dans la partie précédente, lorsque qu'une primitive est identifiée nous calculons la distance entre les données et la surface de la primitive modélisée depuis l'ensemble des données. Concernant les surfaces paramétriques, il n'est pas pertinent d'appliquer la même approche car les surfaces étant extraites localement, la distance entre les points et la surface de Bézier reste toujours faible. Nous proposons de calculer la distance entre deux surfaces obtenues à deux itérations différentes, et donc à différentes densités de points. Pour ce faire, nous procédons de la même façon que dans le cas des primitives. Voici la procédure pour un objet quelconque :

- Première itération  $d = 1/i$  : Une surface de Bézier est approximée sur un objet.
- Deuxième itération  $d = 1/(i - 1)$  : Si l'objet n'a pas été identifié comme une primitive, on approxime alors une nouvelle surface de Bézier.
- Calcul de la distance entre les deux surfaces de Bézier obtenues à deux itérations différentes.
- En acceptant une certaine tolérance  $\tau$  sur la distance entre les deux surfaces, on calcule alors le ratio  $r$  entre le nombre de points dont la distance est inférieure à  $\tau$  et le nombre de points dont la distance est supérieure à  $\tau$ .
- Si le ratio  $r$  est inférieur à un certain seuil, on considère que la surface n'a que peu évolué entre deux acquisitions et que le fait de calculer d'avantage de points sur cet objet n'a pas apporté d'informations supplémentaires. Dans ce cas, on admet que la zone est suffisamment échantillonnée et qu'il n'est pas nécessaire d'ajouter de l'information.
- Dans le cas contraire, on considère que la surface a évolué et que le calcul de nouveaux points est utile. Cette étape sera alors répétée à la prochaine itération.

#### 4.5/ INCRÉMENTATION ET ARRÊT DU SYSTÈME

Une fois que toutes les régions segmentées ont été analysées, le système est soit rebouclé afin de raffiner toutes les régions non marquées, soit arrêté. L'arrêt du système ne se fait que dans deux cas distincts : lorsque toutes les zones sont marquées ou bien lorsque la densité  $d$  est égale à  $d_f$ . Si l'une de ces conditions est respectée alors le système s'arrête, sinon le système est incrémenté et toutes les étapes sont réitérées avec une densité de points  $d$  plus importante pour les zones non marquées :

$$d = \frac{1}{i - 1}. \quad (4.8)$$

Lors de l'itération suivante, afin de ne tenir compte que des zones non marquées, nous appliquons un masque sur la grille de germes servant à l'appariement des points afin de ne conserver que les germes situés sur des zones non marquées (figure 4.16).

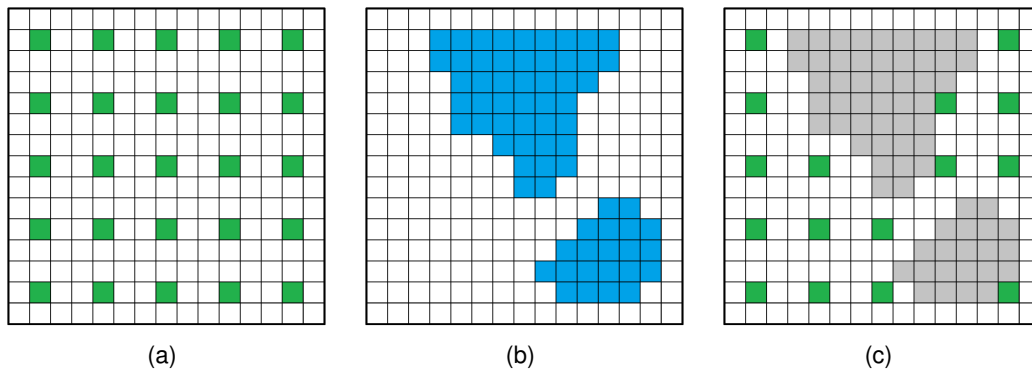


FIGURE 4.16 – *Positionnement des graines servant à la mise en correspondance. a) Grille de germes. b) Zones marquées lors de cette itération. c) Grille de germes modifiée.*

La grille ainsi obtenue est alors utilisée pour appairer les points lors de l'itération suivante, ce qui permet de ne pas raffiner les zones marquées.

## 4.6/ CONCLUSION

Dans cette partie, nous avons présenté le principe du fonctionnement du système de numérisation dynamique étape par étape. Le principe du système consiste à adapter l'acquisition à la complexité de la pièce à numériser. Pour ce faire, le système se décompose en 3 étapes majeures : l'acquisition, le marquage des régions et l'arrêt ou l'incrémentation du système. L'objectif d'un tel système est de fournir des résultats proches de ceux que l'on obtiendrait en passant par la chaîne 3D classique. C'est pourquoi, dans la partie suivante nous présentons les résultats obtenus par ce système étape par étape sur différents objets. Ces résultats seront ensuite comparés aux résultats obtenus avec des méthodes de simplification de maillage.

## RÉSULTATS

Jusqu'à présent, nous avons présenté les différentes parties qui composent notre système, ainsi que son principe de fonctionnement. Dans cette partie, sont présentés les résultats obtenus avec ce même système de numérisation dynamique sur différents objets industriels. Dans un premier temps, nous présenterons les résultats obtenus après chacune des étapes, puis dans un second temps, nous étudierons l'influence des différents paramètres réglables présentés précédemment sur le résultat final.

Par la suite, les résultats obtenus avec ce système seront comparés aux résultats qu'il est possible d'obtenir en passant par la chaîne 3D classique : acquérir un maximum de points, mailler les données, remailler les données pour obtenir un maillage 2-manifold (une arête ne partage pas plus de deux triangles) puis simplifier le maillage en préservant au maximum les zones saillantes.

### 5.1/ OUTIL LOGICIEL

Afin de simplifier l'utilisation du système développé, nous avons développé un outil logiciel permettant de régler les différents paramètres présentés en amont à l'aide d'une interface présentée en figure 5.1. Cette interface permet de régler très simplement les différents paramètres souhaités. L'interface est également munie d'une fenêtre permettant d'afficher les différents résultats présentés dans ce manuscrit (primitives, surfaces, segmentation), facilitant ainsi l'analyse des données.

L'interface dispose de deux modes distincts : le mode *automatique* et le mode *pas à pas*. Le mode *pas à pas* permet d'afficher les résultats obtenus à chaque étape. Ce mode est particulièrement utile pour optimiser l'acquisition en raffinant les différents paramètres en fonction de l'application visée. Le second mode, le mode *automatique* affiche uniquement



que le résultat final, ce qui permet d'obtenir un résultat final très rapidement.

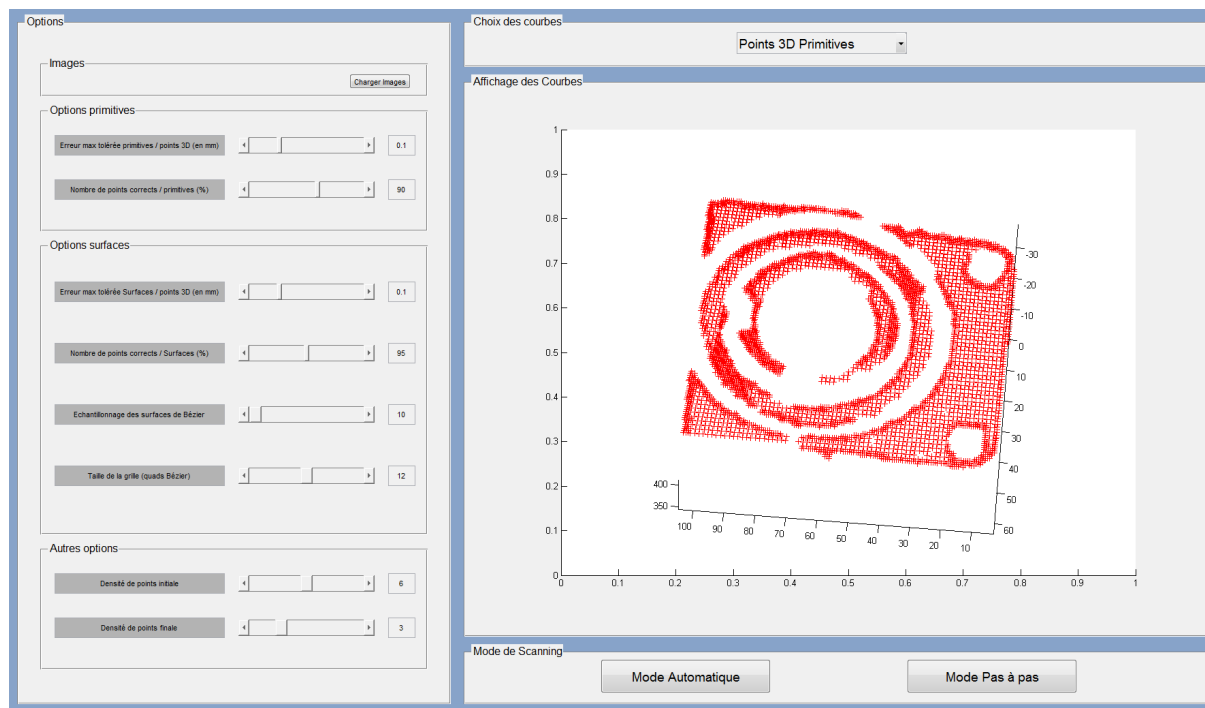
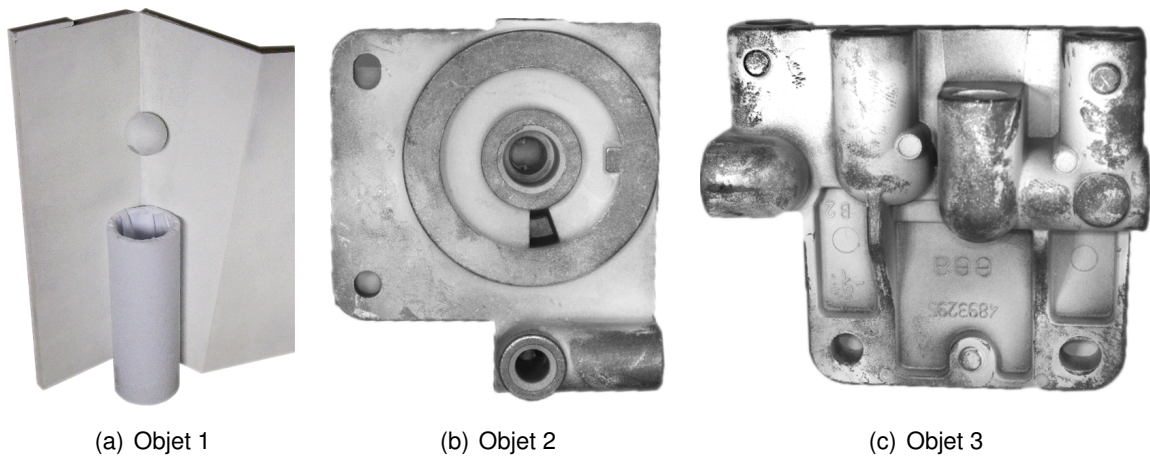


FIGURE 5.1 – Interface graphique du système développé.

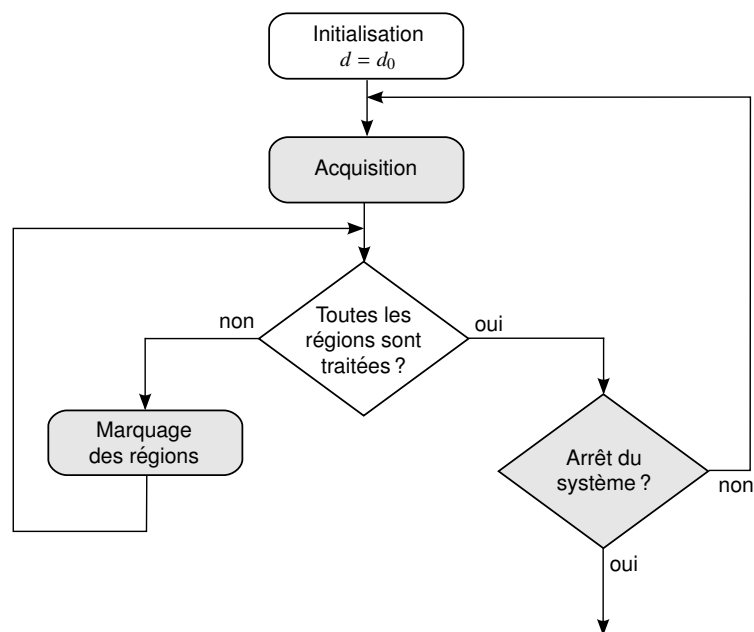
## 5.2/ RÉSULTATS ÉTAPE PAR ÉTAPE

La condition principale pour obtenir une bonne simplification des données avec notre système repose sur la présence de primitives géométriques dans la scène. De ce fait, nous avons choisi de présenter les résultats sur trois objets ayant des caractéristiques bien différentes (figure 5.2). Le premier objet est composé exclusivement de primitives géométriques simples, ce qui devrait permettre de fortement simplifier les données. Le deuxième objet, quant à lui, est un objet manufacturé assez simple présentant des primitives (plans), des trous, ainsi qu'une zone plus complexe (un coude). Enfin, le dernier objet que nous présentons est plus complexe que les deux précédents, il y a moins de primitives géométriques simples, plus de discontinuités et plus de zones courbes ne pouvant pas être identifiées comme des primitives.

Afin de bien comprendre le résultat final, nous présenterons les résultats étape par étape, selon le logigramme simplifié présenté en figure 5.3. On distingue trois étapes distinctes (en gris sur la figure 5.3) : l'acquisition qui consiste à extraire des informations complémentaires et à segmenter les données, le marquage des régions permettant de

FIGURE 5.2 – *Objets utilisés pour tester notre système de numérisation.*

déterminer si une zone doit être raffinée et enfin l'arrêt ou l'incrémentation du système.

FIGURE 5.3 – *Principe du système de numérisation dynamique.*

### 5.2.1/ ACQUISITION

La partie acquisition regroupe les premières étapes de la méthode, à savoir le calcul des normales 3D, des discontinuités et la segmentation des différents objets qui composent la scène. Sans rentrer dans le détail de la méthode, présentée dans la partie précédente, nous exposons ici les résultats obtenus lors de cette étape pour les trois objets test présentés en figure 5.2. Pour ces trois objets, nous avons choisi les paramètres suivants :

- Distance tolérée entre la primitive/surface et les points 3D = 0.1 mm.
- Ratio primitive/surface = 90%.
- Densité initiale  $d_0 = 1/5$ .
- Densité finale  $d_f = 1/2$ .

Une fois le système initialisé  $d = d_0$ , la première étape consiste donc à calculer les normales et les discontinuités, afin de segmenter les différents objets à partir de l'image à l'aide des méthodes présentées dans la partie précédente. Bien que ces différentes étapes n'utilisent pas d'informations 3D, les résultats de l'étape d'acquisition (normales, coefficients de discontinuité, segmentation) seront représentés sur le nuage de points pour un meilleur rendu visuel. Les figures 5.4, 5.5 et 5.6 présentent les résultats que nous obtenons lors de la première étape pour les différents objets.

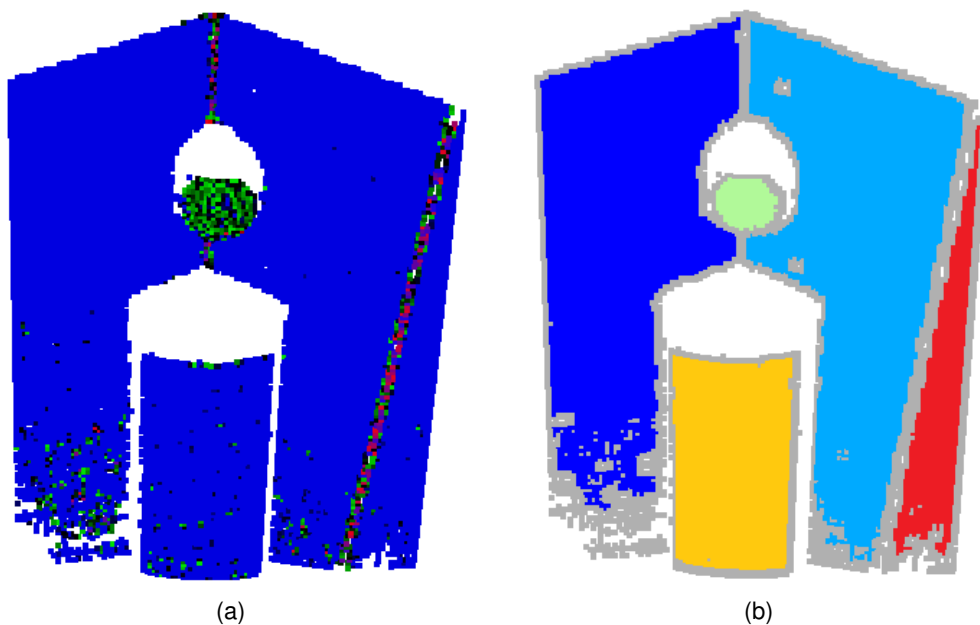


FIGURE 5.4 — Première étape de notre système pour l'objet 1. a) Coefficients des discontinuités plaqués sur les points 3D. b) Segmentation des objets : les points gris représentent les points non segmentés.

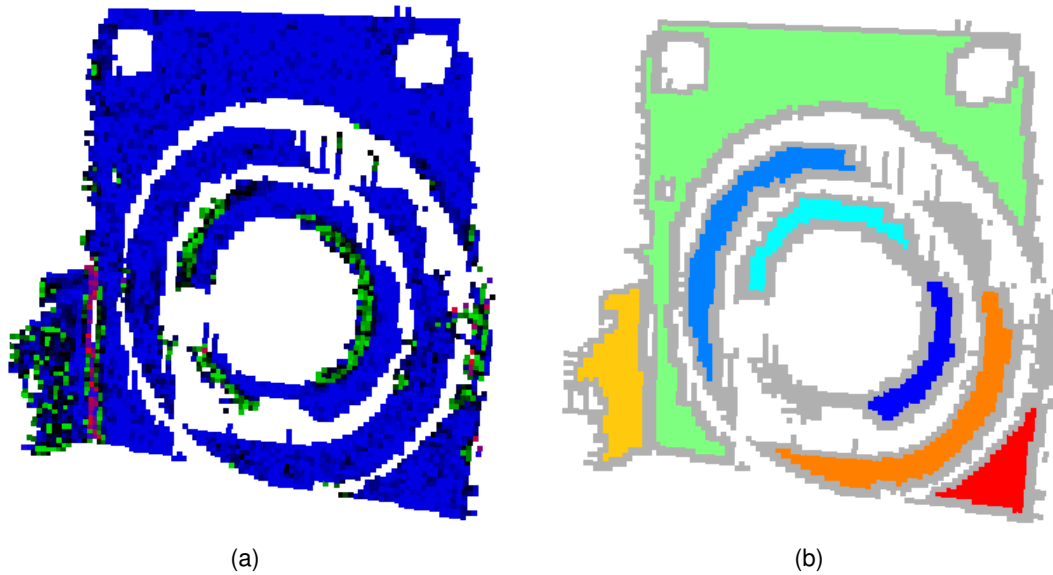


FIGURE 5.5 – Première étape de notre système pour l'objet 2. a) Coefficients des discontinuités plaqués sur les points 3D. b) Segmentation des objets : les points gris représentent les points non segmentés.

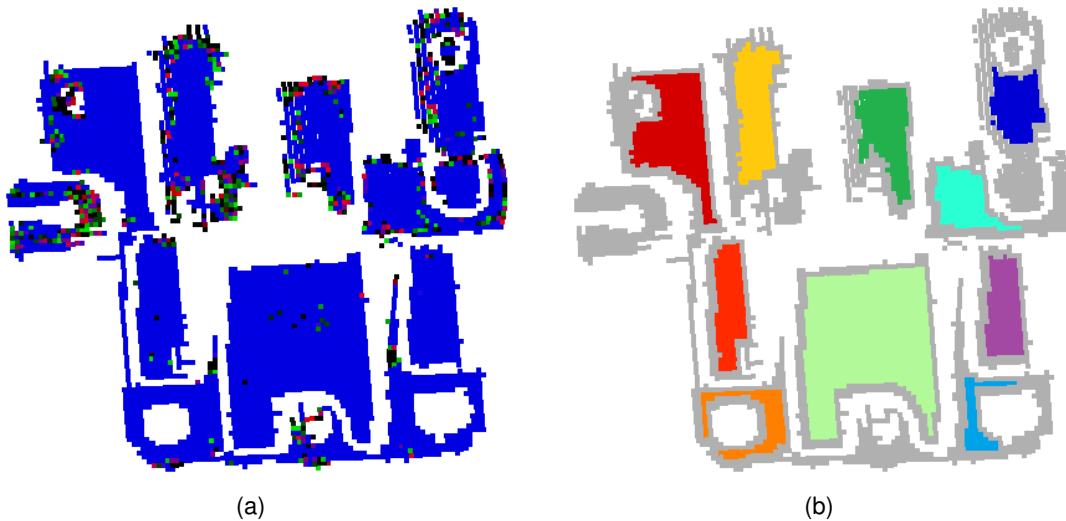


FIGURE 5.6 – Première étape de notre système pour l'objet 3. a) Coefficients des discontinuités plaqués sur les points 3D. b) Segmentation des objets : les points gris représentent les points non segmentés.

### 5.2.2/ MARQUAGE DES RÉGIONS

Une fois les différentes zones segmentées, l'étape suivante consiste à marquer les différentes zones. Pour cela, les valeurs propres de la matrice de covariance sont calculées à partir de la représentation sur la sphère gaussienne de chaque zone. Les figures 5.7, 5.8 et 5.9 illustrent, pour les trois pièces testées, la représentation gaussienne de différentes zones segmentées.

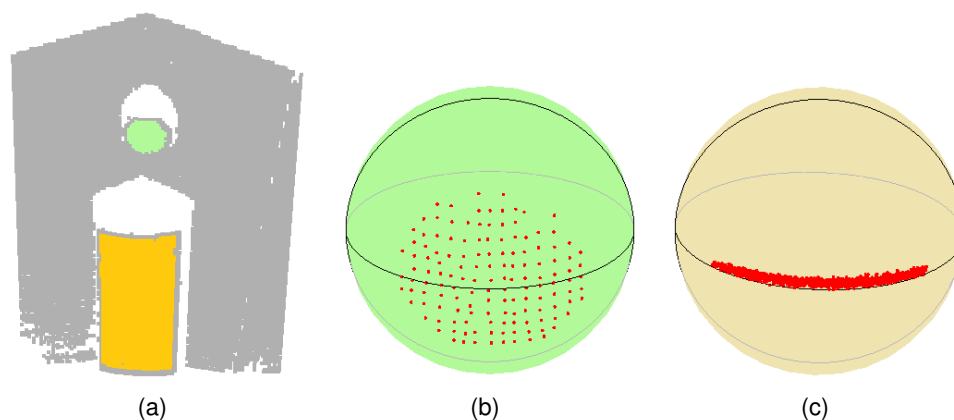


FIGURE 5.7 – Représentation sur la sphère gaussienne de deux zones segmentées (objet 1). a) Représentation 3D de deux zones segmentées. b) Représentation gaussienne de la zone verte. c) Représentation gaussienne de la zone orange.

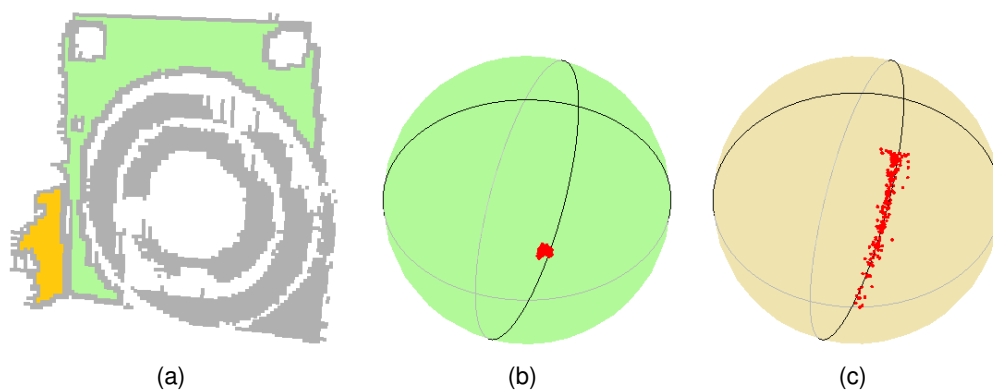


FIGURE 5.8 – Représentation sur la sphère gaussienne de deux zones segmentées (objet 2). a) Représentation 3D de deux zones segmentées. b) Représentation gaussienne de la zone verte. c) Représentation gaussienne de la zone orange.

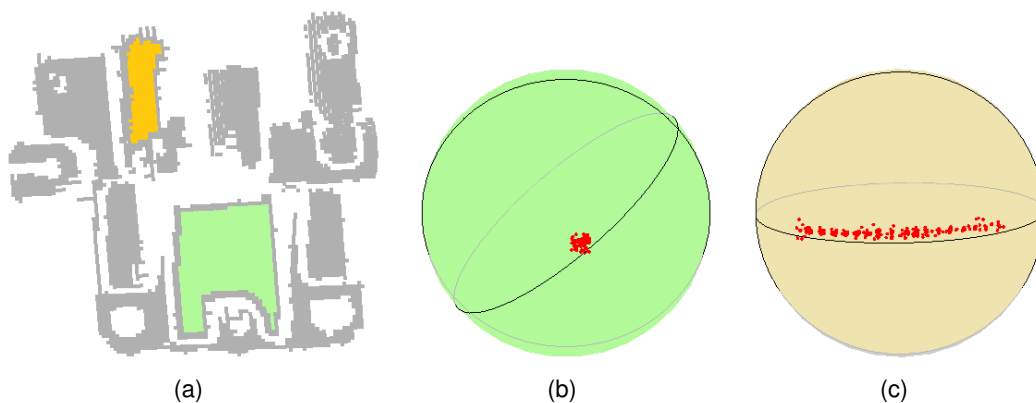


FIGURE 5.9 – Représentation sur la sphère gaussienne de deux zones segmentées (objet 3). a) Représentation 3D de deux zones segmentées. b) Représentation gaussienne de la zone verte. c) Représentation gaussienne de la zone orange.

En nous appuyant sur les valeurs propres de la matrice de covariance, nous sommes alors en mesure d'identifier si une région peut être assimilée à une primitive. Les zones dont les valeurs propres ne correspondent pas aux critères d'appartenance à une primitive seront traitées dans un second temps en approximant une surface de Bézier sur les données. La figure 5.10 illustre les résultats de cette étape sur les trois objets testés. Les zones vertes représentent les zones identifiées comme appartenant potentiellement à une primitive et les zones bleues, représentent les zones sur lesquelles des surfaces de Bézier seront approximées.

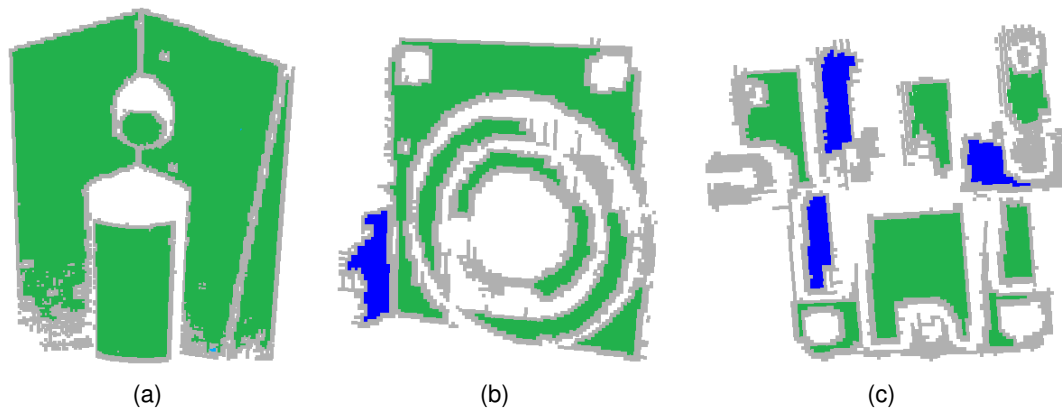


FIGURE 5.10 – *Test d'appartenance à une primitive géométrique simple : en vert les zones identifiées comme des primitives et en bleu les zones qui ne le sont pas. a) Résultats du test pour l'objet 1 : toutes les zones sont identifiées comme des primitives. b) et c) Respectivement, les résultats des objets 2 et 3 : certaines zones n'étant pas identifiées comme une primitive, une surface de Bézier sera approximée sur celles-ci.*

#### 5.2.2.1/ CAS DES PRIMITIVES

Lorsque qu'une zone correspond aux critères d'appartenance à une primitive, une première primitive est alors estimée à partir des valeurs et vecteurs propres de la matrice de covariance puis est ensuite ajustée sur les points 3D par la méthode de Gauss-Newton [100]. Si le ratio  $r$  (nombre de points corrects/nombre de points total de la zone) est supérieur au seuil défini préalablement (90% dans notre exemple), la primitive est considérée comme correcte et la zone est alors marquée. Les points *incorrects* (dont la distance à la primitive est supérieure à 0,1mm dans notre exemple) sont, quant à eux, rejetés de la zone.

Dans le cas de l'objet 1 (figure 5.7), les deux zones sélectionnées sont identifiées comme des primitives, respectivement comme une sphère et un cylindre. La figure 5.11 représente la distance entre la primitive ajustée et les points 3D. On constate, dans le cas de la sphère, que l'ensemble des points est à l'intérieur de la tolérance fixée. De ce

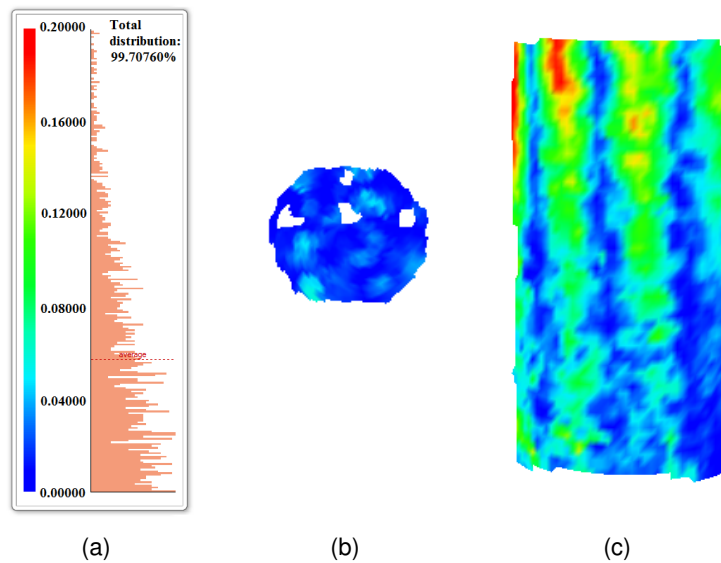


FIGURE 5.11 – *Distance entre la primitive ajustée et les points 3D pour les deux zones sélectionnées sur l'objet 1. a) Échelle des distances. b) Distance points/primitives pour la première zone (figure 5.7-b). c) Distance pour la deuxième zone (figure 5.7-c).*

fait, la zone est entièrement marquée et ne sera plus raffinée. Dans le cas de la seconde primitive, la distance entre les points 3D et le cylindre est parfois supérieure à la tolérance acceptée, mais le ratio  $r$  est de 94,412% et reste donc supérieur au seuil fixé. Dans ce cas, la primitive est conservée et seules les régions respectant la tolérance fixée sont marquées, les autres régions ne seront pas marquées et seront de nouveau traitées lors de l'itération suivante.

#### 5.2.2.2/ CAS DES SURFACES PARAMÉTRIQUES

Comme expliqué en section 4.4.1, une surface de Bézier est estimée sur toutes les zones segmentées ne remplissant pas les critères d'appartenance à une primitive géométrique simple. Les surfaces ainsi obtenues sont alors comparées aux surfaces obtenues lors de l'itération précédente afin de décider si la zone peut être marquée ou non. Dans notre cas, s'agissant de la première itération, il n'existe pas encore de surfaces pour effectuer une comparaison. De ce fait, aucune de ces zones n'est marquée lors de la première itération du système. Dans le cas de l'objet 2 (figure 5.8), la première des deux zones sélectionnées est identifiée comme un plan (figure 5.8-b), tandis que la seconde (figure 5.8-c) ne remplit aucun des critères d'appartenance à une primitive et une surface de Bézier est alors estimée dans cette zone. La figure 5.12 représente les zones marquées (en vert) et les zones non marquées (en rouge) lors de la première itération du système pour les différents objets.

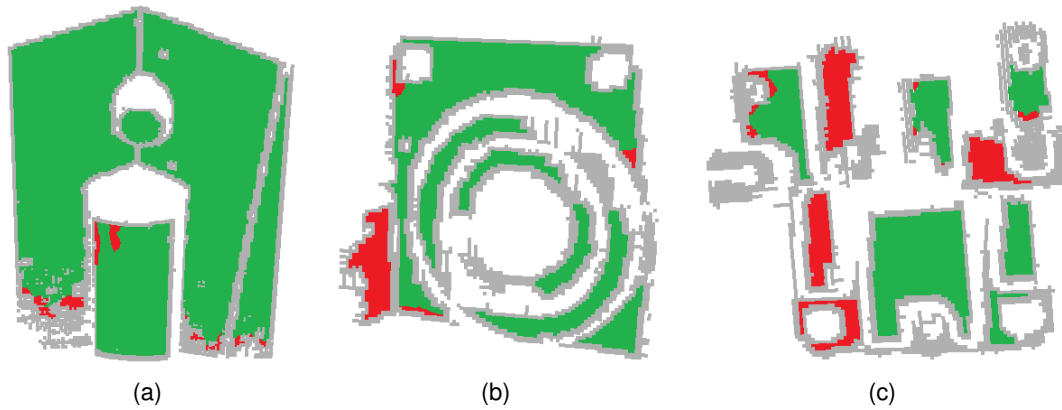


FIGURE 5.12 – Représentation sur la sphère gaussienne de deux zones segmentées (objet 3). a) Représentation 3D de deux zones segmentées. b) Représentation gaussienne de la zone verte. c) Représentation gaussienne de la zone orange.

### 5.2.3/ INCRÉMENTATION DU SYSTÈME

L'étape suivante consiste à incrémenter ou arrêter le système. Le système de numérisation ne s'arrête que dans deux situations : lorsque toutes les zones sont marquées ou bien lorsque la densité  $d$  est égale à  $d_f$ . Dans notre cas, aucune de ces conditions d'arrêt n'est respectée. Le système est alors incrémenté avec une nouvelle densité  $d = 1/4$ . Les points 3D des zones marquées sont, quant à eux, stockés.

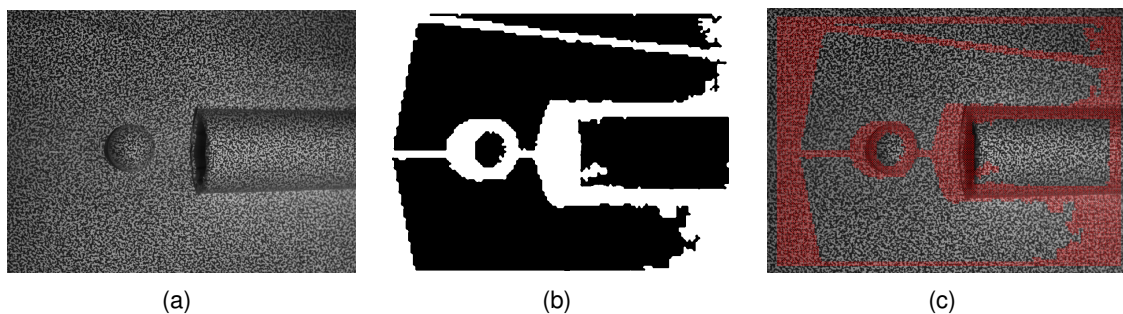


FIGURE 5.13 – Positionnement des graines servant à la mise en correspondance pour l'objet 1. a) Image capturée par le scanner. b) Masque des zones marquées à la première itération. c) Grille de graine (en rouge) utilisée pour la mise en correspondance lors de l'itération suivante.

Lors de la nouvelle itération du système, seules les zones non marquées et non traitées lors de l'itération précédente sont raffinées. Pour cela, on applique le principe présenté en section 4.5 en appliquant un masque sur la grille de graines servant à appairer les points entre les deux images du scanner (figure 5.13).



### 5.2.4/ RÉSULTATS

Le système reprend le même processus que celui que l'on vient de décrire mais en ne traitant que les zones non marquées. Voici les résultats obtenus pour chaque objet à l'itération suivante pour une densité  $d = 1/4$  (figure 5.14).

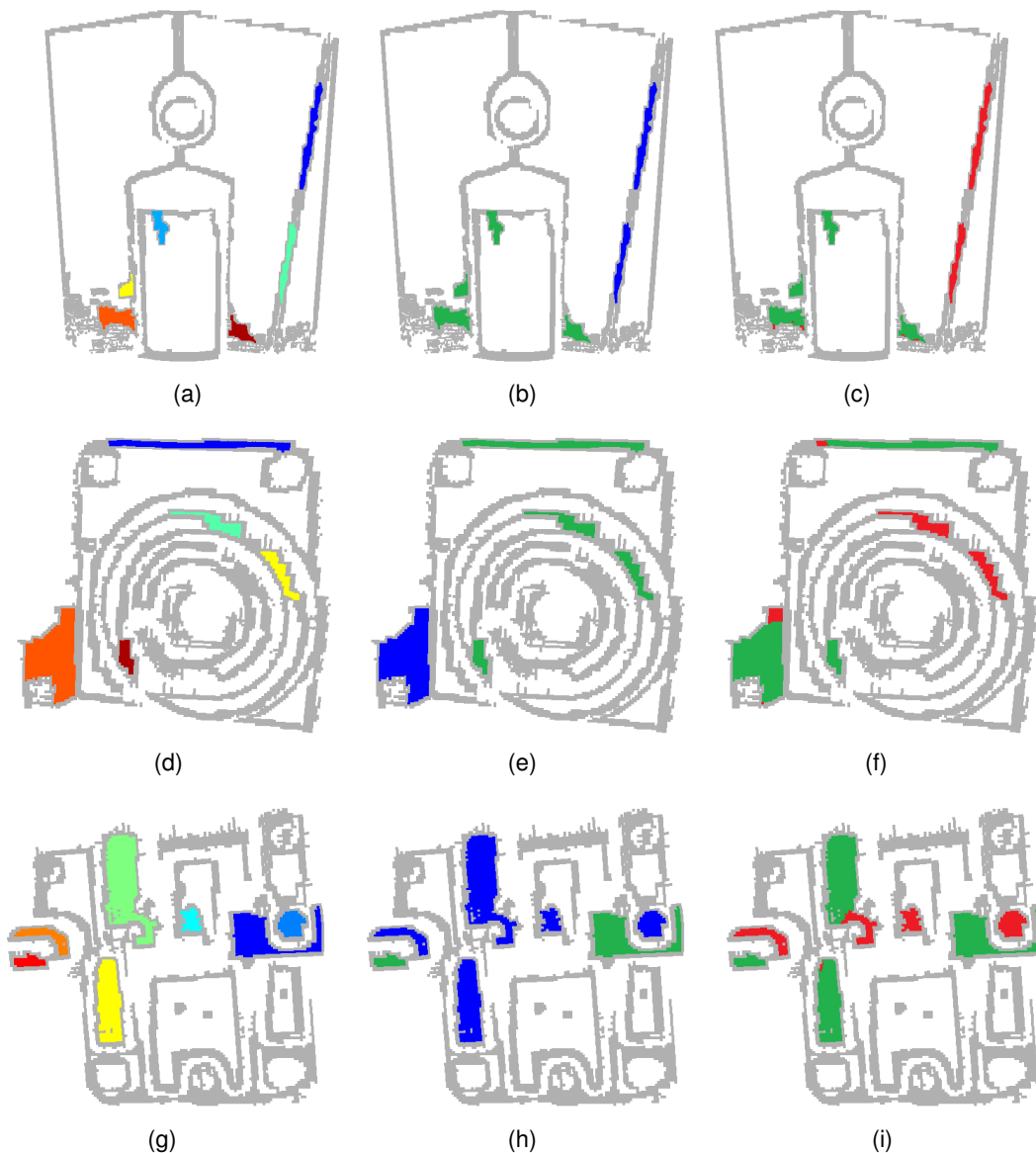


FIGURE 5.14 – Résultats pour les trois objets avec une densité  $d = 1/4$ . a), d) et g) Zones segmentées. b), e) et h) Identification des primitives (zones vertes) et des surfaces (zones bleues). c), f) et i) Zones marquées (vertes) et non marquées (rouges).

On peut remarquer que pour l'objet 1, qui est principalement composé de primitives et de zones à fortes courbures, (figure 5.14) seules les zones identifiées comme des primitives sont marquées. Dans les cas des deux autres objets, qui présentent des formes plus complexes (qui ne peuvent être représentées par des primitives), il est possible de re-

marquer que plusieurs zones marquées le sont par comparaison de la surface de Bézier obtenue à cette itération avec la surface paramétrique obtenue à l'itération précédente. En procédant de la même façon jusqu'à la dernière itération du système ( $d = 1/2$  dans ce cas), ou jusqu'à ce que tous les objets segmentés soient marqués, nous sommes en mesure d'obtenir des données dont la densité de points sera représentative de la complexité de la forme (figure 5.15) sans *a priori* sur la forme de l'objet. Afin d'obtenir un meilleur rendu visuel, les données ont été maillées, cependant le modèle obtenu en sortie du système est un nuage de points.

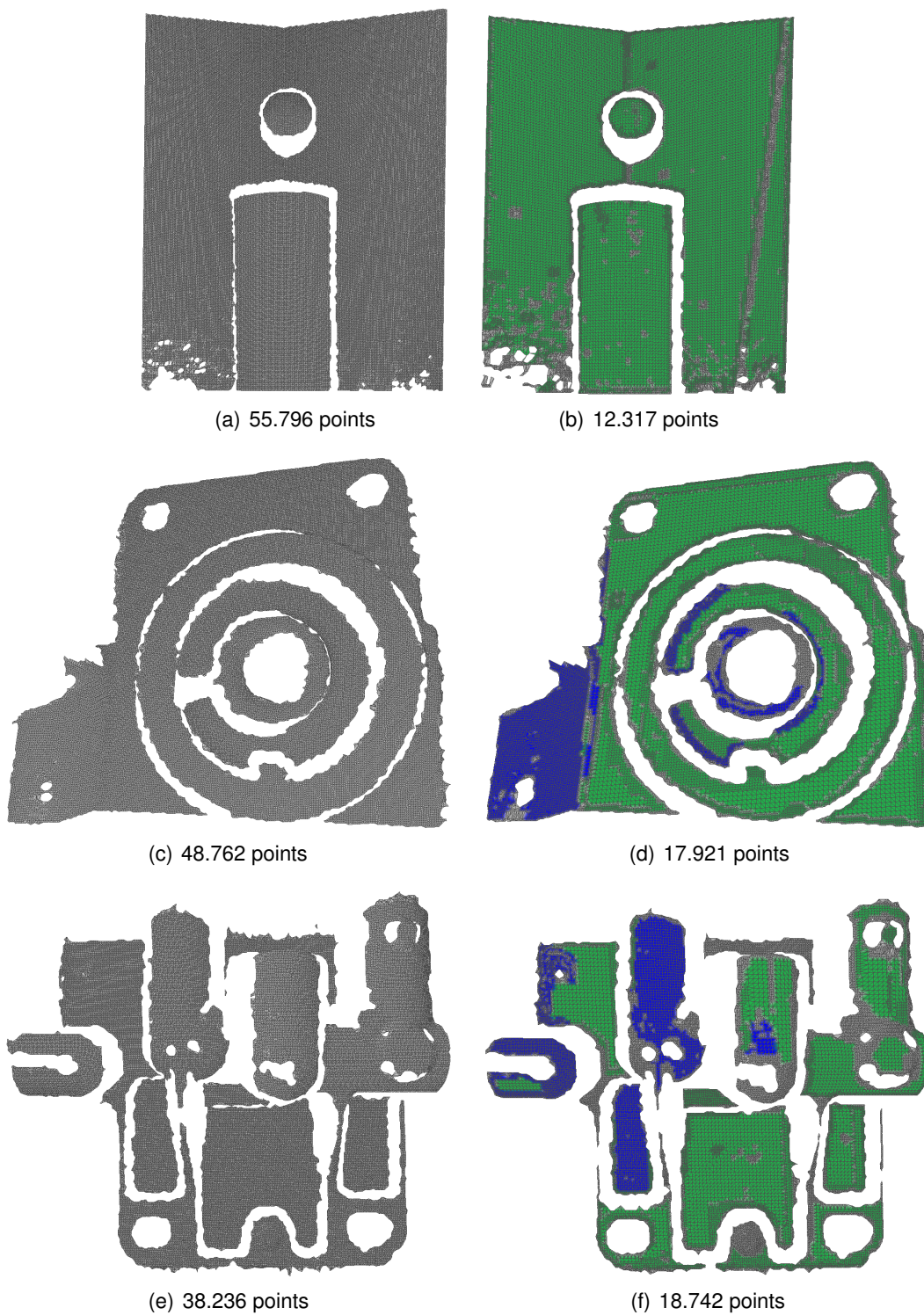


FIGURE 5.15 – Résultats finaux obtenus pour chaque objet. a), c) et e) Nuage de points maillé obtenu à la densité maximale  $d = 1/2$ . b), d) et f) Nuage de points maillé obtenu avec notre système. En vert on trouve les zones identifiées comme des primitives et en bleu les zones identifiées comme des surfaces paramétriques. Les zones grises correspondent aux points calculés pour  $d = d_f$  qui n'ont pas été identifiés par une primitive.

### 5.3/ IMPACT DES DIFFÉRENTS PARAMÈTRES

Comme cela a été mentionné dans la partie précédente, pour chaque étape (acquisition, marquage des régions) l'utilisateur a la possibilité d'agir sur les différents paramètres intervenant dans chaque étape. Le réglage de ces paramètres peut amener à des résultats très différents. En choisissant un ratio faible pour les primitives, il est possible de re-segmenter les objets plus précisément au risque de commettre des erreurs sur les primitives identifiées. A contrario, un ratio important permet de ne de conserver que les zones qui correspondent vraiment à une primitive, et le moindre défaut peut alors entraîner un rejet de la primitive. Les figures 5.16 et 5.17 présentent les résultats que l'on obtient sur les différents objets en nous plaçant dans deux cas distincts. Dans le premier cas on cherche à obtenir les meilleures primitives possibles (ratio  $r = 95\%$ ), au détriment de la simplification et dans le second cas, nous avons choisi de rechercher le maximum de primitives géométriques (ratio  $r = 75\%$ ) au risque de commettre une erreur. Les autres paramètres (distances tolérées, ratio sur les surfaces) restent inchangés par rapport à ceux utilisés précédemment.

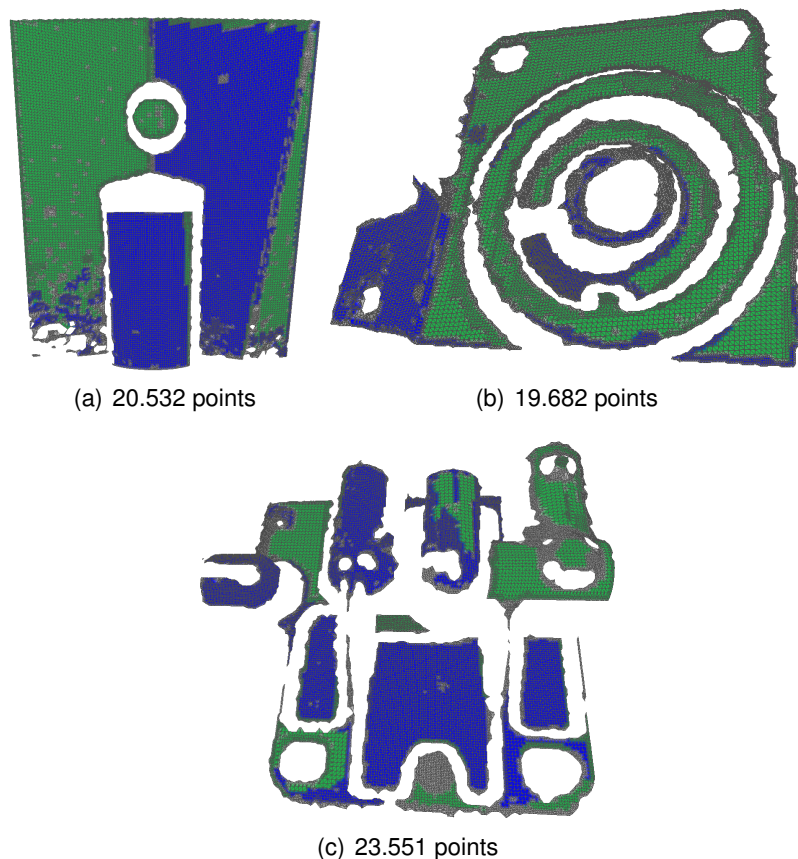


FIGURE 5.16 – Résultats obtenus sur chaque objet avec les paramètres du premier cas. Distance moyenne  $\epsilon$  entre les données obtenues et les données denses. a)  $\epsilon = 0,01832$  mm. b)  $\epsilon = 0,02109$  mm. c)  $\epsilon = 0,01908$  mm.

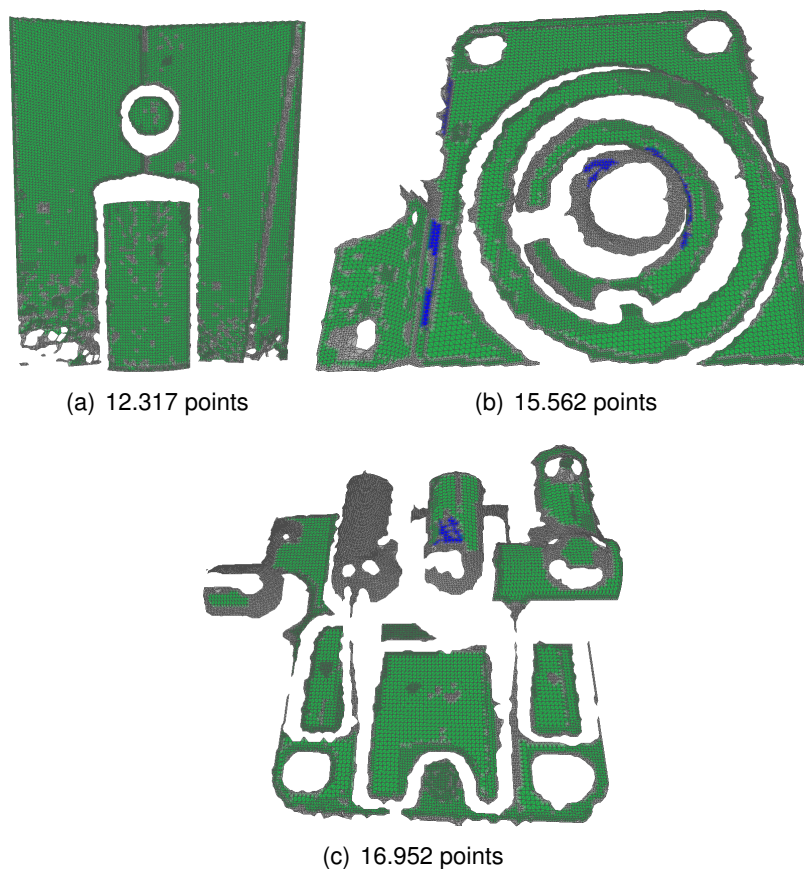


FIGURE 5.17 – Résultats obtenus sur chaque objet avec les paramètres du second cas. Distance moyenne  $\epsilon$  entre les données obtenues et les données denses. a)  $\epsilon = 0,01844$  mm. b)  $\epsilon = 0,02195$  mm. c)  $\epsilon = 0,02506$  mm.

Dans le cas des deux premiers objets, la figure 5.17 montre que l'utilisation d'un ratio faible (75%) permet de réduire le nombre de points final sans pour autant affecter la précision des données. Ce résultat correspond à ce qu'on pouvait attendre dans la mesure où les objets en question sont composés de nombreuses primitives. En revanche, concernant le dernier objet, l'utilisation d'un faible ratio entraîne des erreurs bien plus importantes que dans le cas d'un ratio de 95%. L'objet étant composé de formes plus complexes, le fait de les identifier par des primitives entraîne des erreurs mais permet malgré tout de réduire le nombre de points. La modification du ratio sur les surfaces fonctionne de la même façon que le ratio sur les primitives. Plus celui-ci est grand et plus les données finales seront lourdes et plus le ratio sera faible plus les erreurs engendrées seront grandes.

La distance tolérée entre la primitive/surface et les points 3D affecte indirectement le ratio de points *corrects*. Augmenter la distance tolérée revient finalement à utiliser un ratio plus important, à une différence près, l'erreur en sortie est contrôlée, elle ne dépassera



jamais la distance fixée. Comme nous venons de le voir, les résultats obtenus avec notre système sont complètement dépendants de ces paramètres. Ainsi, l'utilisateur devra ajuster les différents paramètres en fonction de l'utilisation finale des données. Comme illustré par les figures 5.18 et 5.19, si l'on cherche à simplifier au maximum les données, il faudra tolérer une erreur assez importante entre les primitives/surfaces et les points 3D. À l'inverse, si les données finales doivent être précises, il faudra tolérer une erreur plus faible sur la distance tolérée.

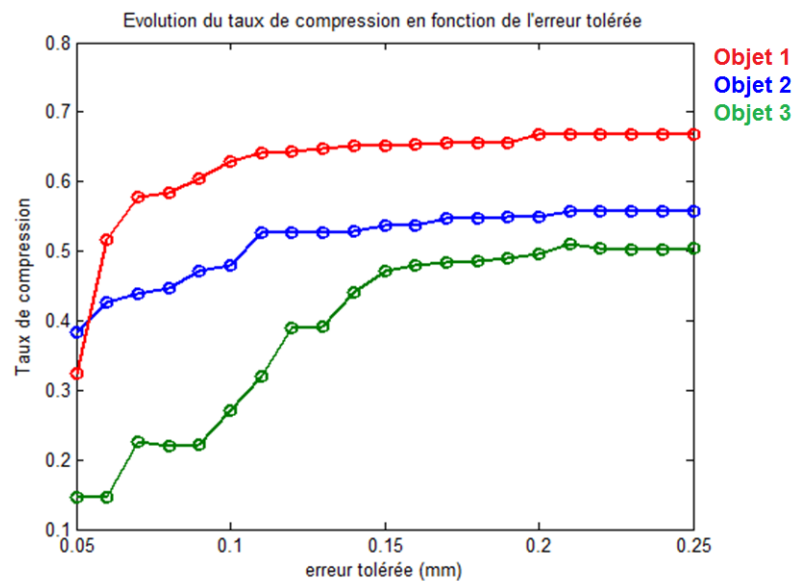


FIGURE 5.18 – Evolution du taux de compression en fonction de l'erreur tolérée.

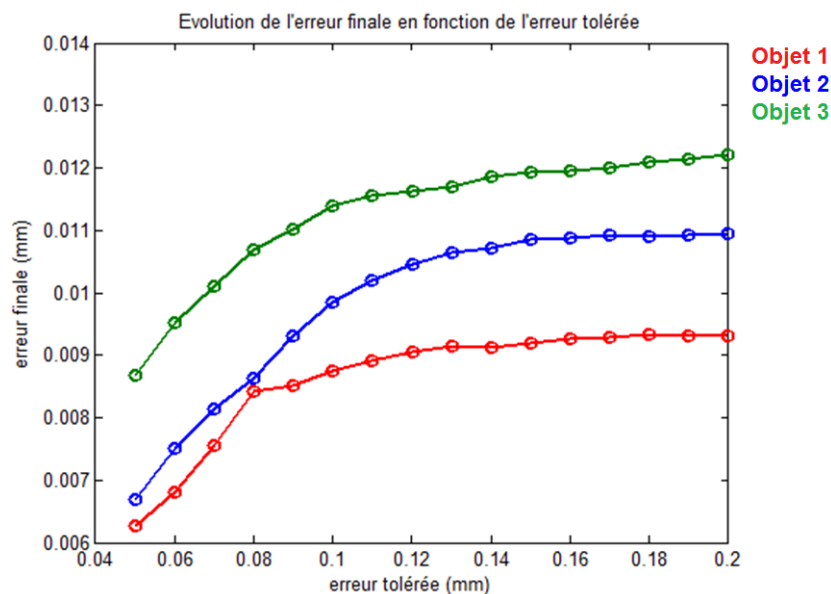


FIGURE 5.19 – Evolution de l'erreur finale en fonction de l'erreur tolérée.

## 5.4/ ÉVALUATION DE NOTRE MÉTHODE

Afin de vérifier le bon fonctionnement, et la précision de notre système, nous avons cherché à comparer notre méthode avec celles déjà existantes. Cependant, à notre connaissance il n'existe pas d'approche permettant d'adapter l'acquisition en vue de simplifier les données dès l'acquisition. De ce fait, nous comparons les résultats que nous obtenons avec des méthodes de simplification basées sur le maillage à partir d'un scan dense, ce qui correspond finalement au cheminement classique de la chaîne 3D. Nous avons comparé les résultats de notre système avec les résultats de deux autres méthodes de simplification de maillage avec préservation d'arêtes : QSlim [75] et ACVD [107] pour un nombre de points final d'environ 15.000 points avec chaque méthode.

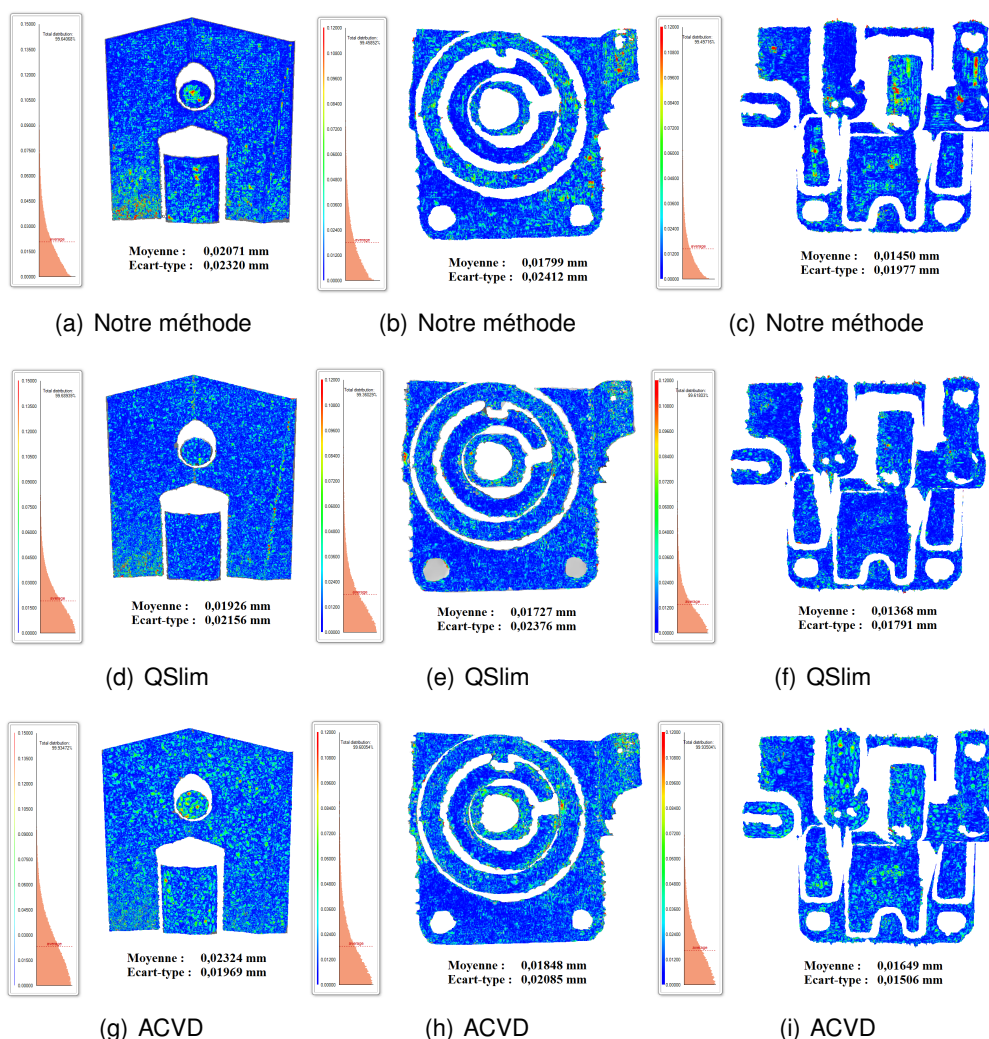


FIGURE 5.20 – Résultats obtenus par différentes méthodes de simplification.

La figure 5.20 représente la distance quadratique entre le nuage de points dense et le

nuage de points simplifié obtenue avec les trois méthodes sur chaque objet. Globalement, les erreurs mesurées sont similaires entre les trois méthodes, et ce, quel que soit l'objet étudié. Les figures 5.21, 5.22 et 5.23 montrent l'évolution de l'erreur finale en fonction du taux de compression pour les trois méthodes (notre méthode, ACVD et Qslim) sur les trois objets présentés dans ce manuscrit.

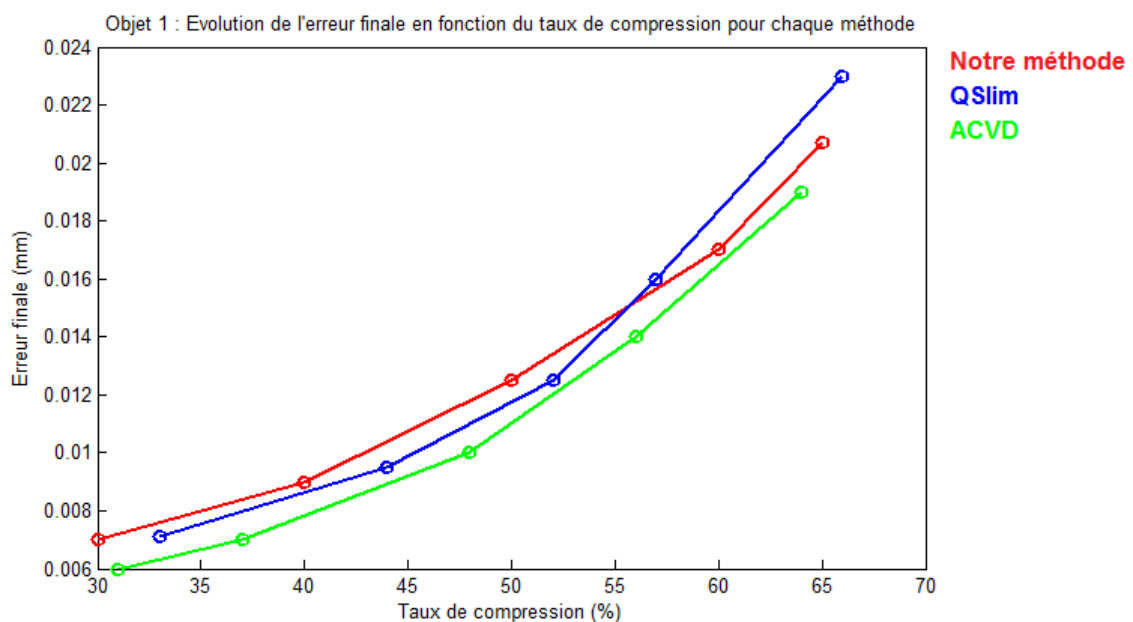


FIGURE 5.21 – *Objet 1 : Evolution de l'erreur finale en fonction du taux de compression.*

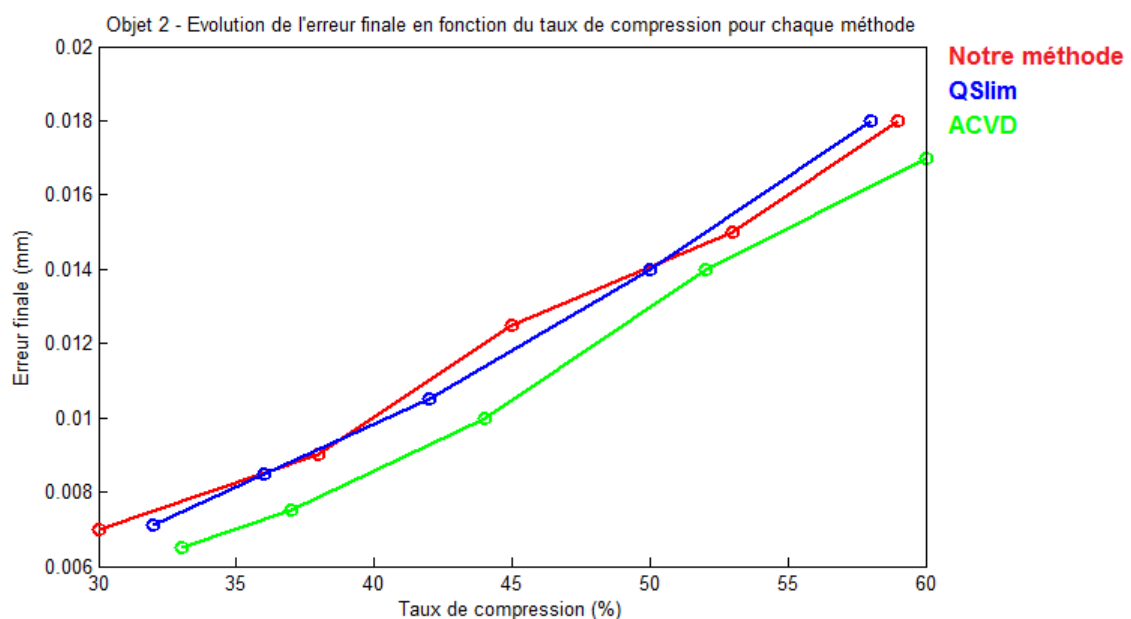


FIGURE 5.22 – *Objet 2 : Evolution de l'erreur finale en fonction du taux de compression.*

Le système de numérisation dynamique présenté dans ce mémoire permet donc



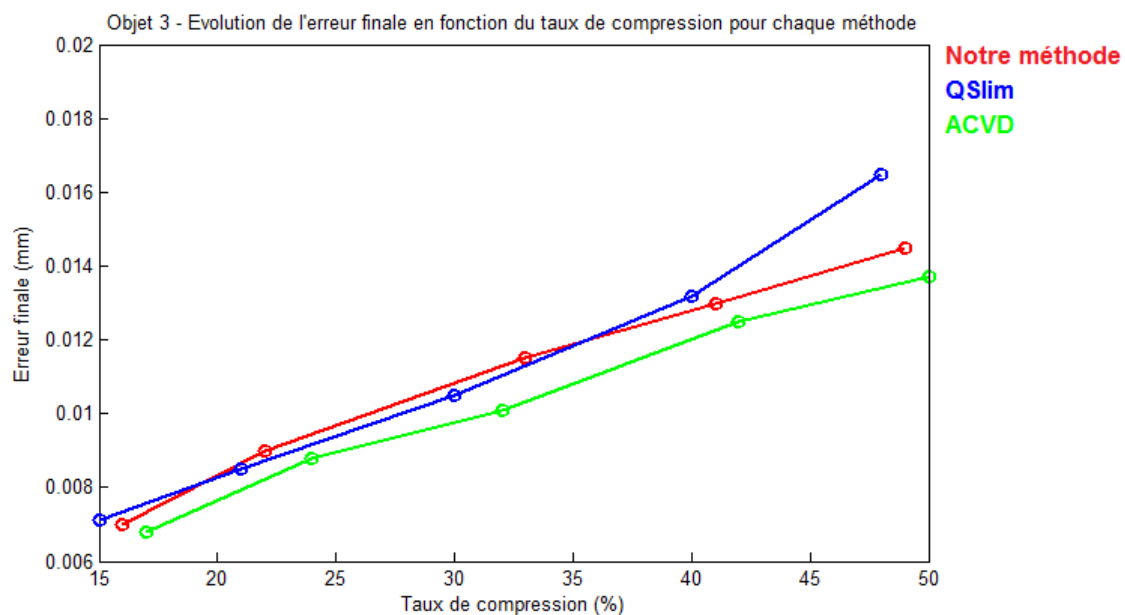


FIGURE 5.23 – *Objet 3 : Evolution de l'erreur finale en fonction du taux de compression.*

d'obtenir un modèle simplifié dès la sortie de l'acquisition sans avoir à passer par les différentes étapes de la chaîne 3D, tout en assurant un résultat similaire à ce que l'on obtiendrait en passant par le cheminement classique. D'autres résultats obtenus sur d'autres objets manufacturés sont présentés en Annexe B.

## 5.5/ AMÉLIORATION DU SYSTÈME

Le système présenté dans ce mémoire permet d'adapter la densité de points obtenus en fonction de la forme de l'objet en recherchant des primitives 3D ou des surfaces de Bézier. Pourtant, il existe encore d'autres éléments simples souvent présents sur des pièces manufacturées, comme des trous de perçage par exemple, qui ne sont pas exploités avec notre système actuel. Malheureusement, le problème avec ce genre de forme se situe au niveau de l'appariement qui ne peut se faire en utilisant le contour étant donné que le motif projeté n'est pas visible au niveau du trou. Pourtant, les contours du trou sont visibles dans les images du scanner et peuvent être utilisés en supposant que la forme 3D correspond bien à une forme circulaire. Ainsi, afin d'ajouter encore plus d'informations, nous avons développé une approche sur l'objet présenté en figure 5.24 obtenu à partir de données de l'OptiNum.



FIGURE 5.24 – Image fournie par l'Optinum sur un objet test.

#### 5.5.1/ PRINCIPE DE LA MÉTHODE

Ici, l'objectif est de calculer de nouveaux points 3D au niveau des contours du trou circulaire. Pour cela, nous supposons que la forme 3D du trou peut être modélisée par une forme circulaire, ou elliptique. La projection de cette forme dans les images formera alors des ellipses. L'ellipse est une conique, définie par l'équation suivante :

$$Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0 \quad (5.1)$$

Le principe de notre approche peut être résumé par le schéma présenté en figure 5.25. La projection de la forme 3D elliptique forme deux ellipses visibles dans les images. L'idée consiste à segmenter la projection du trou dans les deux images afin d'en obtenir le contour. À partir des points du contour, il est possible d'estimer la meilleure conique passant par ces derniers avec la méthode des moindres carrés et ainsi modéliser les deux ellipses dans l'image (en bleu sur la figure 5.25). Les images ayant été rectifiées lors de l'étape de calibration, en calculant l'intersection d'une ligne horizontale (en rouge) avec les deux ellipses, il est possible de déterminer des correspondants et ainsi de remonter à la forme 3D du trou.

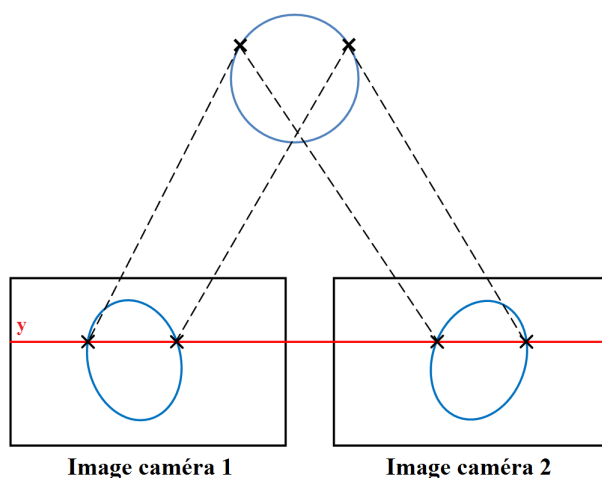


FIGURE 5.25 – Principe utilisé pour déterminer les contours d'un trou de perçage.

### 5.5.2/ EXTRACTION DES PARAMÈTRES DES ELLIPSES

Les paramètres des deux ellipses peuvent facilement être extraits depuis la conique ajustée sur le contour de la forme. Le point central de l'ellipse  $(X_0, Y_0)$  ainsi que les rayons des axes principaux  $r_1$  et  $r_2$  sont obtenus à partir des paramètres de la conique de la façon suivante :

$$X_0 = -\frac{D}{2A} \quad , \quad Y_0 = -\frac{E}{2C} \quad , \quad (5.2)$$

$$r_1 = \sqrt{\frac{F}{A}} \quad , \quad r_2 = \sqrt{\frac{F}{C}} \quad . \quad (5.3)$$

L'équation paramétrique s'écrit :

$$\begin{cases} x = X_0 + r_1 \cdot \cos(\theta) \\ y = Y_0 + r_2 \cdot \sin(\theta) \end{cases} \quad , \quad (5.4)$$

avec  $x$  et  $y$  la position des points appartenant à l'ellipse dans son propre repère. On peut également exprimer ces coordonnées dans le repère image en effectuant une rotation de notre repère, ce qui s'écrit :

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = R \begin{bmatrix} x \\ y \end{bmatrix} \quad , \quad (5.5)$$

avec  $x'$  et  $y'$  les points appartenant à l'ellipse dans le repère image et  $R$  la matrice de

rotation entre les deux repères. Cette matrice de rotation s'écrit :

$$R = \begin{bmatrix} \cos(\phi) & -\sin(\phi) \\ \sin(\phi) & \cos(\phi) \end{bmatrix}, \quad (5.6)$$

avec  $\phi$  l'angle entre les deux repères. Cet angle peut également être obtenu à partir de la conique. Ainsi, simplement en segmentant le contour du trou dans chaque image, puis en ajustant une conique sur ce contour, il est possible de modéliser une ellipse pour chaque image.

### 5.5.3/ RECONSTRUCTION DE LA FORME 3D

En développant l'équation 5.6 à partir de 5.5, pour une image il est alors possible d'écrire :

$$x' = X_0.\cos(\phi) + r_1.\cos(\phi).\cos(\theta) - Y_0.\sin(\phi) - r_2.\sin(\phi).\sin(\theta), \quad (5.7)$$

$$y' = X_0.\sin(\phi) + r_1.\sin(\phi).\cos(\theta) + Y_0.\cos(\phi) + r_2.\cos(\phi).\sin(\theta). \quad (5.8)$$

En calculant  $\theta$  à partir de 5.8 et en le ré-injectant dans 5.7, il est possible de trouver l'intersection entre une ligne de l'image et l'ellipse estimée à partir des contours du trou. En appliquant la même méthode dans l'autre image (pour la même ligne), on trouve ainsi les points correspondants. Les formes des ellipses étant modélisées mathématiquement, il est possible de calculer autant de points que l'on désire au niveau du contour, sans être limité par la résolution du scanner et ainsi obtenir un contour beaucoup plus dense que ce que l'on pourrait obtenir avec un scan classique. La figure 5.26 illustre les résultats obtenus sur l'objet test.

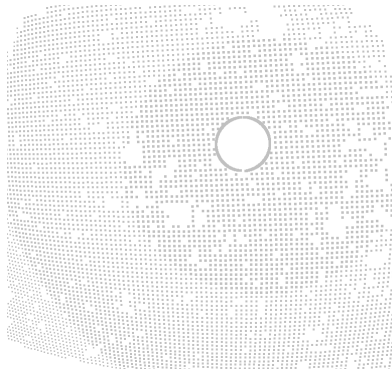


FIGURE 5.26 – Principe utilisé pour déterminer les contours d'un trou de perçage.

Ces résultats montrent qu'en parallèle du système de numérisation dynamique, il est possible d'utiliser cette méthode pour obtenir encore d'avantage d'informations à partir des images, notamment au niveau des contours. En utilisant le même procédé, il est également possible d'identifier d'autres formes comme les lignes droites (bordures de pièces mécaniques) par exemple.

Cependant, cette approche présente une contrainte importante, qui se situe au niveau de la segmentation du trou. Dans le cas où le trou est bien distinct comme dans l'exemple présenté ci-dessus, il est assez simple à segmenter. Malheureusement, dans la plupart des cas traités, les trous sont beaucoup plus difficiles à identifier à cause de la projection du motif pseudo-aléatoire et des conditions d'éclairage changeantes. Bien que le principe présenté dans cette partie soit fonctionnel sur cet objet test, nous ne l'avons pas implémenté directement dans le système de numérisation dynamique à cause de la difficulté à segmenter le trou automatiquement.

## CONCLUSION ET PERSPECTIVES

Actuellement, la chaîne 3D classique menant à un modèle compressé, telle que présentée dans ce document, est séquentielle. Le principe de cette chaîne 3D consiste à acquérir un maximum de données, d'en extraire diverses informations (courbure, normales), puis de simplifier ces données. Dans ce manuscrit, nous avons proposé et développé un système de numérisation dynamique, permettant de réduire les données dès l'acquisition et ainsi d'éviter ce cheminement classique. Le système développé permet d'adapter le nombre de points directement lors de l'acquisition en fonction de la complexité de l'objet à numériser. Dans ce chapitre, nous présentons un bilan de nos travaux de recherche réalisés dans le cadre du projet CreActive 3D ainsi que les perspectives des travaux restants à réaliser.

### 6.1/ RÉSUMÉ DE NOS CONTRIBUTIONS

Dans ce manuscrit, nous avons présenté un système de numérisation dynamique permettant d'adapter la densité de point lors de l'acquisition pour obtenir des données simplifiées, et ce sans avoir à passer par les différentes étapes de la chaîne 3D classique. Pour cela, nous avons dans un premier temps cherché à extraire un maximum d'informations complémentaires aux points 3D en exploitant les images fournies par le scanner. Ces informations sont les suivantes :

- **Les normales à la surface :**

Les normales 3D sont l'information complémentaire la plus fréquemment utilisée dans l'état de l'art. Afin d'estimer les normales depuis les images du scanner, nous nous sommes inspiré d'une méthode existante [56] et l'avons adapté au motif du scanner à notre disposition.

- **Les discontinuités :**

Dans la chaîne 3D classique, le second attribut le plus fréquemment utilisé est la courbure 3D. Malheureusement, nous ne sommes pas parvenu à extraire une courbure 3D à partir des images. Généralement, la courbure 3D est principalement utilisée pour segmenter les zones homogènes d'un objet. C'est pourquoi, nous avons cherché à extraire un autre attribut depuis les images du scanner permettant une utilisation similaire à la courbure 3D. Dans ce manuscrit, nous proposons une approche pour estimer les discontinuités d'un objet.

- **Les primitives :**

Bien que les primitives 3D ne soit pas un attribut complémentaire à proprement parlé, il s'agit néanmoins d'une information importante. En nous inspirant de travaux existants sur des données 3D, nous avons mis en place une méthode permettant d'identifier et de modéliser des primitives à partir d'une analyse en composantes principales depuis la représentation gaussienne des données.

**Système de numérisation dynamique** Dans un second chapitre, nous présentons le fonctionnement du système en lui même. Le schéma présenté en figure 6.1 illustre le principe général du système de numérisation dynamique. Ce système peut se décomposer en 4 parties distinctes :

- **Acquisition :**

Cette étape consiste à acquérir des données images à partir d'un scanner à stéréovision. En appariant des points entre les deux images ainsi obtenues à partir d'une grille de germe, il est possible d'obtenir un nuage de points d'une certaine densité.

- **Extraction d'informations :**

La seconde partie du système consiste à extraire différentes informations complémentaires aux points 3D à partir des images du scanner : normales à la surface et coefficients de discontinuité. L'exploitation de ces attributs permet de segmenter la scène en plusieurs régions homogènes.

- **Marquage des régions :**

Cette partie peut se décomposer en deux sous-parties : l'extraction de primitives et l'approximation de surfaces paramétriques. Les différentes régions préalablement segmentées peuvent être identifiées comme des primitives géométriques ou des surfaces de Bézier, à partir de la l'image gaussienne de chaque région. Lorsqu'une région est identifiée comme appartenant à une primitive, elle est alors marquée et ne sera plus traitée par la suite. Les autres régions, n'ayant pas été identifiées par une primitive, sont approximées par une surface de Bézier. Ces surfaces seront alors comparées avec d'autres surfaces, approximées lors des itérations suivantes. Si les surfaces évoluent entre deux itérations successives, la région sera encore raffinée. A contrario, si les surfaces sont similaires la zone est alors marquée et ne sera plus traitée par la suite.

- **Incrémentation et arrêt du système :**

Lorsque toutes les zones ont été traitées, le système est alors incrémenté et recommence depuis la première étape en utilisant une grille plus fine pour appairer les points entre les deux images. Toutes les zones non marquées seront alors raffinées (calcul de nouveaux points). Le système procède ainsi jusqu'à ce qu'une des conditions d'arrêt soit atteinte.

Ce type de système permet ainsi d'adapter l'acquisition à la forme de l'objet directement à partir de l'acquisition, et fournit ainsi un nuage de points simplifié directement à la sortie du scanner de manière complètement automatique (à l'exception des différents paramètres à régler avant le début de l'acquisition), permettant ainsi d'éviter de passer par les différentes étapes de la chaîne 3D classique.

## 6.2/ PERSPECTIVES

L'efficacité de ce système de numérisation dynamique repose en grande partie sur les primitives présentes dans la scène. En effet, lorsqu'une primitive est identifiée la zone est alors marquée et n'est plus raffinée. De ce fait, plus il y aura de primitives identifiables et plus le système sera performant. Actuellement, le système ne recherche que quatre types de primitives dans la scène à numériser. Une amélioration de ce système consisterait à élargir la gamme de primitives en recherchant d'autres formes plus complexes. Dans la littérature, il existe déjà des travaux permettant d'extraire d'autres primitives géométriques à partir d'un nuage de points, comme des tores [108].



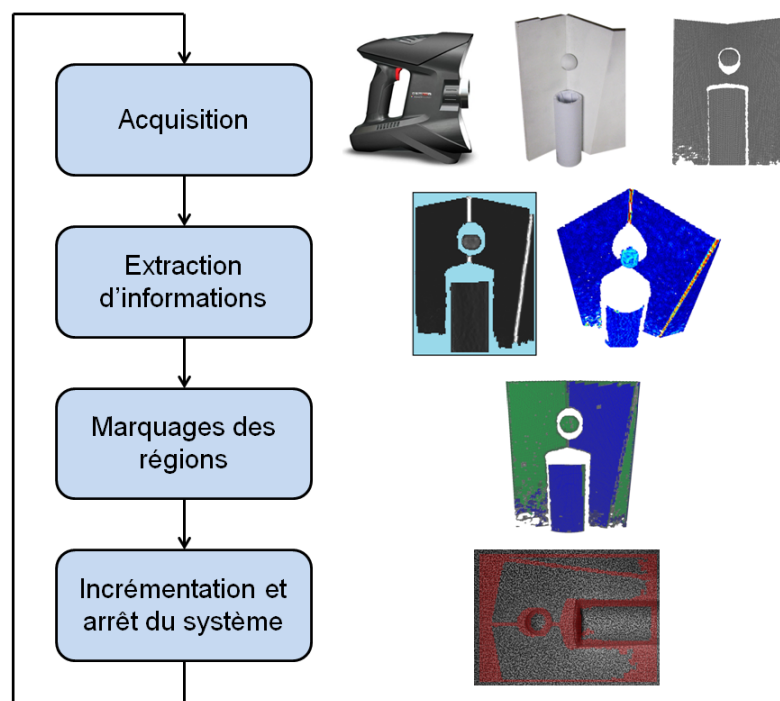


FIGURE 6.1 – schéma du système de numérisation dynamique.

Une autre amélioration du système consisterait à étendre le principe de détection des trous à d'autres formes comme les lignes et les contours. En appliquant le même principe à ces nouvelles formes, il serait possible d'obtenir plus d'informations sur des zones où les scanners classiques n'y parviennent pas. Cependant, le problème reste le même que dans le cas des trous, la segmentation des lignes et des contours s'avère très compliquée à mettre en œuvre à cause du motif pseudo-aléatoire projeté. Une solution envisageable consisterait à acquérir deux images sans le motif projeté en plus des deux images avec le motif, afin d'effectuer la segmentation des trous, lignes et contours directement depuis cette image. Cependant, le scanner et l'objet à numériser ne doivent absolument pas bouger entre l'acquisition des images avec et sans le motif. C'est en partie pour ces raisons que nous n'avons pas pu implémenter cette méthode dans notre système de numérisation dynamique.

## 6.3/ PUBLICATIONS

Les travaux présentés dans ce manuscrit ont fait l'objet de plusieurs publications, listées ci-dessous.

### **Revue :**

- "Interactive 3D Acquisition System", Machine Vision and Application, 2015 (soumis).

### **Conférences nationales et internationales :**

- "Primitives based on structured-light images", V. Daval, F. Truchetet, O. Aubreton, QCAV'13, Fukuoka, 2013.
- "Extraction de primitives à partir d'images de lumière structurée", V. Daval, F. Truchetet, O. Aubreton, Orasis, Congrès des jeunes chercheurs en vision par ordinateur, Cluny, 2013.
- "Estimation de surface de Bézier à partir d'images de lumière structurée dans une optique de compression", V. Daval, A. Bajard, F. Truchetet, O. Aubreton, CORESA 2013, Le Creusot, 2013.
- "Coarse to Fine : Toward a Intelligent 3D Acquisition System", V. Daval, F. Truchetet, O. Aubreton, SPIE Electronic Imaging 2015, San Francisco, 2015.
- "A coarse to fine 3D acquisition system", V. Daval, F. Truchetet, O. Aubreton, QCAV 2015, Le Creusot, 2015 (accepté).



# BIBLIOGRAPHIE

- [1] "Laboratoire d'informatique, signaux et systèmes de sophia antipolis." <http://www.i3s.unice.fr/I3S/index.php>. [Online].
- [2] "Laboratoire electronique informatique et image." <http://le2i.cnrs.fr/>. [Online].
- [3] "Société noomeo." Labège, France.
- [4] P. J. Besl, *Advances in Machine Vision - Chapter : Active Optical Range Imaging Sensors*. New York, NY, USA : Springer-Verlag New York, Inc., 1988.
- [5] B. Curless, "Overview of Active Vision Techniques," *SIGGRAPH'99*, 1999.
- [6] P. Jacquot, "Speckle interferometry : A review of the principal methods in use for experimental mechanics applications," *Strain*, vol. 44, no. 1, pp. 57–69, 2008.
- [7] R. Jones and C. Wykes, *Holographic and Speckle Interferometry*. Cambridge University Press, 1989.
- [8] J. Schmit, K. Creath, and J. C. Wyant, *Surface Profilers, Multiple Wavelength, and White Light Intereferometry, in Optical Shop Testing*. Houghton Mifflin, 2006.
- [9] K. Creath and J. Wyant, "Moiré and fringe projection techniques," *Optical shop testing*, pp. 653–85, 1992.
- [10] L. B. Wolff and T. E. Boulton, "Constraining object features using a polarization reflectance model," *Pattern Analysis and Machine Intelligence*, vol. 13, pp. 635–657, 1991.
- [11] O. Morel, C. Stolz, F. Meriaudeau, and P. Gorria, "Active lighting applied to three-dimensional reconstruction of specular metallic surfaces by polarization imaging," *Applied Optics*, vol. 45, pp. 4062–4068, Jun 2006.
- [12] C. P. Huynh, A. Robles-Kelly, and E. Hancock, "Shape and refractive index recovery from single-view polarisation images," pp. 1229–1236, 2010.
- [13] G. Eren, O. Aubreton, F. Meriaudeau, S. Secades, D. Fofi, T. Naskali, F. Truchetet, and A. Ercil, "Scanning from heating : 3d shape estimation of transparent objects from local surface heating," *Optique Express*, vol. 17, pp. 11457–11468, Jul 2009.
- [14] R. Rantson, C. Stolz, D. Fofi, and F. Meriaudeau, "Optimization of transparent objects digitization from visible fluorescence ultraviolet induced," *Optical Engineering*, vol. 51, no. 3, pp. 033601–1–033601–14, 2012.

- [15] S. Chambon, *Mise en correspondance stéréoscopique d'images couleur en présence d'occultations*. PhD thesis, Université toulouse III-Paul Sabatier, 2005.
- [16] T. Molinier, D. Fofi, J. Salvi, and P. Gorria, "2d virtual texture on 3d real object with coded structured light," *Image Processing : Machine Vision Applications*, 2008.
- [17] J. J. Gibson, *The Perception of the Visual World*. Houghton Mifflin, 1950.
- [18] S. Christy and R. Horaud, "Euclidean shape and motion from multiple perspective views by affine iterations," *Pattern Analysis and Machine Intelligence*, vol. 18, pp. 1098–1104, 1996.
- [19] S. Nayar and Y. Nakagawa, "Shape from focus," *Pattern Analysis and Machine Intelligence*, vol. 16, pp. 824–831, Aug 1994.
- [20] J. Salvi, J. Pagès, and J. Batlle, "Pattern codification strategies in structured light systems," *Pattern Recognition*, vol. 37, pp. 827–849, 2004.
- [21] M. Minou, T. Kanade, and T. Sakai, "A method of time-coded parallel planes of light for depth measurement," *Trans. Institute of Electronics and Communication Engineers of Japan*, vol. E64, pp. 521–528, August 1981.
- [22] J. Posdamer and M. Altschuler, "Surface measurement by space-encoded projected beam systems," *Computer Graphics and Image Processing*, vol. 18, no. 1, pp. 1 – 17, 1982.
- [23] S. Inokuchi, K. Sato, and F. Matsuda, "Range imaging system for 3-d object recognition," *Pattern Recognition*, 1984.
- [24] J. Tajima and M. Iwakawa, "3-d data acquisition by rainbow range finder," *Pattern Recognition*, vol. i, pp. 309–313 vol.1, Jun 1990.
- [25] T. Sato, "Multispectral pattern projection range finder," *Three-Dimensional Image Capture and Applications*, vol. 3640, pp. 28–37, 1999.
- [26] P. Griffin, L. Narasimhan, and S. Yee, "Generation of uniquely encoded light patterns for range data acquisition," *Pattern Recognition*, vol. 25, no. 6, pp. 609 – 616, 1992.
- [27] C. Davies and M. Nixon, "A hough transform for detecting the location and orientation of three-dimensional surfaces via color encoded spots," *Systems, Man, and Cybernetics*, vol. 28, pp. 90–95, Feb 1998.
- [28] S. Kiyasu, H. Hoshino, K. Yano, and S. Fujimura, "Measurement of the 3-d shape of specular polyhedrons using an m-array coded light source," *Instrumentation and Measurement*, vol. 44, pp. 775–778, Jun 1995.
- [29] J. Forest, *New methods for triangulation-based shape acquisition using laser scanners*. PhD thesis, Universitat de Girona, Spain, 2004.

- [30] G. Sansoni, M. Trebeschi, and F. Docchio, "State-of-the-art and applications of 3d imaging sensors in industry, cultural heritage, medicine, and criminal investigation," *Sensors*, vol. 9, pp. 568–601, 2009.
- [31] O. Faugeras, *Three-dimensional Computer Vision : A Geometric Viewpoint*. Cambridge, MA, USA : MIT Press, 1993.
- [32] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. New York, NY, USA : Cambridge University Press, second ed., 2003.
- [33] H. A. Beyer, "Accurate calibration of ccd-cameras," *Computer Vision and Pattern Recognition*, pp. 96–101, 1992.
- [34] J. Weng, P. Cohen, and M. Herniou, "Camera calibration with distortion models and accuracy evaluation," *Pattern Analysis and Machine Intelligence*, vol. 14, pp. 965–980, 1992.
- [35] R. Y. Tsai, "A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses," *Robotics and Automation*, vol. 3, pp. 323–344, 1987.
- [36] O. D. Faugeras and G. Toscani, "The Calibration Problem for Stereo," *Computer Vision and Pattern Recognition*, pp. 15–20, 1986.
- [37] J. Bouguet, "Camera calibration toolbox for matlab." [http://www.vision.caltech.edu/bouguetj/calib\\_doc/index.html](http://www.vision.caltech.edu/bouguetj/calib_doc/index.html). [Online].
- [38] S. Audet, "Procamcalib." <http://www.ok.ctrl.titech.ac.jp/~saudet/research/procamcalib/>. [Online].
- [39] J. Salvi, X. Armangué, and J. Batlle, "A comparative review of camera calibrating methods with accuracy evaluation," *Pattern Recognition*, vol. 35, pp. 1617 – 1635, 2002.
- [40] Z. Zhang, "Camera calibration," *Emergin Topics in Computer Vision*, vol. 2, pp. 4–43, 2004.
- [41] F. Remondino and C. Fraser, "Digital camera calibration methods : considerations and comparisons," *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2006.
- [42] K. Perlin, "An image synthesizer," *SIGGRAPH'85*, vol. 19, pp. 287–296, 1985.
- [43] K. Perlin, "Improving noise," *ACM Trans. Graph.*, vol. 21, pp. 681–682, 2002.
- [44] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International Journal of Computer Vision*, vol. 47, pp. 7–42, 2002.

- [45] E. Castillo, J. Liang, and H. Zhao, *Point Cloud Segmentation and Denoising via Constrained Nonlinear Least Squares Normal Estimates*. Springer Berlin Heidelberg, 2013.
- [46] M. Kazhdan, M. Bolitho, and H. Hoppe, *Poisson Surface Reconstruction*. Eurographics Association, 2006.
- [47] F. Calderon, U. Ruiz, and M. Rivera, "Surface-normal estimation with neighborhood reorganization for 3d reconstruction," *Pattern Recognition*, pp. 321–330, 2007.
- [48] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, "Surface reconstruction from unorganized points," *SIGGRAPH'92*, vol. 26, pp. 71–78, 1992.
- [49] K. Pearson, "On lines and planes of closest fit to systems of points in space," *Philosophical Magazine*, vol. 2, no. 6, pp. 559–572, 1901.
- [50] H. Hotelling, "Analysis of a complex of statistical variables into principal components," *Journal of Educational Psychology*, vol. 24.
- [51] N. J. Mitra and A. Nguyen, "Estimating surface normals in noisy point cloud data," *SCG '03*, pp. 322–328, 2003.
- [52] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and C. T. Silva, "Point set surfaces," *Conference on Visualization'01*, pp. 21–28, 2001.
- [53] M. Berger, A. Tagliasacchi, L. M. Seversky, P. Alliez, J. A. Levine, A. Sharf, and C. Silva, "State of the art in surface reconstruction from point clouds," *Eurographics STAR*, 2014.
- [54] G. Demengel and J. Pouget, "Modèles de bezier, des b-splines et des nurbs : outils pour l'ingénieur, bases pour la cao," *Mathématiques des courbes et des surfaces*, 1998.
- [55] C. Roudet and F. Payan, "Remaillage semi-régulier pour les maillages surfaciques triangulaires : un état de l'art," *Revue Française d'Informatique Graphique*, vol. 5, 2011.
- [56] Z. Song and C. Chung, "Determining both surface position and orientation in structured-light-based sensing," *Pattern Analysis and Machine Intelligence*, vol. 32, pp. 1770–1780, 2010.
- [57] T. Surazhsky, E. Magid, O. Soldea, G. Elber, and E. Rivlin, "A comparison of gaussian and mean curvatures estimation methods on triangular meshes," *ICRA*, pp. 1021–1026, 2003.
- [58] T. D. Gatzke and C. M. Grimm, "Estimating curvature on triangular meshes," *International Journal of Shape Modeling*, 2006.

- [59] I. Douros and B. B., "Three-dimensional surface curvature estimation using quadric surface patches," *Scanning'02*, 2002.
- [60] C. Dong and G. Wang, "Curvatures estimation on triangular mesh," *Journal of Zhejiang*, 2005.
- [61] G. Taubin, "Estimating the tensor of curvature of a surface from a polyhedral approximation," *Computer Vision*, pp. 902–907, 1995.
- [62] C. Lange and K. Polthier, "Anisotropic smoothing of point sets," *Computer Aided Geometric Design*, vol. 22, pp. 680 – 692, 2005.
- [63] J. Berkmann and T. Caelli, "Computation of surface geometry and segmentation using covariance techniques," *Pattern Analysis and Machine Intelligence*, vol. 16, pp. 1114–1116, 1994.
- [64] Y. Zhang, J. Paik, A. Koschan, M. Abidi, and D. Gorsich, "Simple and efficient algorithm for part decomposition of 3-d triangulated models based on curvature analysis," *Image Processing*, vol. 3, 2002.
- [65] A. Mangan and R. Whitaker, "Partitioning 3d surface meshes using watershed segmentation," *Visualization and Computer Graphics*, vol. 5, pp. 308–321, 1999.
- [66] R. Liu and H. Zhang, "Segmentation of 3d meshes through spectral clustering," *Computer Graphics and Applications*, pp. 298–305, 2004.
- [67] L. Yu and X. Youlun, "Automatic segmentation of unorganized noisy point clouds based on the gaussian map," *Computer-Aided Design*, vol. 40, pp. 576 – 594, 2008.
- [68] S. Delest, R. Boné, and H. Cardot, "Etat de l'art de la segmentation de maillage 3D par patches surfaciques," *Groupe de Travail en Modélisation Géométrique*, pp. 171–185, 2007.
- [69] M. Pauly, M. Gross, and L. P. Kobbelt, "Efficient simplification of point-sampled surfaces," *VIS '02*, pp. 163–170, 2002.
- [70] C. Moenning and N. A. Dodgson, "Intrinsic point cloud simplification," *International Conference Graphicon*, 2006.
- [71] H. Song and H. Feng, "A progressive point cloud simplification algorithm with preserved sharp edge data," *The International Journal of Advanced Manufacturing Technology*, vol. 45, pp. 583–592, 2009.
- [72] J. Rossignac and P. Borrel, "Multi-resolution 3d approximations for rendering complex scenes," *Modeling in Computer Graphics*, pp. 455–465, 1993.
- [73] K. L. Low and T. Tan, "Model simplification using vertex-clustering," pp. 75–ff., 1997.
- [74] W. J. Schroeder, J. A. Zarge, and W. E. Lorensen, "Decimation of triangle meshes," *SIGGRAPH'92*, vol. 26, pp. 65–70, 1992.



- [75] M. Garland and P. S. Heckbert, "Surface simplification using quadric error metrics," *SIGGRAPH '97*, pp. 209–216, 1997.
- [76] J. Cohen, A. Varshney, D. Manocha, G. Turk, H. Weber, P. Agarwal, F. Brooks, and W. Wright, "Simplification envelopes," *SIGGRAPH '96*, pp. 119–128, 1996.
- [77] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, "Mesh optimization," *SIGGRAPH '93*, pp. 19–26, 1993.
- [78] S. Valette and R. Prost, "Wavelet-based multiresolution analysis of irregular surface meshes," *Visualization and Computer Graphics*, vol. 10, pp. 113–122, 2004.
- [79] J. L. Peyrot, F. Payan, and M. Antonini, "Simplification de maillages cao avec échantillonnage anti-aliasing," *Groupe de Recherche et d'Etudes du Traitement du Signal et des Images*, 2013.
- [80] P. Cignoni, C. Montani, and R. Scopigno, "A comparison of mesh simplification algorithms," *Computers & Graphics*, vol. 22, pp. 37–54, 1997.
- [81] M. A. Fischler and R. C. Bolles, "Random sample consensus : A paradigm for model fitting with applications to image analysis and automated cartography," *ACM*, vol. 24, pp. 381–395, 1981.
- [82] C. Beder and W. Forstner, "Direct solutions for computing cylinders from minimal sets of 3d points," *Computer Vision*, pp. 135–146, 2006.
- [83] R. Schnabel, R. Wahl, and R. Klein, "Efficient ransac for point-cloud shape detection," *Computer Graphics*, vol. 26, pp. 214–226, 2007.
- [84] F. Bretar and M. Roux, "Hybrid image segmentation using lidar 3d planar primitives," *ISPRS Workshop Laser scanning*, pp. 72–78, 2005.
- [85] P. Hough, "Method and Means for Recognizing Complex Patterns," 1962.
- [86] T. Rabbani, *Automatic reconstruction of industrial installations using point clouds and images*. PhD thesis, Civil Engineering and Geosciences, England, 2006.
- [87] P. Benkő, R. R. Martin, and T. Varady, "Algorithms for reverse engineering boundary representation models," *Computer-Aided Design*, vol. 33, pp. 839 – 851, 2001.
- [88] V. B. Sunil and S. S. Pande, "Automatic recognition of features from freeform surface cad models," *Computer-Aided Design*, vol. 40, pp. 502 – 517, 2008.
- [89] R. Bénéière, *Reconstruction d'un modèle B-Rep à partir d'un maillage 3D : Application à la rétro-ingénierie industrielle*. PhD thesis, 2012.
- [90] K. Gauss, "General investigations curved surfaces surfaces," *English translation*, 1965.
- [91] G. Lukacs, A. D. Marshall, and R. R. Martin, "Geometric least-squares fitting of spheres, cylinders, cones and tori," *Deliverable Report RECCAD*, 1997.

- [92] A. Bey, R. Chaine, R. Marc, G. Thibault, and S. Akkouche, "Consistent 3d model reconstruction from point cloud data using a priori cad models," *Laser Scanning 2011*, 2011.
- [93] T. Chaperon and F. Goulette, "Extracting cylinders in full 3d data using a random sampling method and the gaussian image," *Vision Modeling and Visualization Conference*, 2001.
- [94] T. Chaperon, *Segmentation de nuage de points 3D pour la modélisation automatique d'environnements industriels numérisés*. PhD thesis, 2002.
- [95] G. Slabaugh, R. Schafer, and M. Livingston, "Optimal ray intersection for computing 3d points from n-view correspondences," *Deliverable Report*, 2001.
- [96] R. Ohlander, *Analysis of Natural Scenes*. PhD thesis, Pittsburgh, PA, USA, 1975.
- [97] P. C. Chen and T. Pavlidis, "Segmentation by texture using a co-occurrence matrix and a split-and-merge algorithm," *Computer Graphics and Image Processing*, vol. 10, no. 2, pp. 172 – 182, 1979.
- [98] K. Held, E. Kops, B. Krause, W. Wells, R. Kikinis, and H. Muller-Gartner, "Markov random field segmentation of brain mr images," *Medical Imaging, IEEE Transactions on*, vol. 16, pp. 878–886, Dec 1997.
- [99] D. Mumford and J. Shah, "Optimal approximations by piecewise smooth functions and associated variational problems," *Communications on Pure and Applied Mathematics*, vol. 42, no. 5, pp. 577–685, 1989.
- [100] A. Forbes, *Least squares best fit geometric elements*. Mason, J.C. and Cox, M.G., 1990.
- [101] P. Bezier, *Emploi des Machines à Commande Numérique*. Masson et Cie, 1970.
- [102] P. Bézier, "Courbes et surfaces," *Hermes*, 1996.
- [103] R. H. Bartels, J. C. Beatty, and B. A. Barsky, *An Introduction to Splines for Use in Computer Graphics & Geometric Modeling*. Morgan Kaufmann Publishers Inc., 1987.
- [104] V. Paquit, *Imagerie multispectrale et modélisation 3D pour l'estimation quantitative des vaisseaux sanguins sous cutanés*. PhD thesis, LE2I, Le Creusot, France, 2008.
- [105] W. Boehm and A. Muller, "On de casteljau's algorithm," *Computer Aided Geometric Design*, vol. 16, no. 7, pp. 587 – 605, 1999.
- [106] R. Bénéière, G. Subsol, W. Puech, G. Gesquière, and F. Lebreton, "Decomposition of a 3d triangular mesh into quadrangulated patches," *GRAPP*, pp. 96–103, 2010.
- [107] S. Valette, C. J. M., and R. Prost, "Acvd : Surface mesh coarsening and resampling." <http://www.creatis.insa-lyon.fr/site/en/acvd#ref1>. [Online].

- [108] T. Chaperon, *Segmentation de nuage de points 3D pour la modélisation automatique d'environnements industriels numérisés*. PhD thesis, Ecole Nationale Supérieure des Mines de Paris, 2002.

## ANNEXES



## ANNEXE A

Comme nous l'avons déjà mentionné précédemment, nous sommes en mesure d'identifier les différentes primitives dans la scène en exploitant les résultats fournis par l'ACP sur la sphère de Gauss. Dans cette partie, nous cherchons à démontrer que ces résultats peuvent également être utilisés pour modéliser les primitives. En effet, la matrice de covariance est par construction symétrique, ce qui implique que ses vecteurs propres forment une base orthogonale. Comme présenté dans la section 3.3.2.2, nous cherchons à vérifier que  $U$  (axe unitaire d'une surface de révolution) est toujours l'un des vecteurs propres de la matrice de covariance, ce qui nous permettrait d'utiliser le plan formé par les deux autres vecteurs propres pour modéliser les primitives.

### NOTATIONS

Dans un espace euclidien à 3 dimensions muni d'un repère orthonormé, on suppose qu'une normale quelconque à une surface de révolution définie par un vecteur unitaire

$$U = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}, \quad (1)$$

est notée

$$n_i = \begin{bmatrix} n_{1i} \\ n_{2i} \\ n_{3i} \end{bmatrix}. \quad (2)$$

Ces vecteurs représentant les  $K$  normales observées sont rassemblés dans une matrice de la façon suivante :

$$N = \begin{bmatrix} n_{11} & \dots & n_{1i} & \dots & n_{1K} \\ n_{21} & \dots & n_{2i} & \dots & n_{2K} \\ n_{31} & \dots & n_{3i} & \dots & n_{3K} \end{bmatrix}. \quad (3)$$

Ces coefficients sont centrés par rapport à la moyenne suivant chaque composante

$$\overline{n_1} = \frac{\sum_{i=1}^K n_{1i}}{K}, \quad \overline{n_2} = \frac{\sum_{i=1}^K n_{2i}}{K}, \quad \overline{n_3} = \frac{\sum_{i=1}^K n_{3i}}{K}, \quad (4)$$

pour former la matrice des normales centrées :

$$N' = \begin{bmatrix} n_{11} - \overline{n_1} & \dots & n_{1i} - \overline{n_1} & \dots & n_{1K} - \overline{n_1} \\ n_{21} - \overline{n_2} & \dots & n_{2i} - \overline{n_2} & \dots & n_{2K} - \overline{n_2} \\ n_{31} - \overline{n_3} & \dots & n_{3i} - \overline{n_3} & \dots & n_{3K} - \overline{n_3} \end{bmatrix}. \quad (5)$$

Les coefficients de cette matrice peuvent donc se mettre sous la forme suivante :

$$n'_{ij} = n_{ij} - \overline{n_i}. \quad (6)$$

L'analyse en composantes principales consiste à calculer la matrice de covariance des données centrées. De part sa définition, la matrice de covariance est définie par :

$$C = N'N'^t. \quad (7)$$

Il s'agit d'une matrice  $3 \times 3$  dont les coefficients sont les suivants :

$$c_{ij} = \sum_{k=1}^K n'_{ik} n'_{kj}. \quad (8)$$

La matrice de covariance est par construction symétrique, ses vecteurs propres forment donc une base orthogonale. On note  $\alpha$  l'angle entre la normale et le vecteur directeur de l'axe de symétrie soit

$$n_i^t U = U^t n_i = \cos \alpha_i. \quad (9)$$

L'objectif de l'analyse en composantes principales consiste à chercher les trois valeurs propres  $\lambda$  et les trois vecteurs propres  $X$  associés à la matrice de covariance :

$$\det(C - \lambda I) = 0 \quad (10)$$

$$CX = \lambda X \quad (11)$$

## CAS DES DONNÉES NON BRUITÉES

On cherche à vérifier que  $U$  est toujours l'un de ces vecteurs propres, associé à la plus petite des valeurs propres : 0. Dans le cas de données non bruitées, il est possible d'écrire :

$$CU = (N'N'')U = N'(N''U), \quad (12)$$

$$N''U = \begin{bmatrix} n_{11} - \bar{n}_1 & n_{21} - \bar{n}_2 & n_{31} - \bar{n}_3 \\ \vdots & \vdots & \vdots \\ n_{1i} - \bar{n}_1 & n_{2i} - \bar{n}_2 & n_{3i} - \bar{n}_3 \\ \vdots & \vdots & \vdots \\ n_{1K} - \bar{n}_1 & n_{2K} - \bar{n}_2 & n_{3K} - \bar{n}_3 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}, \quad (13)$$

$$N''U = \begin{bmatrix} (n_{11} - \bar{n}_1)u_1 + (n_{21} - \bar{n}_2)u_2 + (n_{31} - \bar{n}_3)u_3 \\ \vdots \\ (n_{1i} - \bar{n}_1)u_1 + (n_{2i} - \bar{n}_2)u_2 + (n_{3i} - \bar{n}_3)u_3 \\ \vdots \\ (n_{1K} - \bar{n}_1)u_1 + (n_{2K} - \bar{n}_2)u_2 + (n_{3K} - \bar{n}_3)u_3 \end{bmatrix}, \quad (14)$$

donc

$$N''U = \begin{bmatrix} \cos \alpha_1 - (\bar{n}_1 u_1 + \bar{n}_2 u_2 + \bar{n}_3 u_3) \\ \vdots \\ \cos \alpha_i - (\bar{n}_1 u_1 + \bar{n}_2 u_2 + \bar{n}_3 u_3) \\ \vdots \\ \cos \alpha_K - (\bar{n}_1 u_1 + \bar{n}_2 u_2 + \bar{n}_3 u_3) \end{bmatrix}. \quad (15)$$

Si on note  $\bar{\alpha}$  l'angle entre la moyenne des normales et l'axe de symétrie, alors

$$N''U = \begin{bmatrix} \cos \alpha_1 \\ \vdots \\ \cos \alpha_i \\ \vdots \\ \cos \alpha_K \end{bmatrix} - \cos \bar{\alpha} \begin{bmatrix} 1 \\ \vdots \\ 1 \\ \vdots \\ 1 \end{bmatrix}. \quad (16)$$

**Remarque** : dans le cas d'un cône toutes les normales font le même angle avec l'axe de symétrie, on a alors :

$$N''U = 0, \quad (17)$$



et on vérifie immédiatement que

$$CU = N' (N'' U) = 0. \quad (18)$$

Dans ce cas,  $U$  est bien un vecteur propre associé à la valeur propre 0. Dans le cas général  $CU$  s'écrit :

$$CU = \begin{bmatrix} n_{11} - \bar{n}_1 & \cdots & n_{1i} - \bar{n}_1 & \cdots & n_{1K} - \bar{n}_1 \\ n_{21} - \bar{n}_2 & \cdots & n_{2i} - \bar{n}_2 & \cdots & n_{2K} - \bar{n}_2 \\ n_{31} - \bar{n}_3 & \cdots & n_{3i} - \bar{n}_3 & \cdots & n_{3K} - \bar{n}_3 \end{bmatrix} \begin{bmatrix} \cos \alpha_1 \\ \vdots \\ \cos \alpha_i \\ \vdots \\ \cos \alpha_K \end{bmatrix} - \cos \bar{\alpha} \begin{bmatrix} n_{11} - \bar{n}_1 & \cdots & n_{1i} - \bar{n}_1 & \cdots & n_{1K} - \bar{n}_1 \\ n_{21} - \bar{n}_2 & \cdots & n_{2i} - \bar{n}_2 & \cdots & n_{2K} - \bar{n}_2 \\ n_{31} - \bar{n}_3 & \cdots & n_{3i} - \bar{n}_3 & \cdots & n_{3K} - \bar{n}_3 \end{bmatrix} \begin{bmatrix} 1 \\ \vdots \\ 1 \\ \vdots \\ 1 \end{bmatrix}. \quad (19)$$

Comme les données sont centrées, le deuxième terme est nul et il ne reste que :

$$CU = \begin{bmatrix} (n_{11} - \bar{n}_1) \cos \alpha_1 + \cdots + (n_{1i} - \bar{n}_1) \cos \alpha_i + \cdots + (n_{1K} - \bar{n}_1) \cos \alpha_K \\ (n_{21} - \bar{n}_2) \cos \alpha_1 + \cdots + (n_{2i} - \bar{n}_2) \cos \alpha_i + \cdots + (n_{2K} - \bar{n}_2) \cos \alpha_K \\ (n_{31} - \bar{n}_3) \cos \alpha_1 + \cdots + (n_{3i} - \bar{n}_3) \cos \alpha_i + \cdots + (n_{3K} - \bar{n}_3) \cos \alpha_K \end{bmatrix} \quad (20)$$

$$CU = \begin{bmatrix} \sum_{i=1}^K (n_{1i} \cos \alpha_i) - \bar{n}_1 \sum_{i=1}^K \cos \alpha_i \\ \sum_{i=1}^K (n_{2i} \cos \alpha_i) - \bar{n}_2 \sum_{i=1}^K \cos \alpha_i \\ \sum_{i=1}^K (n_{3i} \cos \alpha_i) - \bar{n}_3 \sum_{i=1}^K \cos \alpha_i \end{bmatrix}, \quad (21)$$

donc

$$CU = \sum_{i=1}^K \left( \cos \alpha_i \begin{bmatrix} n_{1i} \\ n_{2i} \\ n_{3i} \end{bmatrix} \right) - \begin{bmatrix} \bar{n}_1 \\ \bar{n}_2 \\ \bar{n}_3 \end{bmatrix} \sum_{i=1}^K \cos \alpha_i \quad (22)$$

$$CU = \sum_{i=1}^K (\cos \alpha_i n_i) - \bar{n} \sum_{i=1}^K \cos \alpha_i. \quad (23)$$

Si le processus de mesure des normales est stochastique, le calcul de la somme des

cosinus est un estimateur de l'espérance de ce cosinus. Si l'estimateur n'est pas biaisé, l'espérance de la somme est la somme des espérances si les mesures peuvent être supposées indépendantes les unes des autres. Dans ce cas l'espérance des cosinus est le cosinus de l'espérance de l'angle, vérifiant ainsi que  $U$  est bien un vecteur propre associé à la valeur propre 0. Cependant, ce résultat est loin de pouvoir s'appliquer dans le cas qui nous concerne, étant donné la nature bruitée des données.

## CAS DU CÔNE OU DU CYLINDRE AVEC DES DONNÉES BRUITÉES

On se place dans le cas du cône (ou du cylindre). Si les normales sont mesurées sans erreur, l'angle  $\alpha$  est constant et l'axe de symétrie porte le vecteur propre associé à la valeur propre nulle.

Le repère des vecteurs propres étant orthogonal, **les deux autres vecteurs propres sont dans un plan perpendiculaire à l'axe de symétrie du cône** dans la direction qui maximise la variance pour l'un d'entre eux. Si les données sont bruitées, deux questions se posent : l'axe de symétrie est-il toujours un vecteur propre ? et si oui, qu'elle est la valeur propre associée ?

Reprenons le calcul précédent, en supposant que l'angle de chaque normale est mesuré avec une petite erreur  $\varepsilon_i$  comprise entre  $-\varepsilon_m$  et  $\varepsilon_m$  suivant une loi d'espérance nulle. La figure 2 précise les notations utilisées par la suite sur laquelle l'axe de symétrie est porté par le vecteur unitaire  $U$ .

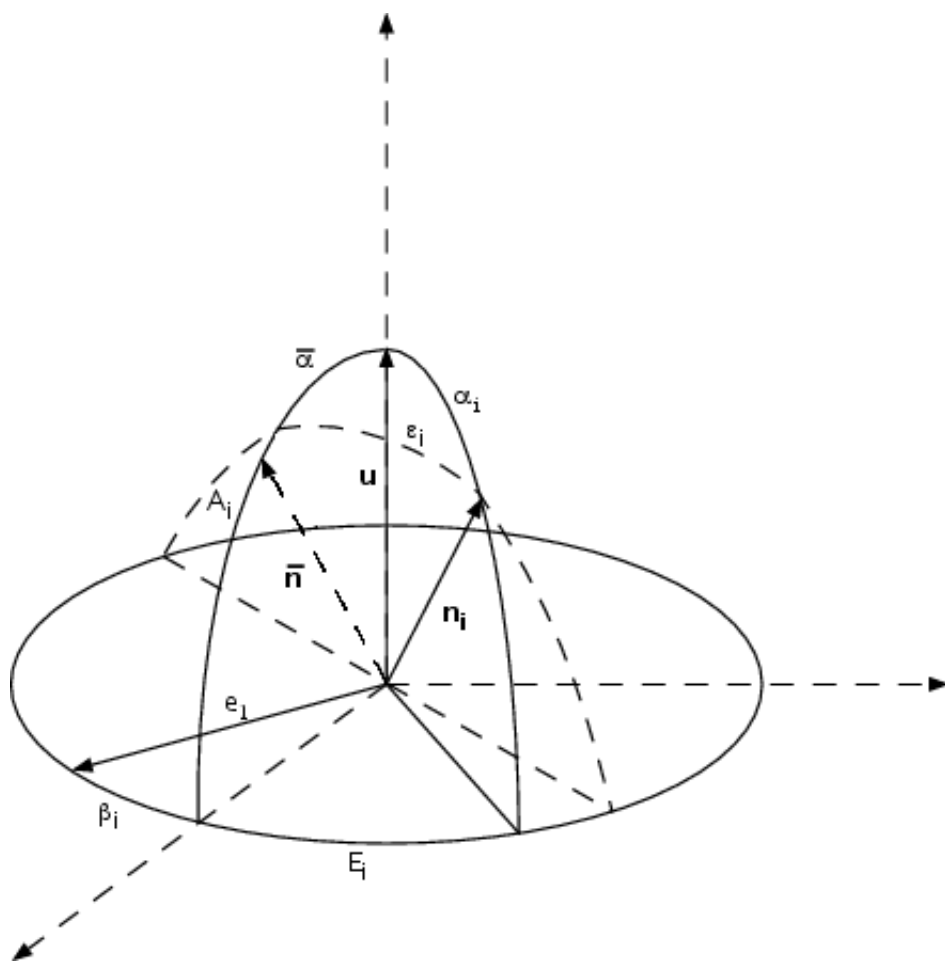


FIGURE 2 – Représentation de la normale par rapport à l'axe de symétrie.

On appelle  $A_i$  l'angle qui définit la position angulaire de la normale mesurée autour de la normale vraie, c'est aussi une variable aléatoire qui est comprise entre 0 et  $\pi$  et qui suit une loi uniforme. L'angle  $\beta_i$  repère la position de la normale vraie (sans erreur de mesure) au point considéré. On admet que cet angle varie de façon aléatoire autour de zéro de sorte que les points de mesure soient répartis autour du plan qui sert d'origine pour le repère. L'angle  $E_i$  est l'angle opposé (dans le triangle sphérique) de  $\epsilon_i$ , c'est donc une variable aléatoire liée à  $\alpha_i$  et  $\epsilon_i$ . Les relations du triangle sphérique donnent :

$$\cos \alpha_i = \cos \alpha \cos \epsilon_i + \sin \alpha \sin \epsilon_i \cos A_i. \quad (24)$$

Si on suppose que l'erreur de mesure  $\epsilon_i$  reste petite, on peut faire un développement au premier ordre :

$$\cos \alpha_i \simeq \cos \alpha + \epsilon_i \sin \alpha \cos A_i. \quad (25)$$

Dans ce cas l'équation 16, dans laquelle  $\bar{\alpha} = \alpha$  (puisque on a un cône) s'écrit :

$$N''U = \cos \alpha \begin{bmatrix} 1 \\ \vdots \\ 1 \\ \vdots \\ 1 \end{bmatrix} + \sin \alpha \begin{bmatrix} \varepsilon_1 \cos A_1 \\ \vdots \\ \varepsilon_i \cos A_i \\ \vdots \\ \varepsilon_K \cos A_K \end{bmatrix} - \cos \alpha \begin{bmatrix} 1 \\ \vdots \\ 1 \\ \vdots \\ 1 \end{bmatrix} \quad (26)$$

$$N''U = \sin \alpha \begin{bmatrix} \varepsilon_1 \cos A_1 \\ \vdots \\ \varepsilon_i \cos A_i \\ \vdots \\ \varepsilon_K \cos A_K \end{bmatrix} \quad (27)$$

et

$$CU = \sin \alpha \begin{bmatrix} n_{11} - \bar{n}_1 & \cdots & n_{1i} - \bar{n}_1 & \cdots & n_{1K} - \bar{n}_1 \\ n_{21} - \bar{n}_2 & \cdots & n_{2i} - \bar{n}_2 & \cdots & n_{2K} - \bar{n}_2 \\ n_{31} - \bar{n}_3 & \cdots & n_{3i} - \bar{n}_3 & \cdots & n_{3K} - \bar{n}_3 \end{bmatrix} \begin{bmatrix} \varepsilon_1 \cos A_1 \\ \vdots \\ \varepsilon_i \cos A_i \\ \vdots \\ \varepsilon_K \cos A_K \end{bmatrix} \quad (28)$$

$$CU = \sin \alpha \begin{bmatrix} \sum_{i=1}^K (n_{1i} \varepsilon_i \cos A_i) - \bar{n}_1 \sum_{i=1}^K \varepsilon_i \cos A_i \\ \sum_{i=1}^K (n_{2i} \varepsilon_i \cos A_i) - \bar{n}_2 \sum_{i=1}^K \varepsilon_i \cos A_i \\ \sum_{i=1}^K (n_{3i} \varepsilon_i \cos A_i) - \bar{n}_3 \sum_{i=1}^K \varepsilon_i \cos A_i \end{bmatrix}. \quad (29)$$

On suppose que le nombre de mesures est suffisant et qu'il n'y a pas de biais, de sorte que la somme est un estimateur non biaisé de l'espérance. La variable aléatoire  $\cos A_i$  suit une loi de probabilité comprise entre  $-1$  et  $+1$  et qui est symétrique (le cosinus est une fonction paire), son espérance est donc nulle. Les deux variables  $\varepsilon_i$  et  $\cos A_i$  sont indépendantes donc l'espérance du produit est le produit des espérances. Les éléments du deuxième terme de l'équation précédente sont donc nuls et on a :

$$CU = \sin \alpha \begin{bmatrix} \sum_{i=1}^K n_{1i} \varepsilon_i \cos A_i \\ \sum_{i=1}^K n_{2i} \varepsilon_i \cos A_i \\ \sum_{i=1}^K n_{3i} \varepsilon_i \cos A_i \end{bmatrix} \quad (30)$$

$$CU = \sin \alpha \sum_{i=1}^K \varepsilon_i \cos A_i n_i. \quad (31)$$

Remarque :  $n_i$  et les deux variables  $\cos A_i$  et  $\varepsilon_i$  ne sont pas indépendantes, il n'y a donc pas de raison pour que la somme de leur produit soit nul. Si  $U$  reste un vecteur propre, alors

$$CU = \sin \alpha \sum_{i=1}^K \varepsilon_i \cos A_i n_i = \lambda U. \quad (32)$$

Peut-on montrer que si l'erreur angulaire est distribuée de façon isotrope autour de la normale, alors  $\sum_{i=1}^K \varepsilon_i \cos A_i n_i$  est colinéaire avec  $U$  ?

On peut faire le calcul dans un repère unique lié au centre de la sphère de Gauss, le repère orthonormé associé est tel que  $e_3$  est confondu avec l'axe de symétrie  $U$  (repère de la figure 2), de sorte que :

$$U = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (33)$$

et

$$n_i = \begin{bmatrix} \sin \alpha_i \cos(E_i + \beta_i) \\ -\sin \alpha_i \sin(E_i + \beta_i) \\ \cos \alpha_i \end{bmatrix}. \quad (34)$$

La trigonométrie du triangle sphérique donne la relation suivante (figure2) :

$$\sin E_i = \frac{\sin \varepsilon_i \sin A_i}{\sin \alpha_i}. \quad (35)$$

Donc, au premier ordre

$$\begin{aligned}\sin E_i &= \frac{\varepsilon_i \sin A_i}{\sin \alpha} \\ \cos E_i &= 1\end{aligned}\tag{36}$$

donc

$$\begin{aligned}\cos(E_i + \beta_i) &= \cos E_i \cos \beta_i - \sin E_i \sin \beta_i \\ \cos(E_i + \beta_i) &= \cos \beta_i - \frac{\varepsilon_i \sin A_i}{\sin \alpha} \sin \beta_i\end{aligned}\tag{37}$$

et

$$\begin{aligned}\sin(E_i + \beta_i) &= \cos E_i \sin \beta_i + \sin E_i \cos \beta_i \\ \sin(E_i + \beta_i) &= \sin \beta_i + \frac{\varepsilon_i \sin A_i}{\sin \alpha} \cos \beta_i\end{aligned}\tag{38}$$

donc

$$\sum_{i=1}^K \varepsilon_i \cos A_i n_i = \begin{bmatrix} \sum_{i=1}^K \varepsilon_i \sin \alpha_i \cos A_i \left( \cos \beta_i - \frac{\varepsilon_i \sin A_i}{\sin \alpha} \sin \beta_i \right) \\ - \sum_{i=1}^K \varepsilon_i \sin \alpha_i \cos A_i \left( \sin \beta_i + \frac{\varepsilon_i \sin A_i}{\sin \alpha} \cos \beta_i \right) \\ \sum_{i=1}^K \varepsilon_i \cos \alpha_i \cos A_i \end{bmatrix}.\tag{39}$$

Pour montrer que c'est un vecteur propre il faut montrer que les deux premières composantes sont nulles. Si  $\varepsilon_i$  reste petit on peut faire, comme pour  $\cos \alpha_i$ , une estimation au premier ordre de  $\sin \alpha_i$ . En effet

$$\begin{aligned}\sin \alpha_i &= \sqrt{1 - \cos^2 \alpha_i} \\ \sin \alpha_i &\simeq \sqrt{1 - (\cos \alpha + \varepsilon_i \sin \alpha \cos A_i)^2} \\ \sin \alpha_i &\simeq \sqrt{\sin^2 \alpha - 2\varepsilon_i \cos \alpha \sin \alpha \cos A_i} \\ \sin \alpha_i &\simeq \sin \alpha \sqrt{1 - 2\varepsilon_i \frac{\cos \alpha}{\sin \alpha} \cos A_i} \\ \sin \alpha_i &\simeq \sin \alpha \left( 1 - \varepsilon_i \frac{\cos \alpha}{\sin \alpha} \cos A_i \right)\end{aligned}$$

donc finalement

$$\sin \alpha_i \simeq \sin \alpha - \varepsilon_i \cos \alpha \cos A_i.\tag{40}$$

La première composante s'écrit ainsi

$$\sum_{i=1}^K \varepsilon_i \sin \alpha_i \cos A_i \cos(E_i + \beta_i) \simeq \sum_{i=1}^K \varepsilon_i (\sin \alpha - \varepsilon_i \cos \alpha \cos A_i) \cos A_i \left( \cos \beta_i - \frac{\varepsilon_i \sin A_i}{\sin \alpha} \sin \beta_i \right)$$

soit

$$\sum_{i=1}^K \left( \varepsilon_i \sin \alpha \cos A_i \cos \beta_i + \varepsilon_i^2 \cos A_i \sin A_i \sin \beta_i - \varepsilon_i^2 \cos \alpha \cos^2 A_i \cos \beta_i + \varepsilon_i^3 \cos \alpha \cos^2 A_i \frac{\sin A_i}{\sin \alpha} \sin \beta_i \right).$$

Les processus  $A$ ,  $\varepsilon$  et  $\beta$  sont indépendants et de moyennes nulles, donc les termes en  $\sin \beta_i$  sont de moyenne nulle, de même que le terme en  $\varepsilon_i^3$ , il ne reste qu'un terme

$$\sin \alpha \sum_{i=1}^K \varepsilon_i \cos A_i n_{i1} = - \sin \alpha \cos \alpha \sum_{i=1}^K \varepsilon_i^2 \cos^2 A_i \cos \beta_i. \quad (41)$$

Un calcul identique pour la deuxième composante conduit à

$$\sum_{i=1}^K \varepsilon_i \sin \alpha_i \cos A_i \left( \sin \beta_i + \frac{\varepsilon_i \sin A_i}{\sin \alpha} \cos \beta_i \right) = \sum_{i=1}^K \varepsilon_i (\sin \alpha - \varepsilon_i \cos \alpha \cos A_i) \cos A_i \left( \sin \beta_i + \frac{\varepsilon_i \sin A_i}{\sin \alpha} \cos \beta_i \right).$$

Le seul terme non directement nul est donc :

$$\sum_{i=1}^K \varepsilon_i^2 \cos A_i \sin A_i \cos \beta_i = \sum_{i=1}^K \frac{\varepsilon_i^2 \sin 2A_i \cos \beta_i}{2}. \quad (42)$$

Ce terme est finalement nul car la moyenne de  $\sin 2A_i$  sur  $0, \pi$  est nulle. Le calcul de la troisième composante donne

$$\begin{aligned} \sum_{i=1}^K \varepsilon_i \cos \alpha_i \cos A_i &= \sum_{i=1}^K \varepsilon_i \cos A_i (\cos \alpha + \varepsilon_i \sin \alpha \cos A_i) \\ \sum_{i=1}^K \varepsilon_i \cos \alpha_i \cos A_i &= \cos \alpha \sum_{i=1}^K \varepsilon_i \cos A_i + \sin \alpha \sum_{i=1}^K \varepsilon_i^2 \cos^2 A_i \\ &= \sin \alpha \sum_{i=1}^K \varepsilon_i^2 \cos^2 A_i \end{aligned}$$

En résumé, si  $K$  est suffisamment grand et si le processus est ergodique, le premier terme, proportionnel à la variance du bruit, peut être supposé nul (voir ci-dessous) et le deuxième est nul. **En toute rigueur, la première composante n'est donc pas nulle si le bruit de mesure ne l'est pas. Donc l'axe de symétrie du cône n'est pas un vecteur propre.** Cependant, le bruit sur les mesures étant faible, on peut ainsi admettre que  $U$  est approximativement un vecteur propre et la valeur propre qui lui est associée est

$$\lambda = K \sin^2 \alpha \text{var}(\varepsilon). \quad (43)$$

## CAS PARTICULIERS

Dans le cas particulier du **cylindre**,  $\alpha = \frac{\pi}{2}$  et

$$\lambda^2 = Kvar(\varepsilon). \quad (44)$$

Pour un **plan**,  $\alpha = 0$ , l'axe de symétrie est perpendiculaire au plan et est donc porté par les normales, dans le repère de la sphère gaussienne, l'axe de symétrie est donc porté par un rayon suivant la direction de la normale au plan et la valeur propre correspondante est nulle comme dans le cas non bruité. Les deux autres vecteurs principaux sont dans un plan perpendiculaire à ce rayon (parallèle au plan tangent à la sphère) passant par le barycentre des normales projetées sur la sphère.

$$\lambda = 0$$

## CONCLUSION

En définitif, dans le cas de données bruitées, il s'avère que le vecteur propre associé à la plus petite valeur propre n'est pas colinéaire à l'axe de la forme de révolution. Cependant, comme nous l'avons montré en section 3.1.3, dans notre cas le bruit de mesure est très faible. De ce fait, il est toutefois possible d'approximer le vecteur propre associé à la plus petite valeur propre comme étant l'axe de la surface de révolution et ainsi utiliser les vecteurs et valeurs propres pour modéliser les primitives.





## ANNEXE B

Dans cette partie, nous présentons d'autres résultats sur d'autres objets manufacturés. Ces résultats sont présentés de la même façon que dans ce document, selon deux paramétrages différents :

### Cas 1

- Distance tolérée entre la primitive/surface et les points 3D = 0.1 mm.
- Ratio primitive = 90%.
- Ratio surface = 90%.
- Densité initiale  $d_0 = 1/5$ .
- Densité finale  $d_f = 1/2$ .

### Cas 2

- Distance tolérée entre la primitive/surface et les points 3D = 0.1 mm.
- Ratio primitive = 75%.
- Ratio surface = 90%.
- Densité initiale  $d_0 = 1/5$ .
- Densité finale  $d_f = 1/2$ .

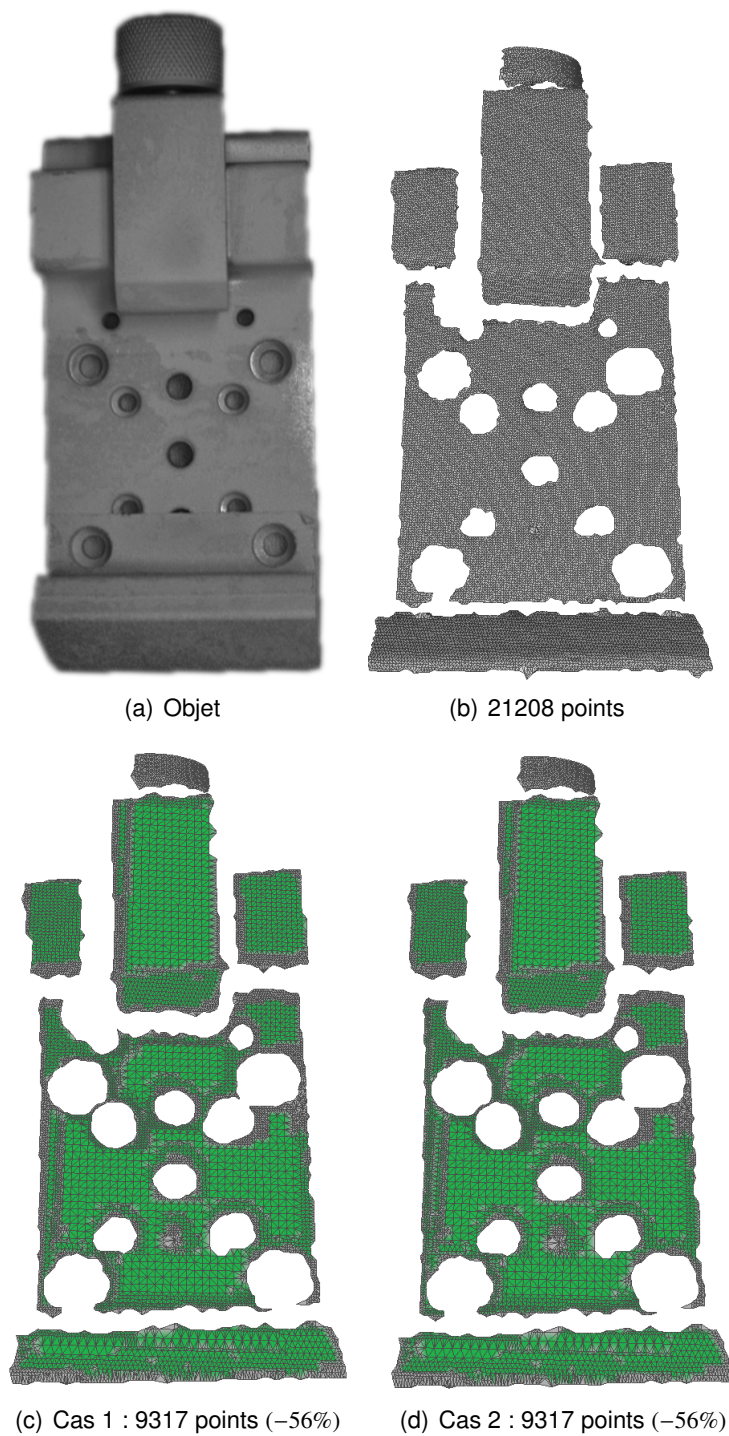


FIGURE 3 – Résultats sur une pièce manufacturée.

La figure 3 présente les résultats obtenus sur un objet composé de nombreuses zones planes. Dans ce cas de figure, le seuil sur le ratio de la primitive n'influence pas les résultats.

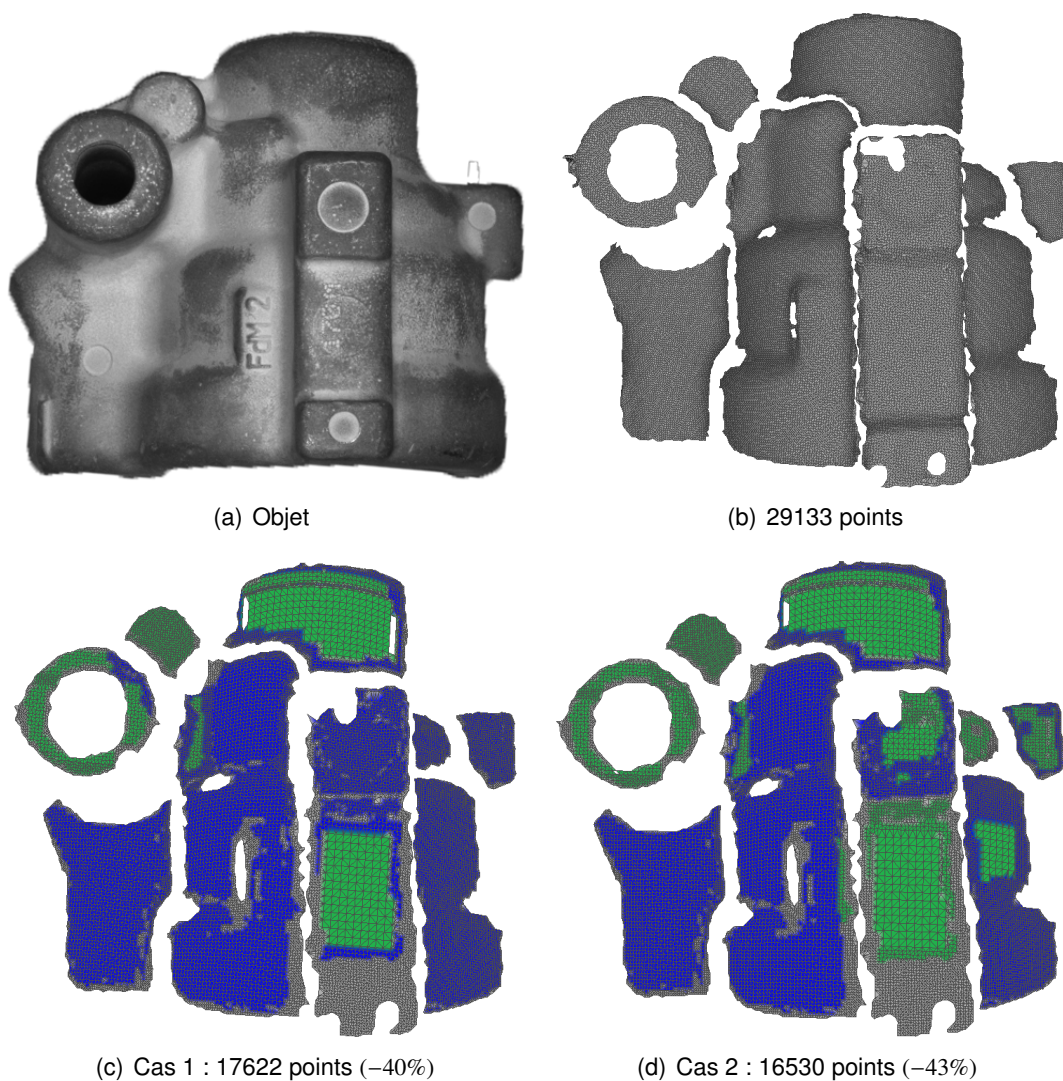


FIGURE 4 – *Résultats sur une pièce manufacturée.*

L'objet présenté en figure 4 est composé de formes assez variées. Visuellement, on distingue assez peu de primitives. Dans ce cas, baisser le ratio des primitives (cas 2) permet de détecter un peu plus de primitives et donc de simplifier un peu plus les données.



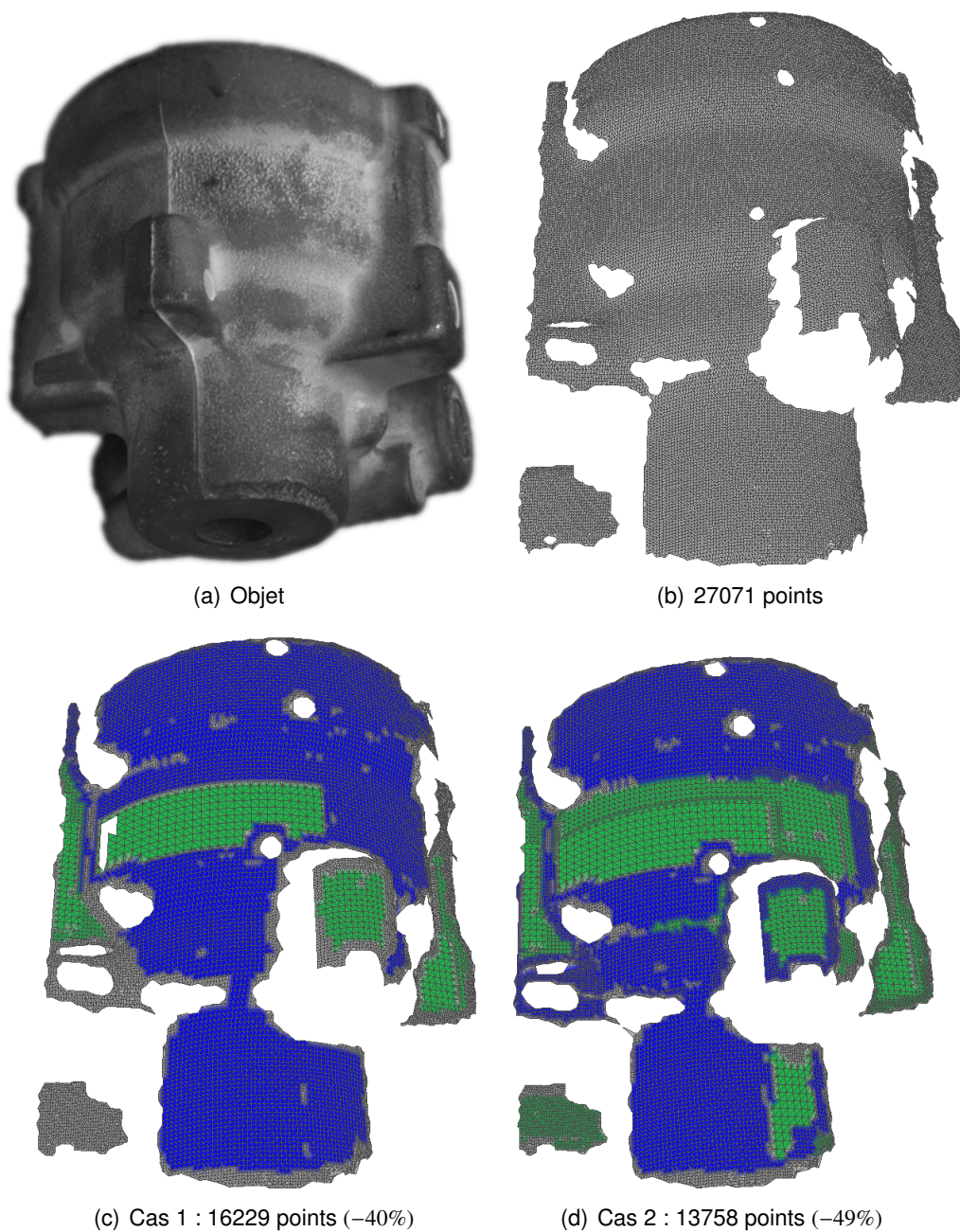


FIGURE 5 – *Résultats sur une pièce manufacturée.*

Sur l'objet présenté en figure 5, le deuxième cas permet de détecter plus de primitives, permettant ainsi de simplifier assez fortement les données par rapport au cas 1.

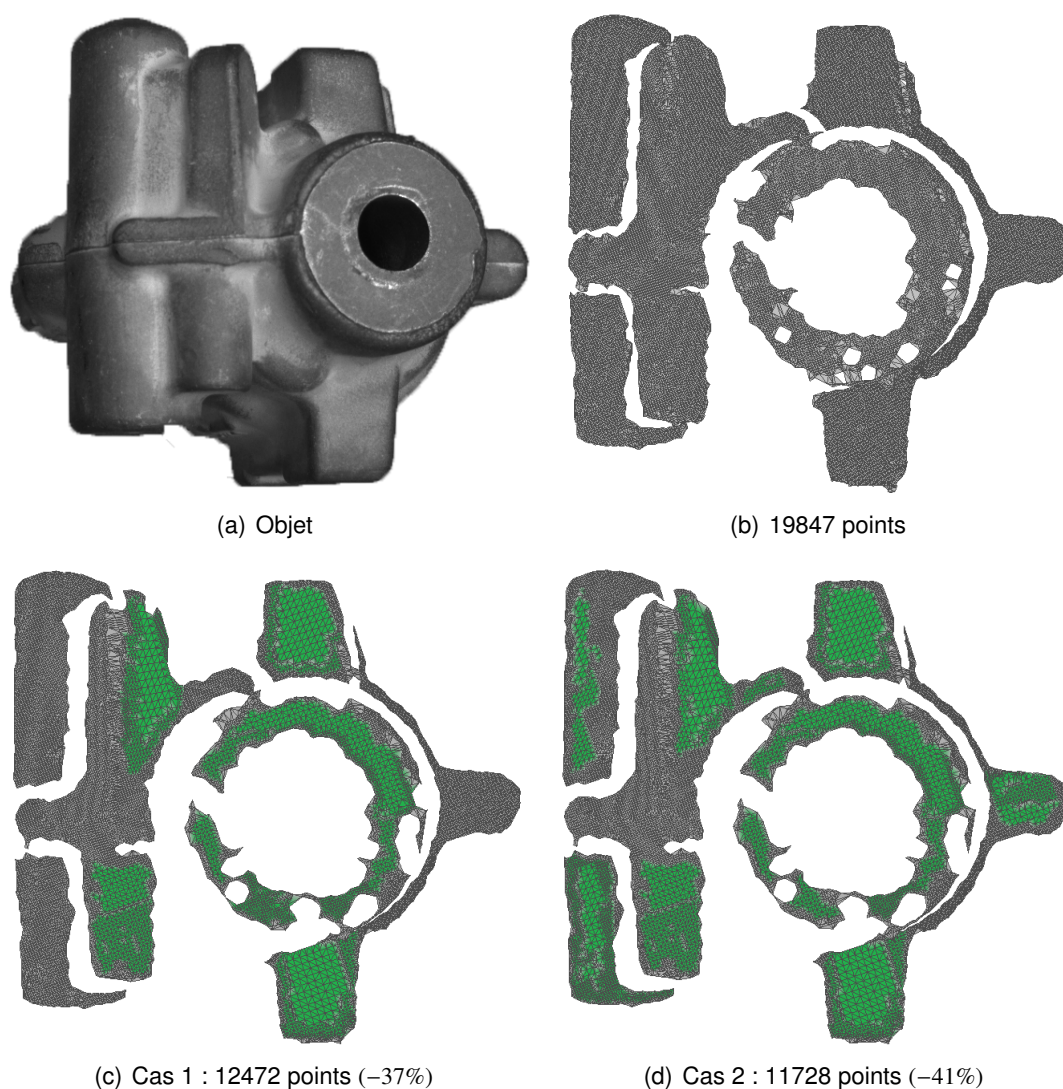


FIGURE 6 – Résultats sur une pièce manufacturée.

Sur cet objet (figure 6), la plupart des primitives peuvent être extraites directement depuis le cas 1. De ce fait, le cas 2 n'apporte qu'une faible simplification par rapport au cas 1.

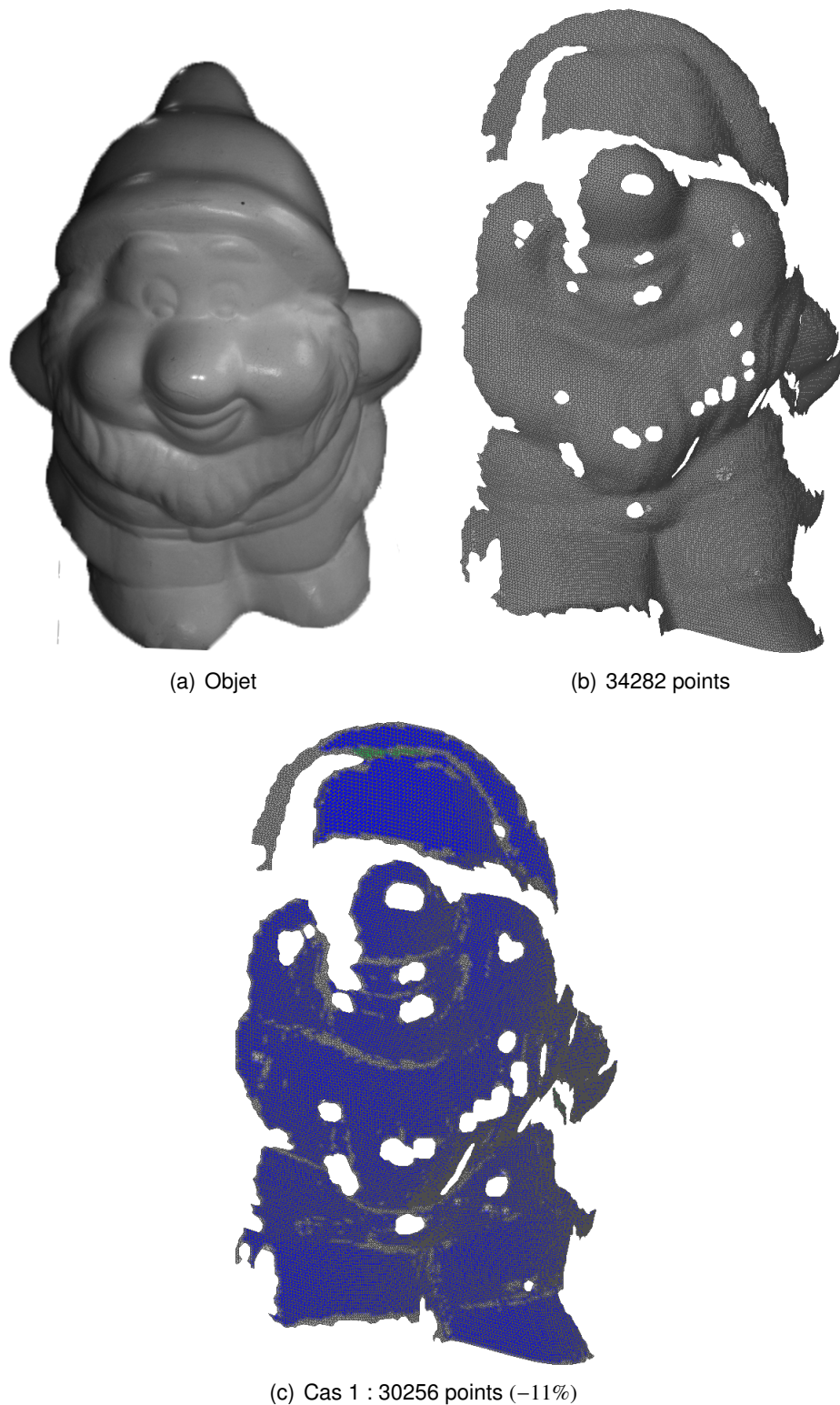


FIGURE 7 – *Limite du système.*

L'objet illustré en figure 7 présente les limites du système. Lorsque l'objet n'est pas composé de primitives, la simplification de points s'avère nettement moins efficace sur ce type d'objet.



# TABLE DES FIGURES

1.1	<i>Diverses applications faisant appel à la numérisation 3D.</i>	1
1.2	<i>Différents types de scanner 3D.</i>	2
1.3	<i>La chaîne 3D classique : de l'acquisition à la compression.</i>	3
2.1	<i>Graphe non exhaustif des systèmes d'acquisition 3D.</i>	8
2.2	<i>Principe de la stéréovision.</i>	10
2.3	<i>Motifs de lumière structurée.</i>	12
2.4	<i>Modélisation d'une caméra : modèle sténopé.</i>	15
2.5	<i>Modélisation d'un système de stéréovision.</i>	18
2.6	<i>Illustration de la géométrie épipolaire. En rouge, les lignes épipolaires et en bleu, le plan épipolaire.</i>	21
2.7	<i>Rectification des images. a) Images non rectifiées. b) Images rectifiées.</i>	22
2.8	<i>Conception d'un bruit de Perlin en deux dimensions. La valeur de chaque pixel est déterminée à partir des gradients de chaque sommet voisin. L'image de droite est le résultat obtenu par cette méthode.</i>	24
2.9	<i>Principe de la triangulation. Connaissant <math>q_0</math>, <math>q_1</math>, <math>\alpha</math> et <math>\beta</math>, il est possible de calculer la position du point <math>Q</math>.</i>	25
2.10	<i>Éléments qui composent un maillage 3D.</i>	29
2.11	<i>Courbure d'un objet 3D. a) Principe de mesure de la courbure. b) Courbure moyenne obtenue sur un objet 3D.</i>	31
2.12	<i>Image gaussienne (b) du nuage de points présenté en (a). Extrait de [67].</i>	34
2.13	<i>Les trois différentes méthodes de décimation.</i>	38
2.14	<i>Simplification de maillage par la méthode présentée par Hoppe [77].</i>	40



2.15	<i>Principe de la transformée de Hough. A chaque point A,B et C correspond une sinusoïde dans l'espace de Hough. Si les sinusoïdes se croisent en un point, il existe alors une droite qui passe par les trois points A,B et C dont l'orientation correspond à l'intersection des sinusoïdes dans l'espace de Hough. . . . .</i>	42
2.16	<i>Propriétés de courbure de différentes primitives simples. Les flèches correspondent aux directions principales. Extrait de [89]. . . . .</i>	45
3.1	<i>La chaine 3D classique : de l'acquisition à la modélisation. . . . .</i>	47
3.2	<i>Scanner de la société Noomeo. . . . .</i>	48
3.3	<i>La géométrie permettant de déterminer l'orientation de la surface à un point de la grille à partir des lignes de cette grille. . . . .</i>	50
3.4	<i>Adaptation de la méthode de Song [56] au scanner utilisé. a) Grille utilisée dans la méthode présentée par Song. b) Grille de germes (en vert) utilisée pour adapter la méthode de Song. . . . .</i>	51
3.5	<i>Résultats obtenus en adaptant la méthode de Song avec le scanner 3D utilisé. . . . .</i>	52
3.6	<i>Etude de l'influence de la densité de points et du nombre de tangentes utilisées. . . . .</i>	53
3.7	<i>Caractérisation du bruit des normales. . . . .</i>	54
3.8	<i>Superposition de l'histogramme et de la loi normale estimée à partir de la variance et la moyenne du bruit (courbe rouge). . . . .</i>	55
3.9	<i>Loi du <math>\chi^2</math> pour 22 degrés de libertés. Avec une erreur <math>\alpha = 5\%</math> autorisée, le <math>\chi^2</math> doit être inférieur à 33.9244 pour que l'hypothèse nulle soit respectée. . .</i>	56
3.10	<i>Schématisation de la problématique. (a) Principe utilisé pour déterminer la courbure. (b) Relation entre le plan d'éclairage (demi-cercle) et le plan de la caméra (ellipse). . . . .</i>	57
3.11	<i>Principe utilisé pour calculer les coefficients de discontinuité. a) La ligne projetée va se déformer en fonction de la forme de l'objet. b) L'orientation des normales à la ligne varie en fonction de la forme 3D de l'objet. . . . .</i>	60

3.12	<i>Calcul des coefficients de discontinuité. a) Objet test. b) Coefficients de discontinuité calculés pour chaque point image (la zone en bleu représente les points n'ayant pas de coefficients de discontinuité). c) Résultats plaqués sur le nuage de points en fausses couleurs pour un meilleur rendu visuel. . . . .</i>	61
3.13	<i>Représentation de la carte de Gauss. Toutes les normales à la surface sont placées au centre d'un disque unitaire. . . . .</i>	62
3.14	<i>Représentation de différentes primitives géométriques sur la sphère gaussienne. a-d) Représentation théorique pour un plan, un cylindre, un cône puis une sphère. e-h) Représentation pratique pour les mêmes primitives avec notre système. La représentation est partielle car l'acquisition ne fournit qu'une portion de la primitive. . . . .</i>	63
3.15	<i>Calcul de l'angle du cône. a) Normale à la surface et angle du cône. b) Représentation sur la sphère gaussienne et distance entre le centre du cercle et de la sphère. . . . .</i>	68
3.16	<i>Résultats sur une pièce test. a) Valeurs propres obtenues sur les différents objets. Les valeurs propres qui correspondent à un plan sont représentées en vert, en bleu pour une sphère et en rouge pour un cylindre. b) Primitives modélisées à partir des vecteurs propres. . . . .</i>	70
3.17	<i>Résultats sur un objet test composé de deux cylindres et d'un méplat. a) Valeurs propres obtenues pour les deux cylindres. b) Primitives modélisées à partir des vecteurs propres. . . . .</i>	71
3.18	<i>Résultats sur un cône. a) Valeurs propres obtenues. b) Primitive modélisée à partir des vecteurs propres. . . . .</i>	71
4.1	<i>Exemple de positionnement des germes (en vert) dans l'image de référence dans le cas où <math>d = \frac{1}{3}</math> et <math>d = \frac{1}{2}</math>. . . . .</i>	74
4.2	<i>Principe de notre système de numérisation dynamique. L'idée est d'acquérir un nuage de points peu dense puis d'extraire différents attributs afin de sélectionner les zones qui ont besoin ou non d'être raffinées. Le système étant itératif, il permet d'adapter la densité de points à la complexité de la pièce à numériser. . . . .</i>	75
4.3	<i>Principe détaillé du système de numérisation dynamique. . . . .</i>	77
4.4	<i>Classement par catégories des différentes méthodes de segmentation d'images. . . . .</i>	78

4.5	<i>Principe du Split and Merge. La première étape (en haut) consiste à diviser l'image en plusieurs parties égales tant que la zone n'est pas homogène. La seconde partie (en bas) consiste à regrouper les graphes obtenus lors de la première étape afin de regrouper les zones qui se ressemblent. . . .</i>	79
4.6	<i>Calcul des coefficients de discontinuité depuis les images du scanner. a) Image de la caméra de référence. b) Coefficients de discontinuité. . . . .</i>	81
4.7	<i>Segmentation des différents objets qui composent la scène à partir des coefficients de discontinuité. . . . .</i>	81
4.8	<i>Partie d'un cylindre avec une variation de la surface localisée sur les bords.</i>	82
4.9	<i>Représentation de la distance entre la surface modélisée et les points 3D. a-b) Distance primitive/points 3D dans le cas d'un cylindre avant et après ajustement de la primitive sur les points. c-d) Distance primitive/points 3D dans le cas d'un plan. . . . .</i>	83
4.10	<i>Distance entre les points 3D et la primitive modélisée sur des objets mal segmentés. a-b) Objets mal segmentés. c-d) Distance entre la surface de la primitive et les points 3D respectivement pour les objets a) et b). . . . .</i>	85
4.11	<i>Représentation tridimensionnelle des courbes et carreaux de Bézier. a) Courbe de Bézier. b) Carreau de Bézier cubique. . . . .</i>	87
4.12	<i>Estimation des points de contrôle d'une courbe de Bézier d'ordre 3. . . . .</i>	89
4.13	<i>Estimation des points de contrôle en 3D. a) Plans tangents à la surface aux points <math>P_0</math> et <math>P_3</math>. b) Plans orthogonaux (<math>P_0, P_3</math>) et passant par les points A et B. c) Les intersections des plans forment deux lignes <math>l_1</math> et <math>l_2</math>. d) Les intersections des lignes avec le plan <math>\Pi_r</math> forment les points de contrôle <math>P_1</math> et <math>P_2</math>. . . . .</i>	90
4.14	<i>Représentation de l'erreur entre un nuage de points dense et des surfaces paramétriques obtenues depuis un nuage de points de densité <math>d = 1/5</math>. a) Surface estimée sur un cylindre avec le logiciel RapidForm. b) Surface de Bézier avec prise en compte des normales sur le même cylindre. c) Surface RapidForm sur des objets non identifiés. d) Surface de Bézier sur ces mêmes objets. . . . .</i>	91
4.15	<i>Grille utilisée pour décomposer les objets en quadrilatères. a) Grille régulière générée dans l'image. b) Déformation de la grille en fonction des données. . . . .</i>	92

4.16	<i>Positionnement des graines servant à la mise en correspondance. a) Grille de germes. b) Zones marquées lors de cette itération. c) Grille de germes modifiée. . . . .</i>	94
5.1	<i>Interface graphique du système développé. . . . .</i>	96
5.2	<i>Objets utilisés pour tester notre système de numérisation. . . . .</i>	97
5.3	<i>Principe du système de numérisation dynamique. . . . .</i>	97
5.4	<i>Première étape de notre système pour l'objet 1. a) Coefficients des discontinuités plaqués sur les points 3D. b) Segmentation des objets : les points gris représentent les points non segmentés. . . . .</i>	98
5.5	<i>Première étape de notre système pour l'objet 2. a) Coefficients des discontinuités plaqués sur les points 3D. b) Segmentation des objets : les points gris représentent les points non segmentés. . . . .</i>	99
5.6	<i>Première étape de notre système pour l'objet 3. a) Coefficients des discontinuités plaqués sur les points 3D. b) Segmentation des objets : les points gris représentent les points non segmentés. . . . .</i>	99
5.7	<i>Représentation sur la sphère gaussienne de deux zones segmentées (objet 1). a) Représentation 3D de deux zones segmentées. b) Représentation gaussienne de la zone verte. c) Représentation gaussienne de la zone orange. . . . .</i>	100
5.8	<i>Représentation sur la sphère gaussienne de deux zones segmentées (objet 2). a) Représentation 3D de deux zones segmentées. b) Représentation gaussienne de la zone verte. c) Représentation gaussienne de la zone orange. . . . .</i>	100
5.9	<i>Représentation sur la sphère gaussienne de deux zones segmentées (objet 3). a) Représentation 3D de deux zones segmentées. b) Représentation gaussienne de la zone verte. c) Représentation gaussienne de la zone orange. . . . .</i>	100
5.10	<i>Test d'appartenance à une primitive géométrique simple : en vert les zones identifiées comme des primitives et en bleu les zones qui ne le sont pas. a) Résultats du test pour l'objet 1 : toutes les zones sont identifiées comme des primitives. b) et c) Respectivement, les résultats des objets 2 et 3 : certaines zones n'étant pas identifiées comme une primitive, une surface de Bézier sera approximée sur celles-ci. . . . .</i>	101

5.11	<i>Distance entre la primitive ajustée et les points 3D pour les deux zones sélectionnées sur l'objet 1. a) Échelle des distances. b) Distance points/primitives pour la première zone (figure 5.7-b). c) Distance pour la deuxième zone (figure 5.7-c).</i>	102
5.12	<i>Représentation sur la sphère gaussienne de deux zones segmentées (objet 3). a) Représentation 3D de deux zones segmentées. b) Représentation gaussienne de la zone verte. c) Représentation gaussienne de la zone orange.</i>	103
5.13	<i>Positionnement des graines servant à la mise en correspondance pour l'objet 1. a) Image capturée par le scanner. b) Masque des zones marquées à la première itération. b) Grille de graine (en rouge) utilisée pour la mise en correspondance lors de l'itération suivante.</i>	103
5.14	<i>Résultats pour les trois objets avec une densité <math>d = 1/4</math>. a), d) et g) Zones segmentées. b), e) et h) Identification des primitives (zones vertes) et des surfaces (zones bleues). c), f) et i) Zones marquées (vertes) et non marquées (rouges).</i>	104
5.15	<i>Résultats finaux obtenus pour chaque objet. a), c) et e) Nuage de points maillé obtenu à la densité maximale <math>d = 1/2</math>. b), d) et f) Nuage de points maillé obtenu avec notre système. En vert on trouve les zones identifiées comme des primitives et en bleu les zones identifiées comme des surfaces paramétriques. Les zones grises correspondent aux points calculés pour <math>d = d_f</math> qui n'ont pas été identifiés par une primitive.</i>	106
5.16	<i>Résultats obtenus sur chaque objet avec les paramètres du premier cas. Distance moyenne <math>\epsilon</math> entre les données obtenues et les données denses. a) <math>\epsilon = 0,01832</math> mm. b) <math>\epsilon = 0,02109</math> mm. c) <math>\epsilon = 0,01908</math> mm.</i>	107
5.17	<i>Résultats obtenus sur chaque objet avec les paramètres du second cas. Distance moyenne <math>\epsilon</math> entre les données obtenues et les données denses. a) <math>\epsilon = 0,01844</math> mm. b) <math>\epsilon = 0,02195</math> mm. c) <math>\epsilon = 0,02506</math> mm.</i>	108
5.18	<i>Evolution du taux de compression en fonction de l'erreur tolérée.</i>	109
5.19	<i>Evolution de l'erreur finale en fonction de l'erreur tolérée.</i>	109
5.20	<i>Résultats obtenus par différentes méthodes de simplification.</i>	110
5.21	<i>Objet 1 : Evolution de l'erreur finale en fonction du taux de compression.</i>	111
5.22	<i>Objet 2 : Evolution de l'erreur finale en fonction du taux de compression.</i>	111
5.23	<i>Objet 3 : Evolution de l'erreur finale en fonction du taux de compression.</i>	112

5.24	<i>Image fournie par l'Optinum sur un objet test.</i>	113
5.25	<i>Principe utilisé pour déterminer les contours d'un trou de perçage.</i>	114
5.26	<i>Principe utilisé pour déterminer les contours d'un trou de perçage.</i>	115
6.1	<i>schéma du système de numérisation dynamique.</i>	120
2	<i>Représentation de la normale par rapport à l'axe de symétrie.</i>	138
3	<i>Résultats sur une pièce manufacturée.</i>	146
4	<i>Résultats sur une pièce manufacturée.</i>	147
5	<i>Résultats sur une pièce manufacturée.</i>	148
6	<i>Résultats sur une pièce manufacturée.</i>	149
7	<i>Limite du système.</i>	150



# LISTE DES TABLES

2.1	Tableau comparatif des différentes techniques optiques de numérisation 3D [30]. . . . .	13
3.1	<i>Résultats obtenus sur trois cylindres différents.</i> . . . .	58
3.2	<i>Résultats obtenus en fonction du nombre de points utilisés pour déterminer la conique <math>A_Q</math>.</i> . . . .	58
3.3	<i>Valeurs propres de la matrice de covariance pour les différentes primitives modélisées.</i> . . . .	65







## Résumé :

Avec l'évolution constante des technologies, les scanners 3D fournissent de plus en plus de données avec une précision toujours plus grande. Cependant, l'augmentation considérable de la taille des données pose des problèmes, les fichiers deviennent très lourds et il peut en découler des difficultés de transmission ou de stockage. C'est pourquoi, la plupart du temps les données obtenues par les scanners vont être analysées, traitées puis simplifiées ; on parle alors de chaîne d'acquisition 3D. Ce manuscrit présente une approche qui permet de numériser les objets de manière dynamique, en adaptant la densité de points dès l'acquisition en fonction de la complexité de l'objet à numériser, et ce sans a priori sur la forme de l'objet. Ce système permet d'éviter de passer par la chaîne 3D classique, en ne calculant pas un nuage de points dense qu'il faudra simplifier par la suite, mais en adaptant la densité de points au niveau de l'acquisition afin d'obtenir des données simplifiées directement à la sortie de l'acquisition, permettant ainsi de réduire considérablement le temps de traitement des données.

**Mots-clés :** Chaîne 3D, Acquisition, Simplification, Extraction de primitives

## Abstract:

With constant evolution of technology, 3D scanners are providing more and more data with ever greater precision. However, the substantial increase of the data size is problematic. Files become very heavy and can cause problems in data transmission or data storage. Therefore, most of the time, the data obtained by the 3d scanners will be analyzed, processed and simplify; this is called 3D acquisition chain. This manuscript presents an approach which digitize objects dynamically by adapting point density during the acquisition depending on the object complexity, without informations on object shape. This system allows to avoid the use of the classic 3D chain. This system do not calculate a dense points cloud which will be simplify later, but it adapts the points density during the acquisition in order to obtain directly simplified data to the acquisition output. This process allows to reduce significantly processing time.

**Keywords:** 3D chain, Acquisition, Simplification, Primitives extraction

