

N° d'ordre:

THÈSE DE L'UNIVERSITÉ DE LYON

Délivrée par

L'UNIVERSITÉ CLAUDE BERNARD LYON 1

ÉCOLE DOCTORALE INFOMATHS

DIPLÔME DE DOCTORAT

Conventions Industrielles du Formation par la REcherche (CIFRE)

Soutenue publiquement le 25 Juin 2012

par **Aurélien Bey**

TITRE :

**Reconstruction de modèles CAO de scènes complexes à partir de nuages de points basée sur l'utilisation de connaissances *a priori***

DIRECTEURS:

Samir AKKOUCHE (LIRIS - CNRS)  
Raphaëlle CHAINE (LIRIS - CNRS)  
Raphaël MARC (EDF R&D)  
Guillaume THIBAUT (EDF R&D - LPPA - CNRS)

JURY:

Rapporteurs: Xavier DESCOMBES  
Jean-Claude LÉON  
Examineurs: Marc DANIEL  
Thomas CHAPERON



## Résumé

Certaines opérations de maintenance sur sites industriels nécessitent une planification à partir de modèles numériques 3D des scènes où se déroulent les interventions. Pour permettre la simulation de ces opérations, les modèles 3D utilisés doivent représenter fidèlement la réalité du terrain. Ces représentations virtuelles sont habituellement construites à partir de nuages de points relevés sur le site, constituant une description métrologique exacte de l'environnement sans toutefois fournir une description géométrique de haut niveau. Il existe une grande quantité de travaux abordant le problème de la reconstruction de modèles 3D à partir de nuages de points, mais peu sont en mesure de fournir des résultats suffisamment fiables dans un contexte industriel et cette tâche nécessite en pratique l'intervention d'opérateurs humains.

Les travaux réalisés dans le cadre de cette thèse visent l'automatisation de la reconstruction, avec comme principal objectif la fiabilité des résultats obtenus à l'issue du processus. Au vu de la complexité de ce problème, nous proposons d'exploiter des connaissances et données a priori pour guider la reconstruction. Le premier a priori concerne la composition des modèles 3D : en Conception Assistée par Ordinateur (CAO), les scènes industrielles sont couramment décrites comme des assemblages de primitives géométriques simples telles que les plans, sphères, cylindres, cônes, tores, etc. Nous hiérarchisons l'analyse en traitant dans un premier temps les plans et les cylindres, comme un préalable à la détection des tores. On obtient ainsi une description fiable des principaux composants d'intérêt dans les environnements industriels. Nous proposons en outre d'exploiter un certain nombre de règles régissant la manière dont ces primitives s'assemblent en un modèle CAO, basées sur des connaissances "métier" caractérisant les scènes industrielles que nous traitons. De plus, nous tirons parti d'un modèle CAO existant décrivant une scène similaire à celle que nous souhaitons reconstruire, provenant typiquement de la reconstruction antérieure d'un site semblable au site d'intérêt. Bien que semblables en théorie, ces scènes peuvent présenter des différences significatives qui s'accroissent au cours de leur exploitation.

La méthode que nous développons se fonde sur une formulation Bayésienne du problème de reconstruction : il s'agit de retrouver le modèle CAO le plus probable vis à vis des différentes attentes portées par les données et les a priori sur le modèle à reconstruire. Les diverses sources d'a priori s'expriment naturellement dans cette formulation. Pour permettre la recherche du modèle CAO optimal, nous proposons une approche basée sur des tentatives d'insertion d'objets générés aléatoirement. L'acceptation ou le rejet de ces objets repose ensuite sur l'amélioration systématique de la solution en cours de construction. Le modèle CAO se construit ainsi progressivement, par ajout et suppression d'objets, jusqu'à obtention d'une solution localement optimale.

**Keywords :** Reconstruction, Modèle CAO, Reconnaissance de formes, Nuage de points, optimisation stochastique



# Reconstruction of CAD model of industrial scenes using *a priori* knowledge

## Abstract

3D models are often used in order to plan the maintenance of industrial environments. When it comes to the simulation of maintenance interventions, these 3D models have to describe accurately the actual state of the scenes they stand for. These representations are usually built from 3D point clouds that are huge set of 3D measurements acquired in industrial sites, which guarantees the accuracy of the resulting 3D model. Although there exists many works addressing the reconstruction problem, there is no solution to our knowledge which can provide results that are reliable enough to be further used in industrial applications. Therefore this task is in fact handled by human experts nowadays.

This thesis aims at providing a solution automating the reconstruction of industrial sites from 3D point clouds and providing highly reliable results. For that purpose, our approach relies on some available *a priori* knowledge and data about the scene to be processed. First, we consider that the 3D models of industrial sites are made of simple primitive shapes. Indeed, in the Computer Aided Design (CAD) field, this kind of scenes are described as assemblies of shapes such as planes, spheres, cylinders, cones, tori, . . . Our own work focuses on planes, cylinders and tori since these three kind of shapes allow the description of most of the main components in industrial environment. Furthermore, we set some *a priori* rules about the way shapes should be assembled in a CAD model standing for an industrial facility, which are based on expert knowledge about these environments. Eventually, we suppose that a CAD model standing for a scene which is similar to the one to be processed is available. This *a priori* CAD model typically comes from the prior reconstruction of a scene which looks like the one we are interested in. Despite the fact that they are similar theoretically, there may be significant differences between the sites since each one has its own life cycle.

Our work first states the reconstruction task as a Bayesian problem in which we have to find the most probable CAD Model with respect to both the point cloud and the *a priori* expectations. In order to reach the CAD model maximizing the target probability, we propose an iterative approach which improves the solution under construction each time a new randomly generated shape is tried to be inserted in it. Thus, the CAD model is built step by step by adding and removing shapes, until the algorithm gets to a local maximum of the target probability.

**Mots-clé:** Reconstruction, CAD Model, Shapes recognition, Point cloud, Stochastic optimization

---

*Cette thèse s'inscrit dans le cadre de Conventions Industrielles de Formation par la Recherche (CIFRE).*

*Elle fut financée conjointement par l'Association Nationale Recherche et Technologie (ANRT) et Électricité De France, Recherche et Développement (EDF R&D) sur une période allant de Décembre 2008 à Mai 2012, et co-encadrée durant ces trois ans et demi par le Laboratoire d'InfoRmatique en Image et Systèmes d'information (LIRIS) et EDF R&D.*

---

# Contents

<b>Introduction</b>	<b>3</b>
<b>1 État de l'art</b>	<b>13</b>
1.1 Modèles 3D	15
1.1.1 Représentation surfacique discrète	15
1.1.2 Représentation volumique discrète	16
1.1.3 Représentations continues	17
1.2 Nuages de points	24
1.2.1 Acquisition	24
1.2.2 Recalage	27
1.2.3 Normales	29
1.2.4 Défauts	31
1.3 Reconnaissance et ajustement de formes	33
1.3.1 Ajustement de surfaces	34
1.3.2 Segmentation	38
1.3.3 Détection de primitives géométriques	46
1.3.4 Appariement de formes similaires	52
1.4 Reconstruction de scènes 3D	53
1.4.1 Surfaces libres	53
1.4.2 Primitives géométriques	55
1.4.3 Hybride	56
1.4.4 Recours à l' <i>a priori</i>	57
1.5 Conclusion	58
1.5.1 Positionnement par rapport à l'existant	58
1.5.2 La reconstruction en pratique	59
<b>2 Formulation du problème</b>	<b>63</b>
2.1 Données et connaissances <i>a priori</i>	65
2.1.1 Nuage de points	65
2.1.2 Modèle 3D CAO	65
2.1.3 Connaissances métier	69
2.2 Formulation probabiliste du problème	70
2.2.1 Formalisme Bayésien	70
2.2.2 Application au cas de la reconstruction	73
2.3 Proposition d'énergies	75
2.3.1 Attache aux données	75

2.3.2	Énergie géométrique	81
2.3.3	Énergie topologique	85
2.3.4	Évaluation des paramètres de pondération	95
<b>3</b>	<b>Optimisation pour la reconstruction</b>	<b>101</b>
3.1	Considérations générales d'optimisation	103
3.2	Méthodes de Monte Carlo sur les Chaînes de Markov	104
3.2.1	Cadre théorique	104
3.2.2	Applications	111
3.2.3	Difficultés	114
3.3	Approche gloutonne	116
3.3.1	Éléments incompatibles	116
3.3.2	Transition optimale localement	117
3.3.3	Calcul efficace des conflits	119
3.4	Traitement des coudes	121
3.4.1	Déterminer les points à la jonction des cylindres	122
3.4.2	Calcul du tore approximant les points	123
3.5	Résultats	125
3.5.1	Expérimentation pour la détection de formes dans des images	125
3.5.2	Expérimentations sur des données 3D réelles	127
<b>4</b>	<b>Génération de primitives</b>	<b>137</b>
4.1	Principe	139
4.1.1	Échantillons aléatoires	139
4.1.2	Recours à l' <i>a priori</i>	140
4.2	Création stochastique de primitives	141
4.2.1	Localiser la zone de création	141
4.2.2	Sélection des points restants	143
4.2.3	Création à partir des points	149
4.2.4	Analyse du processus et amélioration	149
4.3	Borner les primitives	154
4.3.1	Limites grossières	155
4.3.2	Limites effectives	155
4.3.3	Bornes du cylindre	155
4.3.4	Bornes du plan	157
4.3.5	Taille des tranches et cellules	159
4.4	Calcul de la couverture	160
4.5	Résultats	161
4.5.1	Calcul des limites de cylindres	161
4.5.2	Adéquation vis-à-vis des données et de l' <i>a priori</i>	162
	<b>Conclusion</b>	<b>169</b>
<b>A</b>	<b>Primitives géométriques pour la construction CAO</b>	<b>173</b>
A.1	Plans	175
A.1.1	Cas du plan infini	175
A.1.2	Outils géométriques	175
A.2	Cylindres	176
A.2.1	Cas du cylindre infini	177
A.2.2	Outils géométriques	177

A.3	Tores à section circulaire . . . . .	178
A.3.1	Outils géométriques . . . . .	178
A.4	Cônes . . . . .	180
A.5	Sphéroïde . . . . .	180
A.6	Divers . . . . .	181
<b>B</b>	<b>Collisions entre primitives géométriques</b>	<b>183</b>
B.1	Plan - Plan . . . . .	185
B.2	Cylindre - Cylindre . . . . .	186
B.3	Cylindre - Plan . . . . .	188
<b>C</b>	<b>Accès rapide aux points</b>	<b>191</b>
C.1	Arbres de subdivision . . . . .	193
C.1.1	<i>Octree</i> . . . . .	193
C.1.2	<i>KD-tree</i> . . . . .	193
C.2	Requête de plus proche voisin . . . . .	195
C.3	Voisinage d'une primitive . . . . .	195



# List of Figures

1	Simulation 3D d'intervention pour la maintenance . . . . .	5
2	Modèle 3D d'un bâtiment de centrale électrique . . . . .	6
3	Comparaison d'équipements supposés semblables . . . . .	7
1.1	Description de maillages triangulaires . . . . .	16
1.2	Exemple de maillage surfacique . . . . .	17
1.3	Exemple de maillage volumique . . . . .	18
1.4	Surface implicite . . . . .	19
1.5	Représentation paramétrique du volume d'un cylindre . . . . .	20
1.6	Surface de subdivision . . . . .	21
1.7	Constructive Solid Geometry . . . . .	23
1.8	Boundary Representation . . . . .	23
1.9	Principe de la stéréorestitution en photogrammétrie . . . . .	25
1.10	Principe de l'acquisition LIDAR . . . . .	26
1.11	Recalage de nuages de points . . . . .	28
1.12	Normales dans un nuage de points . . . . .	32
1.13	Validation d'ajustement par détection de tendances . . . . .	37
1.14	Moindres carrés pondérés glissants . . . . .	38
1.15	Exemples de segmentation sur images de profondeur . . . . .	39
1.16	Segmentation d'image par croissance de régions . . . . .	40
1.17	Segmentation d'image par division-fusion . . . . .	40
1.18	Segmentation d'image par détection de contours . . . . .	41
1.19	Segmentation d'image par coupe de graphe . . . . .	41
1.20	Courbures principales en un point d'une surface . . . . .	42
1.21	Diagramme de complexité des formes . . . . .	43
1.22	Sphères gaussiennes associées aux primitives usuelles . . . . .	45
1.23	Transformée de Hough d'un nuage de points . . . . .	51
1.24	Résultat du RANSAC sur une scène industrielle . . . . .	59
1.25	Reconstruction manuelle d'un nuage de points . . . . .	60
2.1	Traitement du génie civil par analyse de différences . . . . .	67
2.2	Comparaison d'un modèle CAO <i>a priori</i> et d'un nuage de points . . . . .	69
2.3	Configurations métier . . . . .	71
2.4	Contribution d'attache aux données . . . . .	78
2.5	Distances et angles pour le calcul de la contribution $h_D$ . . . . .	79
2.6	Évolution des scores d'attache aux données en fonction d'une perturbation . . . . .	81
2.7	Comparaison de cylindres . . . . .	85

2.8	Comparaison de plans	86
2.9	Connexion entre deux cylindres	87
2.10	Contribution topologique en fonction de la distance entre axes de cylindres	88
2.11	Contribution topologique en fonction de l'angle entre axes de cylindres	89
2.12	Contribution topologique en fonction de la différence de rayon de cylindres	89
2.13	Contribution topologique d'une paire de cylindres connectés	90
2.14	Approximation du volume d'intersection de cylindres	91
2.15	Connexion entre deux plans	92
2.16	Contribution topologique en fonction de l'angle entre plans	92
2.17	Connexion entre un plan et un cylindre	93
2.18	Contribution topologique en fonction de l'angle entre plan et cylindre	94
2.19	Évaluation du seuil de bruit $\epsilon$ par étude statistique	96
3.1	Exemple de minimisation par recuit simulé	109
3.2	Exemple de conflit entre éléments	116
3.3	Graphe de connexion entre éléments de la configuration	119
3.4	Points pour la jonction torique de deux cylindres	123
3.5	Initialisation des paramètres du tore	124
3.6	Détection de disques dans un image synthétique bruitée	126
3.7	Évolution de l'énergie lors de la détection de disques	127
3.8	Détection de disques dans des images réelles	128
3.9	Reconstruction d'un composant de centrale	129
3.10	Évolution de l'énergie au cours de la reconstruction	130
3.11	Arrêt prématuré de l'optimisation	130
3.12	Évolution de la solution au cours de la reconstruction	131
3.13	Reconstruction à partir d'un modèle partiel et de tolérances permissives	132
3.14	Erreur dans la reconstruction	133
3.15	Reconnaissance d'un modèle CAO <i>a priori</i> partiel	134
3.16	Reconstruction d'une scène complexe	135
3.17	Gestion des tuyauteries	135
3.18	Gestion des nouveaux éléments	136
3.19	Gestion des petits éléments: exemple sur des rambardes	136
4.1	Effet du bruit sur la construction de primitives géométriques	140
4.2	Probabilité $P_S$ de sélection d'un point	142
4.3	Résultat du tirage de graines dans un nuage de points	144
4.4	Densité de probabilité pour le choix du rayon de voisinage	146
4.5	Voisinage de la graine pour la construction de candidats	147
4.6	Première estimation du candidat à partir du voisinage retenu	148
4.7	Calcul des extrémités d'un cylindre candidat	157
4.8	Calcul des extrémités d'un plan candidat	159
4.9	Estimation de la couverture de la surface d'une primitive	161
4.10	Portions de cylindres dans un nuage industriel	161
4.11	Influence des données sur le calcul de portions	162
4.12	Analyse statistique des candidats générés - exemple 1	163
4.13	Visualisation des candidats générés - exemple 1	164
4.14	Analyse statistique des candidats générés - exemple 2	165
4.15	Visualisation des candidats générés - exemple 2	166
4.16	Analyse statistique des candidats générés - exemple 3	166
4.17	Visualisation des candidats générés - exemple 3	167

4.18	Analyse statistique des candidats générés - exemple 4	167
4.19	Visualisation des candidats générés - exemple 4	168
B.1	Détection de collision de plans	186
B.2	Détection de collision de cylindres	188
B.3	Détection de collision plan-cylindre	189
C.1	Principe de la subdivision spatiale hiérarchique	193
C.2	Découpage d'un nuage de points en <i>quadtree</i>	194
C.3	Découpage d'un nuage de points en <i>KD-tree</i>	194
C.4	Recherche de plus proche voisin dans un <i>KD-tree</i>	195



# Liste des Algorithmes

1	Algorithme RANSAC pour la détection de formes. . . . .	47
2	Détection de formes par transformées généralisées de Hough. . . . .	50
3	Algorithme de Metropolis-Hastings. . . . .	107
4	Birth and Death Markov Chain Monte Carlo pour la détection de disques . .	113
5	Algorithme glouton de construction d'une configuration optimale. . . . .	118
6	Algorithme heuristique approximant l'ensemble optimal des conflits. . . . .	121
7	Tirage par acceptation-rejet basé sur l'exploration uniforme des données. . .	143
8	Générateur de graine biaisé. . . . .	152



Quelques notations utilisées par la suite, pour lever toute éventuelle ambiguïté :

$\operatorname{argmax}_x f(x)$	:	Argument du maximum de la fonction " $f$ "
$\operatorname{argmin}_x f(x)$	:	Argument du minimum de la fonction " $f$ "
$\mathbb{X} \subset \mathbb{Y}$	:	Inclusion ensembliste : " $\mathbb{X}$ inclus dans $\mathbb{Y}$ ".
$\mathbb{X} \setminus \mathbb{Y}$ ou $\mathbb{X} \backslash \mathbb{Y}$	:	Soustraction ensembliste : " $\mathbb{X}$ privé de $\mathbb{Y}$ ". On suppose $\mathbb{Y} \subset \mathbb{X}$
$ \mathbb{X} $	:	Cardinal de l'ensemble " $\mathbb{X}$ ".
$\mathbb{S}^n$	:	Sphère unitaire centrée dans $\mathbb{R}^{n+1}$ .
$ M $	:	Déterminant de la matrice " $M$ "
$P(X Y)$	:	Probabilité conditionnelle de "l'évènement $X$ sachant l'évènement $Y$ "
$\nabla f$	:	Gradient de la fonction " $f$ "
$\mathbf{u}$	:	Vecteur " $u$ ". Les vecteurs apparaissent en gras et en minuscule.
$\mathbf{u} \cdot \mathbf{v}$	:	Produit scalaire entre les vecteurs " $u$ " et " $v$ ".
$\mathbf{u} \times \mathbf{v}$	:	Produit vectoriel entre les vecteurs " $u$ " et " $v$ ".



# Introduction

## Contexte industriel: maintenance des outils de production

La maintenance d'un site industriel consiste en la réparation, le changement, le démantèlement ou plus simplement le contrôle de composants de manière à permettre le bon déroulement de la tâche à laquelle le site en question est dédié. Certaines opérations de maintenance de sites de production industriels tels que les centrales électriques, les installations pharmaceutiques ou les complexes de traitement pétrochimique par exemple, peuvent comporter des risques ou des difficultés majeures. Ce constat s'applique particulièrement au cas des centrales nucléaires de production électrique, où la durée et le bon déroulement des opérations constituent un enjeu critique. Les raisons pour lesquelles il est souhaitable que tout se passe pour le mieux sont nombreuses et relativement évidentes.

D'une part, les incidents survenant lors d'interventions en centrale nucléaire peuvent avoir un impact significatif sur le cycle de production, et il est évidemment souhaitable que la maintenance ne soit pas elle-même une source de nouveau dysfonctionnement. Citons par exemple l'extraction de gros composants d'un bâtiment quelconque de la centrale. Il s'agit de faire en sorte qu'aucun équipement ne soit endommagé lors de la dépose ou de l'acheminement de l'élément ciblé. Il est en particulier indispensable de s'assurer de la faisabilité de l'intervention. Ceci n'est pas forcément une évidence pour les plus gros composants, et il peut être nécessaire d'envisager des travaux visant à modifier la structure des installations pour permettre le bon déroulement des opérations, tout en s'assurant qu'elles demeurent opérationnelles.

D'autre part, certaines des interventions de maintenance nécessitent l'interruption du cycle de production, ce qui est typiquement le cas lorsque les outils concernés constituent des "organes vitaux" du site sans lesquels la production ne peut être assurée. Dans le cas des centrales nucléaires, on parle alors d'arrêt de tranche<sup>1</sup>. Pour des raisons économiques et pour garantir l'approvisionnement en électricité<sup>2</sup> dans la mesure des besoins du consommateur, il est souhaitable de faire en sorte que l'arrêt ne dure pas, et que le site retrouve sa

---

<sup>1</sup>Une tranche désigne l'ensemble des installations définissant une unité de production électrique. Pour les centrales nucléaires de seconde génération telles que celles en activité actuellement en France, une tranche se compose schématiquement d'un bâtiment réacteur permettant la pressurisation de l'eau et d'une salle des machines contenant notamment l'alternateur produisant l'électricité à partir de l'eau pressurisée. Chaque tranche contient de plus un certain nombre de bâtiments annexes parmi lesquels figurent la salle de commandes ainsi qu'un bâtiment de stockage du combustible servant notamment pendant les arrêts de tranche.

<sup>2</sup>Les 58 tranches du parc nucléaire français assurent environ 80% de la production de l'électricité fournie par EDF au niveau national en 2010, d'après l'exploitant. Toutes ne possèdent pas le même rendement, mais l'arrêt d'une tranche impacte indéniablement la capacité à conserver le niveau de production d'électricité.

pleine capacité de production au plus vite. Tout doit être fait pour qu'aucun imprévu ne vienne perturber le calendrier des opérations, et le respect des délais est un enjeu crucial pour l'exploitant du site de production.

En outre, le temps est un paramètre clé dans un type particulier d'opérations: la maintenance en environnement irradié. Les opérateurs humains ne peuvent consacrer qu'un temps limité aux interventions sous ces conditions, notamment selon le niveau d'irradiation du site où les opérations doivent avoir lieu. Pour des raisons sanitaires aujourd'hui reconnues, il est donc indispensable de réaliser la maintenance dans les meilleurs délais.

Pour qu'elles se déroulent parfaitement, les opérations de maintenance non triviales doivent être planifiées. Les difficultés doivent être identifiées au préalable pour éviter les incidents, sources notamment de contretemps en plus des conséquences directes qu'ils peuvent occasionner sur le bon fonctionnement du site. Un calendrier doit être établi, et le site doit être préparé pour permettre l'intervention.

Outre la planification des opérations à court terme, l'exploitant doit s'assurer de la capacité de son personnel à réaliser ces opérations. Ceci passe par la formation des intervenants, autre enjeu majeur lié à la maintenance des sites.

De nos jours, l'informatique permet de proposer les outils d'ingénierie permettant la planification des opérations d'une part, et la formation aux techniques d'intervention et de contrôle d'autre part. La division de recherche et développement d'EDF consacre une partie de ses travaux à ces problématiques, en participant notamment au développement de solutions pour :

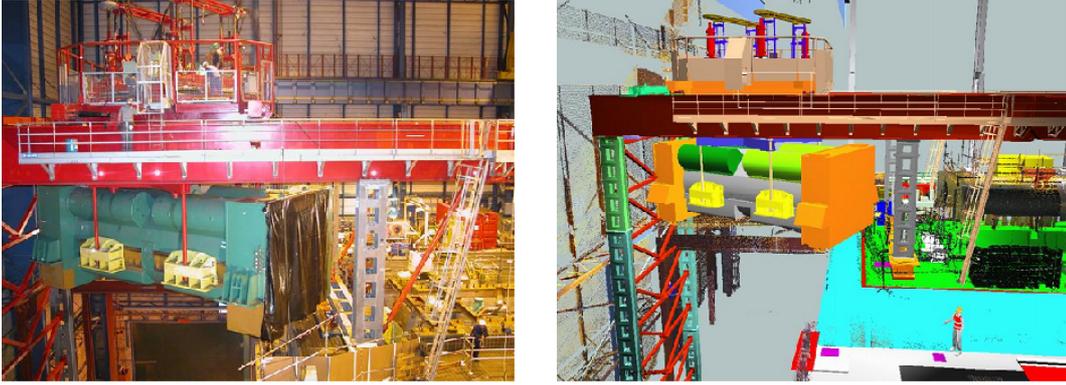
**la simulation d'interventions** via le logiciel ADRM entre autres. Cet outil permet la simulation du déplacement de composants dans un milieu donné, grâce à un système de calcul des collisions entre les objets manipulés et leur environnement. Il est aujourd'hui utilisé par les unités opérationnelles d'EDF pour permettre l'étude de la faisabilité de certaines interventions.

**la planification** via la plateforme SAAM entre autres. Cette solution logicielle propose par exemple des outils pour le colisage, c'est-à-dire la gestion du stockage des composants (dans l'espace et dans le temps) sur site au cours de l'arrêt de tranche. Elle permet de plus d'accéder à une vue statique 3D prévisionnelle de l'état du site lors des différentes étapes de la maintenance. Le logiciel permet en outre la traçabilité des divers composants du site.

**la formation** via divers projets menés en interne. Il s'agit par exemple de permettre l'étude de certaines pièces spécifiques aux environnements étudiés, ou encore l'apprentissage de pratiques et gestes "métier" pour la maintenance.

## Reconstruction de modèles 3D

Les diverses thématiques traitant de la préparation de la maintenance reposent sur un besoin commun: la nécessité de disposer de représentations informatiques tridimensionnelles des sites. On parle de modèle 3D pour désigner ces représentations. Pour certaines de ces applications, en particulier pour la simulation, ces modèles 3D doivent correspondre précisément (de l'ordre de quelques centimètres) à l'état réel des scènes qu'ils représentent. On parle alors de modèle "Tel Que Construit" (abrégé TQC) pour désigner ces



**Figure 1:** Simulation 3D d'intervention pour la maintenance d'une centrale électrique (droite). Ce procédé virtuel permet l'étude de la faisabilité et l'anticipation des risques de certaines opérations conséquentes sur site industriel.

représentations effectives, par opposition aux représentations théoriques des sites issues par exemple de plans. La vérité terrain est en effet souvent différente de l'état prévu à la construction d'un site, et il peut exister des différences significatives du fait de contraintes rencontrées lors de la construction, ou de l'évolution du site au cours de son exploitation et des diverses interventions qu'il aura éventuellement subies.

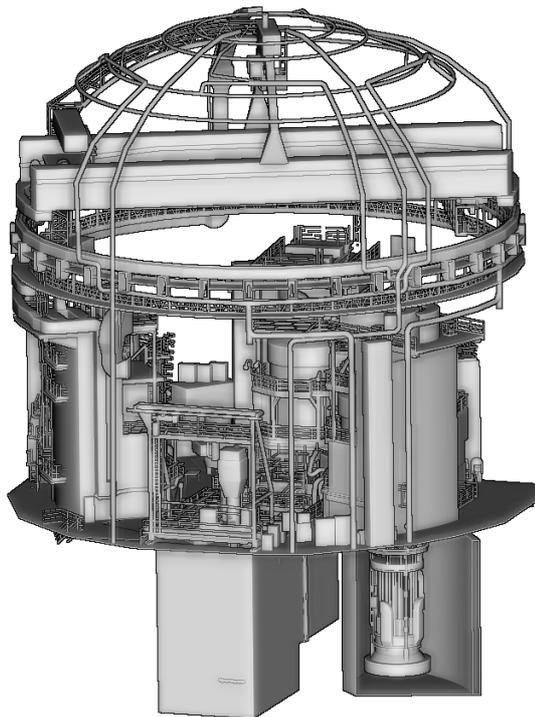
Finalement, il n'existe donc pas de modèle 3D générique permettant de représenter l'ensemble des centrales du parc de production électrique au sens du TQC, puisque chaque tranche est unique. Les centrales appartenant à un même palier<sup>3</sup> pourraient *a priori* présenter une structure identique, mais les raisons évoquées précédemment rendent cette supposition fautive en pratique. À plus forte raison, les centrales issues de paliers distincts sont significativement différentes. Bien qu'il existe des similarités entre centrales d'un même palier, les différences qu'acquière chaque tranche au cours de sa construction et de son cycle de vie (cf. figure 3) rendent nécessaire la mise à disposition d'un modèle 3D spécifique à chaque tranche.

Pour obtenir un modèle 3D fidèle à la réalité, la seule solution envisageable consiste en l'utilisation de mesures relevées sur le terrain. On parle alors de reconstruction TQC de modèles 3D.

Or les scènes industrielles auxquelles nous nous intéressons ici sont extrêmement complexes, et la reconstruction d'un modèle 3D nécessite une quantité considérable de mesures. La technologie LIDAR (Light Detection And Ranging), basée sur l'émission/réception d'un signal laser, permet de nos jours l'acquisition très rapide de volumes immenses de données tridimensionnelles représentant des points à la surface des objets de la scène relevée. On parle de nuages de points pour désigner ces collections de mesures 3D. En recalculant *a posteriori* les surfaces à partir desquelles les points ont été acquis, on peut ainsi reconstruire le modèle 3D TQC du site étudié.

Comme nous le verrons par la suite (cf. chapitre 1), il existe une grande quantité de travaux de recherche abordant ce problème de reconstruction TQC à partir de nuages de

<sup>3</sup>Les centrales nucléaires actuellement en service en France reposent sur le principe de réacteurs à eau pressurisée (REP) et appartiennent toutes à la seconde génération de réacteurs (la classification en générations définit la technologie sur laquelle se base le réacteur et compte aujourd'hui quatre classes retraçant l'évolution des techniques). Elles se divisent toutefois en paliers distincts définissant leur architecture ainsi que leur capacité de production électrique: P4, P'4, N4, CP0, CP1, CP2.



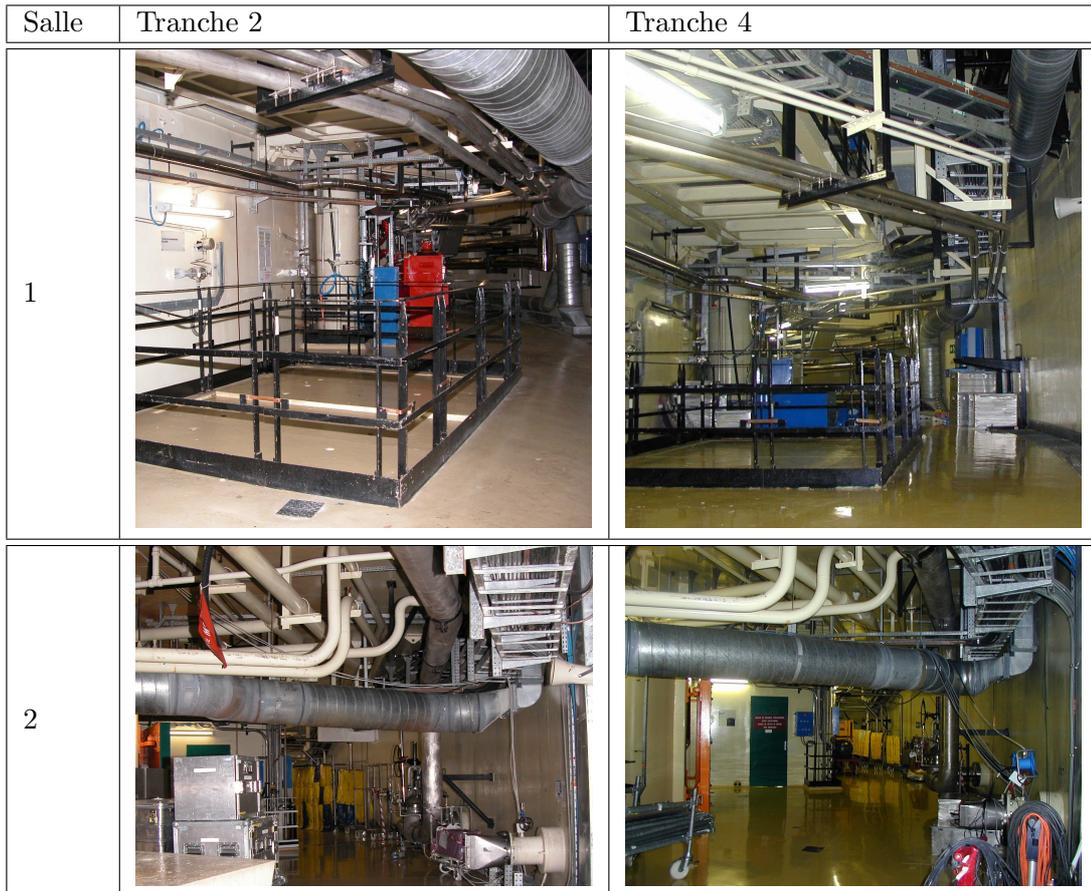
**Figure 2:** *Modèle 3D d'un bâtiment de centrale électrique. On peut constater, malgré la simplification de cette représentation par rapport à la réalité terrain, que les scènes industrielles présentent une grande complexité de par la quantité, la nature et la structuration des éléments qui la composent.*

points. Toutefois, il n'existe pas, à notre connaissance, de solution suffisamment fiable pour permettre l'automatisation de ce processus. Divers outils reconnaissent, avec plus ou moins de succès, des surfaces parmi les points du nuage. Mais beaucoup ne sont pas en mesure de traiter la totalité des données issues d'acquisition en situation réelle. Étrangement, la reconnaissance des formes parmi de grands volumes de données est aussi complexe pour l'ordinateur qu'elle est simple et intuitive pour l'œil (ou plutôt le cerveau) humain, qui sait quant à lui isoler et reconnaître immédiatement les objets représentés par les données grâce à sa capacité d'interprétation multi-échelle de l'information.

De sorte qu'à l'heure actuelle, la reconstruction de modèles 3D est gérée par des opérateurs humains dont l'activité principale consiste en la reconnaissance de surfaces parmi des nuages de points. Cette solution est aujourd'hui la plus fiable qui existe, et la seule qui semble applicable dans le contexte industriel. Elle mobilise toutefois des ressources humaines non négligeables, puisque la reconstruction d'un modèle 3D de centrale peut nécessiter des mois de travail de la part d'une équipe d'experts. Qui plus est, cette tâche est relativement pénible puisque la visualisation et la manipulation de gros nuages de points requiert malgré tout une concentration considérable, et n'est pas toujours intuitive.

## **Le rôle de l'*a priori***

Devant la complexité indéniable de ce problème, il apparaît judicieux de tirer parti de la situation particulière que nous offre le contexte dans lequel se placent nos travaux. Nous nous proposons en effet de tenir compte de deux principales sources d'*a priori* inhérentes au traitement des scènes industrielles, pour permettre la reconstruction fiable de modèles 3D.



**Figure 3:** Comparaison de mêmes équipements dans deux tranches jumelles. On constate que, bien que ces scènes soient très semblables, il existe quelques différences entre certains de leurs composants (tuyauteries au plafond).

**Les connaissances métier** quant à la composition des scènes industrielles établissent un certain nombre d'attentes quant à la géométrie ainsi que la structuration des modèles 3D représentant les scènes traitées.

Ainsi estime-t-on tout d'abord qu'un modèle 3D de ce type de scène se définit comme un assemblage de primitives géométriques simples telles que les plans, cylindres, tores, cônes ou ellipsoïdes permettant de décrire la plupart des composants métier que l'on peut trouver sur un site industriel. Cette représentation est aujourd'hui adoptée dans la plupart des approches de **Conception Assistée par Ordinateur (CAO)** pour la modélisation en 3D d'environnements industriels. Parmi ces primitives, la connaissance des équipements métier qu'elles décrivent nous conduit à nous intéresser plus particulièrement aux plans, cylindres et tores représentant principalement le bâti et les lignes de tuyauteries, qui constituent des éléments majeurs de par leur présence et leur rôle sur les sites industriels. En outre, le rôle secondaire des tores en tant que coudes dans les lignes de tuyauterie nous conduit à hiérarchiser le traitement de la reconstruction de modèles CAO à partir de nuages de points. On se propose ainsi de ne s'intéresser à la reconstruction des tores (coudes) qu'une fois les cylindres (tuyaux) convenablement reconstruits. Ces premiers *a priori* forts quant au contenu des scènes industrielles nous permettent donc de spécifier la géométrie des modèles CAO, ainsi qu'une ébauche de traitement du problème de reconstruction.

Parallèlement, on se donne un ensemble de règles métier portant sur l'assemblage des primitives géométriques reconnues dans le nuage de points, afin de garantir la cohérence du modèle CAO reconstruit. Tout agrégat de primitives géométriques ne définit pas un modèle CAO, encore faut-il que le résultat soit "consistant". Cette consistance s'évalue en terme de conformité des interactions entre les primitives du modèle CAO vis-à-vis d'un certain nombre d'attentes plus ou moins évidentes. Par exemple, on s'attend *a priori* à ce que la jonction de deux cylindres consécutifs d'une même ligne de tuyauterie satisfasse certaines contraintes élémentaires, de telle manière que la circulation des fluides dans le tuyau représenté soit possible. Ces règles d'assemblage constituent un *a priori* portant sur la structuration de la scène, et doivent être prises en compte pour garantir la fiabilité de la reconstruction des modèles CAO.

**Un corpus de représentations existantes** d'environnements industriels peut être utilisé. Ces représentations peuvent prendre plusieurs formes : il peut s'agir de plans architecturaux, de photographies ou encore de modèles CAO (TQC ou bien issus de cotes prises sur plan) dont dispose d'ores et déjà l'exploitant du site en question. Au sein de ce corpus, certaines données peuvent décrire des environnements semblables à la scène que l'on cherche à traiter. En effet, comme nous le mentionnions précédemment, bien que chaque tranche soit unique, il existe des similarités entre certaines centrales, du fait par exemple qu'elles sont issues d'un même palier. Typiquement, un site de production nucléaire français comporte de 2 à 6 tranches numérotées, et les tranches de même parité (2 et 4 par exemple) sont en théorie jumelles, puisqu'elles sont issues des mêmes plans. Insistons sur le fait que cette gémellité théorique ne signifie en rien que les centrales sont rigoureusement identiques, puisqu'après plus de 30 ans d'existence, chacune aura connu son développement propre et aura vieilli différemment de ses pairs. Outre la distinction des différents sites représentés dans le corpus, on pourrait aussi s'appuyer sur une représentation antérieure d'une même centrale. Ici encore, il s'agit de tenir compte des dissimilarités induites par l'évolution du site depuis la dernière mise à jour de sa représentation TQC. Quoi qu'il en soit, les données retenues au sein de ce corpus disponible doivent être pertinentes vis-à-vis de la scène à traiter, et concerner un site industriel *a priori* semblable à celui que l'on cherche à reconstruire.

Nous avons choisi, dans cette thèse, de nous concentrer plus particulièrement sur l'utilisation de modèles CAO existants. Ces modèles CAO "*a priori*" ont un double intérêt. Ils peuvent, d'une part, guider le processus de reconstruction en apportant des informations nécessaires au traitement du nuage de points, qui pourraient difficilement être devinées sans cet apport (par exemple, quant à la taille des objets à reconstruire). D'autre part, un modèle CAO *a priori* permet de légitimer et de valider le modèle CAO reconstruit. En effet, puisque l'*a priori* et le reconstruit représentent deux scènes semblables, par hypothèse, le modèle CAO reconstruit doit alors être lui-même semblable au modèle CAO *a priori* choisi. Cette évaluation de la similarité doit laisser la place à certaines différences pouvant naturellement exister entre les deux scènes. Si la solution calculée est effectivement semblable au modèle CAO *a priori*, aux différences tolérées près, sa légitimité est ainsi renforcée. À l'inverse, on pourrait émettre des doutes sur les parties du modèle CAO reconstruit qui ne seraient pas validées par une correspondance avec le modèle CAO *a priori* retenu.

En résumé, la problématique à laquelle nous nous intéressons dans cette thèse est la suivante:

*Étant donné*

- un nuage de points issu d'une acquisition sur un site industriel
- un ensemble de règles métier d'assemblage des primitives géométriques
- un modèle CAO a priori issu du corpus de données dont dispose l'exploitant du site, tel que la scène représentée par ce modèle CAO soit théoriquement semblable à la scène échantillonnée par le nuage de points
- une spécification de l'amplitude des différences pouvant être attendues entre les scènes représentées par le nuage de points et le modèle CAO a priori

Il s'agit de reconstruire par le calcul un modèle CAO à base de primitives géométriques simples qui:

- corresponde au nuage de points
- satisfasse les contraintes d'assemblage des primitives
- soit semblable au modèle CAO a priori, aux différences tolérées près

Nos travaux s'articulent essentiellement autour de la question de la qualité des modèles CAO reconstruits. L'efficacité en terme de temps de calcul, bien que souhaitable, n'est pas un but dans le cadre de cette thèse. En effet, nous cherchons à soulager l'opérateur humain d'une tâche pénible en l'automatisant dans la mesure du possible. Si l'on suppose qu'il existe un algorithme capable de traiter convenablement ce problème, en garantissant l'obtention de résultats comparables avec ce que fournirait l'expert humain réalisant la même tâche, alors le temps que passerait l'algorithme à résoudre le problème n'aurait pas vraiment d'importance. En effet, le temps de calcul d'une machine est considérablement moins coûteux que le temps de travail d'un humain, et l'ordinateur est capable de travailler sans interruption et à moindre frais. Sous les conditions précédentes, quel que soit le temps qu'aura mis l'ordinateur à trouver son résultat, il sera toujours plus intéressant d'un point de vue économique que le résultat fourni par l'opérateur humain (dans la mesure tout de même où l'ordinateur est plus rapide que l'humain, évidemment). En revanche, les résultats devant être vérifiés par l'humain induisent un nouveau coût, et il est donc souhaitable de faire en sorte que les vérifications ne soient plus une étape indispensable. En conclusion, il est préférable de laisser à l'ordinateur le temps de bien faire les choses, plutôt que de les précipiter.

## Vue d'ensemble et contributions

Ce document introduit dans un premier temps un descriptif des technologies et travaux existants en lien avec la problématique de la reconstruction TQC de modèles CAO (chapitre 1). Cet état de l'art propose tout d'abord un survol des différentes représentations pouvant être adoptées pour décrire les modèles 3D (section 1.1), donnant ainsi un aperçu de l'immense étendue de ce champ disciplinaire. Il s'intéresse ensuite à la question de l'acquisition de gros volumes de mesures 3D en conditions réelles (section 1.2), cruciale pour comprendre et cerner les difficultés inhérentes à l'ensemble des problèmes de traitement de nuages de points. Par la suite, quelques uns des très nombreux travaux existants autour de la thématique de la reconstruction à partir de nuages de points sont présentés (sections 1.3 et 1.4). Parmi ces approches, nous portons particulièrement notre attention sur un certain nombre de travaux intégrant des connaissances *a priori* au problème de la reconstruction

(sections 1.4 et 1.5), parmi lesquels figurent les méthodes probabilistes.

Nos propres travaux s'articulent autour de la formulation probabiliste du problème de reconstruction étant données des connaissances et données *a priori* quant à la scène traitée (chapitre 2). Après avoir détaillé le rôle et la composition des *a priori* industriels (section 2.1), nous proposons en effet de formuler la reconstruction comme correspondant à la recherche du modèle CAO le plus probable par rapport aux contraintes que doit satisfaire la solution calculée (section 2.2).

Notre première contribution consiste donc en la mise en place d'une probabilité permettant d'évaluer la pertinence d'un modèle CAO donné. Cette probabilité nous permet de modéliser formellement le problème de reconstruction, en quantifiant la conformité de tout modèle CAO vis-à-vis des trois contraintes cibles. Elle intègre donc à la fois une évaluation de l'attache aux données du nuage de points, mais aussi une évaluation de la ressemblance au modèle CAO *a priori* et de la qualité de l'assemblage des primitives (section 2.3). Plus un modèle est conforme aux attentes associées à ces trois critères, plus sa probabilité est élevée.

Puisque la résolution du problème de reconstruction passe par la recherche du modèle CAO le plus probable, il nous faut alors proposer une méthode permettant de calculer un tel modèle (chapitre 3). Il existe un certain nombre d'approches permettant d'aborder ce problème d'optimisation, parmi lesquelles les approches stochastiques à base de recuit simulé semblent appropriées (sections 3.1 et 3.2). Toutefois, bien qu'elles reposent sur un socle théorique fort, ces approches présentent notamment l'inconvénient non négligeable d'être très difficiles à paramétrer.

Notre seconde contribution consiste donc en la mise en place d'un algorithme stochastique simple gérant l'assemblage de primitives géométriques en s'inspirant des approches existantes (section 3.3). Cette approche se base sur la construction itérative de la solution, où chaque étape correspond à une tentative d'insertion dans la solution d'une primitive géométrique gérée aléatoirement. Lors de chaque tentative, un ensemble de calculs déterministes permet de décider (en fonction de l'évolution de la probabilité de la solution) de l'acceptation ou du refus de l'insertion du nouvel objet proposé. En cas d'acceptation, l'insertion peut alors s'accompagner de la suppression d'objets qui figuraient déjà dans la solution en cours de construction, mais dont l'incompatibilité avec le nouvel élément ne permettait pas leur conservation au sein du modèle CAO. Chaque nouvelle itération de cet algorithme garantit l'augmentation de la probabilité de la solution reconstruite en cas d'acceptation, ou la stagnation en cas de refus. Finalement, plus l'algorithme avance, plus le modèle CAO s'améliore au sens de la probabilité que nous avons définie. Malgré le fait qu'elle ne possède pas les garanties d'obtention du modèle CAO qui soit effectivement le plus probable, l'heuristique que nous proposons permet toutefois d'obtenir des résultats satisfaisants (section 3.5), et ne requiert quasiment aucun paramétrage contrairement aux recuits simulés.

Une fois le modèle CAO à base de cylindres et de plans reconstruit via cette heuristique d'optimisation, nous proposons une méthode visant à reconnaître les tores joignant des cylindres du modèle, de manière à identifier les coudes dans les lignes de tuyauterie (section 3.4). La reconnaissance des tores dans les nuages de points n'est pas un problème simple (la reconnaissance de plans et cylindres, bien que loin d'être infaillible, est mieux maîtrisée dans la littérature). Le recours aux cylindres fiables d'ores et déjà reconnus permet en revanche de simplifier grandement le problème, et d'obtenir des résultats de bonne qualité.

Notre dernière contribution s'axe autour de la génération aléatoire de plans et cylindres requise par l'algorithme d'optimisation que nous proposons (chapitre 4). La génération purement aléatoire de formes n'est en effet pas envisageable, puisque la reconstruction n'aurait alors quasiment aucune chance de progresser. Pour accélérer la convergence de la résolution, il nous faut "resserrer" la probabilité d'apparition des formes générées autour des primitives les plus aptes à augmenter la pertinence de la solution (section 4.1). Nous proposons ainsi un ensemble d'outils permettant de biaiser le générateur de manière à favoriser la création des primitives les plus pertinentes, en nous appuyant simultanément sur les points du nuage et sur le modèle CAO *a priori* (sections 4.2). On constate alors que le biais introduit par cette approche permet effectivement, en pratique, de voir apparaître fréquemment les objets aptes à améliorer la solution (section 4.5).

L'approche stochastique que nous proposons se distingue des approches habituelles (déterministes) de reconstruction dans le sens où elle n'interdit l'exploration d'aucune solution. En effet, toute primitive de probabilité non nulle peut figurer dans le modèle CAO reconstruit. Toutefois, nous faisons en sorte que les chances de visiter une configuration peu pertinente soient minimales. De plus, aucun choix n'est validé avant la fin du processus: chaque forme composant le modèle CAO reconstruit peut être remise en cause à tout moment. Cette démarche permet l'obtention de résultats de qualité, en ne validant pas en cours de calcul des choix dont on ne peut dire s'ils sont optimaux, contrairement aux approches telles que l'algorithme RANSAC par exemple.

L'approche en elle-même n'est pas nouvelle, puisque les algorithmes basés sur le recuit simulé fonctionnent sur un principe très proche. Mais l'application de ce type de démarche au domaine de la reconstruction TQC est très rare [LKBH10], et nos travaux se distinguent de l'existant dans la mesure où ils intègrent le recours à un modèle CAO disponible supposé semblable à la solution recherchée.



# Chapter 1

## État de l'art

### Sommaire

---

<b>1.1 Modèles 3D</b>	<b>15</b>
1.1.1 Représentation surfacique discrète	15
1.1.2 Représentation volumique discrète	16
1.1.3 Représentations continues	17
<b>1.2 Nuages de points</b>	<b>24</b>
1.2.1 Acquisition	24
1.2.2 Recalage	27
1.2.3 Normales	29
1.2.4 Défauts	31
<b>1.3 Reconnaissance et ajustement de formes</b>	<b>33</b>
1.3.1 Ajustement de surfaces	34
1.3.2 Segmentation	38
1.3.3 Détection de primitives géométriques	46
1.3.4 Appariement de formes similaires	52
<b>1.4 Reconstruction de scènes 3D</b>	<b>53</b>
1.4.1 Surfaces libres	53
1.4.2 Primitives géométriques	55
1.4.3 Hybride	56
1.4.4 Recours à l' <i>a priori</i>	57
<b>1.5 Conclusion</b>	<b>58</b>
1.5.1 Positionnement par rapport à l'existant	58
1.5.2 La reconstruction en pratique	59

---

---

### Préambule

---

*Le traitement de nuages de points et la construction de modèles 3D sont au cœur de multiples travaux depuis une quinzaine d'années. En effet, l'acquisition de données laser revêt une importance croissante dans une très grande variété de disciplines (industrie, géologie, aérospatiale, héritage culturel, architecture, géographie, ...). Le nombre d'équipes de recherche ou d'entreprises s'équipant d'appareils d'acquisition 3D est en augmentation, du fait des avancées technologiques importantes et de l'accessibilité grandissante de ces équipements. L'intérêt que suscitent les méthodes de traitement de données 3D croît avec le succès des appareils d'acquisition, et le nombre d'événements scientifiques intégrant ces problématiques est aujourd'hui considérable.*

*Parallèlement, la demande en modèles 3D devance largement le succès incontestable des technologies de relevé des données 3D. Les modèles 3D concernent en effet un public bien plus large que les professionnels de l'acquisition. Leur utilisation est intensive dans l'industrie du jeu vidéo, où il sont souvent générés procéduralement. Les outils logiciels permettant à tout un chacun de créer son propre modèle 3D sont aujourd'hui très accessibles (certains sont même gratuits), et les modèles 3D se retrouvent tant dans la documentation de l'existant, permettant la représentation et le stockage numérique d'environnements ou d'objets réels, que dans le domaine de l'architecture où l'utilisateur peut créer et visualiser sa maison ou son mobilier virtuels.*

*Une partie seulement de la création de modèles 3D concerne le Tel Que Construit visant à documenter le plus fidèlement possible les environnements existants. C'est dans ce domaine que se rejoignent données 3D acquises et modèles 3D.*

---

## 1.1 Modèles 3D

La dénomination de "modèle 3D" peut revêtir un sens très différent selon la situation dans laquelle elle est utilisée. On ne peut pas réellement donner de définition à la fois formelle et précise de ce qu'est un modèle 3D, tant ce concept recouvre de significations en pratique. Le plus souvent, on parle de modèle 3D pour désigner la représentation d'une surface (habituellement, une 2-variété<sup>1</sup> plongée dans  $\mathbb{R}^3$ ). Ce type de description est majoritairement utilisé dans la communauté de la visualisation, où seule importe l'apparence externe des objets. Toutefois, un modèle 3D peut être une représentation volumique, proposant une notion d'occupation de l'espace: les solides sont ainsi dotés d'un intérieur supposé plein, potentiellement évidé lorsque c'est explicitement spécifié. Ce type de représentation s'utilise généralement dans des applications de simulation physique, nécessitant une description complète de l'espace effectivement occupé par les solides considérés, rendant la notion d'intérieur et d'extérieur indispensable.

Outre la distinction dans ce que peut représenter un modèle 3D, il existe diverses descriptions de modèles 3D. Certaines applications nécessitent en effet de discrétiser l'espace représenté par le modèle 3D, pour en permettre la visualisation ou l'utilisation dans le cadre de calculs complexes. À l'inverse il est possible d'adopter une description continue du modèle 3D à des fins de précision ou de concision, puisque les descriptions continues sont souvent plus compactes que les représentations discrètes.

### 1.1.1 Représentation surfacique discrète

La description très largement dominante de nos jours pour manipuler et stocker les surfaces est le maillage triangulaire (éventuellement quadrangulaire). Cette prépondérance trouve diverses explications, dont la principale est certainement liée à l'histoire des principaux moteurs d'affichage 3D (OpenGL, DirectX) qui, pour être efficaces, ne sont capables de traiter que des triangles. Très longtemps, il a donc fallu fournir des ensembles de triangles aux ordinateurs pour permettre la visualisation des surfaces, engendrant ainsi tout naturellement l'émergence des maillages triangulaires comme description principale des modèles 3D surfaciques.

Concrètement, un maillage triangulaire est un ensemble de triangles connectés. On parle de sommets pour désigner les points définissant les faces triangulaires (3 points distincts forment un triangle) et les arêtes (2 points distincts appartenant à un même triangle forment une arête). Deux triangles adjacents dans un maillage partagent un sommet commun ou une arête commune. Cette définition s'étend évidemment aux maillages polygonaux de manière générale.

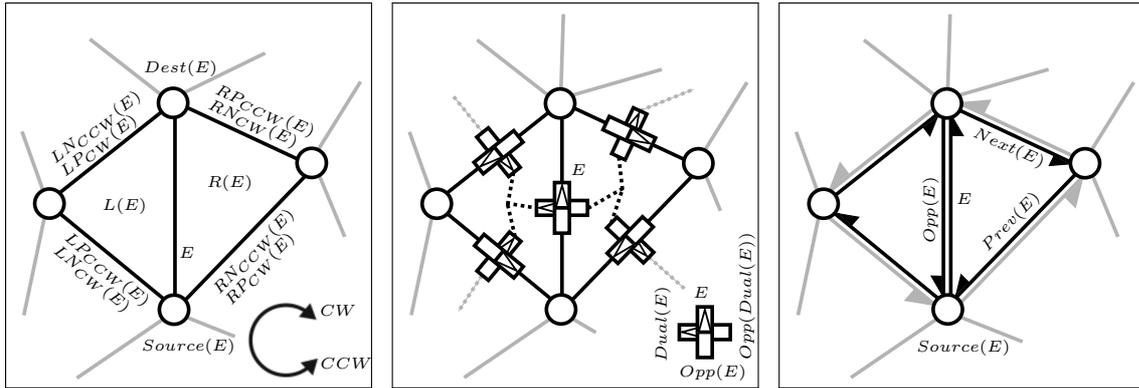
Généralement, on considère que les sommets échantillonnent la surface représentée par le maillage. Comme tout échantillonnage, le maillage ne constitue qu'une information incomplète de la surface qu'il représente, et introduit donc un certain niveau d'approximation de la surface.

Il existe diverses manières de représenter informatiquement une structure de maillage. Les plus utilisées proposent une structure permettant un parcours efficace du maillage, et reposent sur la description des arêtes. L'arête ailée (*winged edge*) proposée par B.Baumgart [Bau72] propose un accès instantané à ses faces adjacentes ainsi qu'aux arêtes

---

<sup>1</sup>une  $n$ -variété dans  $\mathbb{R}^m$ , où  $m > n$ , est une notion topologique désignant un espace assimilable (plus rigoureusement, homéomorphe) localement à un espace euclidien de dimension  $n$ . Typiquement, une 2-variété dans  $\mathbb{R}^3$  est une surface de l'espace 3D usuel dont tout point est localement homéomorphe à un disque.

ailées suivantes dans une face selon un sens de parcours choisi (*ClockWise* ou *Counter-ClockWise*). La quad-arête (*quad edge*) proposée par L. Guibas et J. Stolfi [GS85] étend la demi arête en intégrant une représentation permettant de parcourir aussi bien le maillage primal que son maillage dual<sup>2</sup>. Quant aux demi arêtes proposées par M. Mantyla [Man88], elles sont orientées et proposent le parcours en sens fixe de chaque face du maillage. Chaque demi arête n'appartient qu'à une face, propose un accès immédiat à ses arêtes suivantes et précédentes, et possède une demi arête opposée sur la face adjacente.



(a) *Winged Edge*. L'arête  $E$  est rattachée à deux faces  $R(E)$  et  $L(E)$ . Selon le sens de parcours (indice  $CW$  ou  $CCW$ ), on accède à l'arête suivante ou précédente (suffixe  $N/P$ ) sur la face voulue (préfixe  $R/L$ ).

(b) *Quad Edge*. Aux winged edges, on ajoute la possibilité d'intervir sommets et faces, via les arêtes duales  $Dual(E)$ , et d'inverser le sens de parcours via les arêtes opposées  $Opp(E)$  ou  $Opp(Dual(E))$ .

(c) *Half Edge*. L'arête  $E$  est orientée, et possède des arêtes suivante et précédente ( $Next(E)/Prev(E)$ ). On passe d'un triangle à son triangle adjacent via l'arête opposée  $Opp(E)$ .

**Figure 1.1:** Descriptions pour le parcours efficace d'un maillage triangulaire.

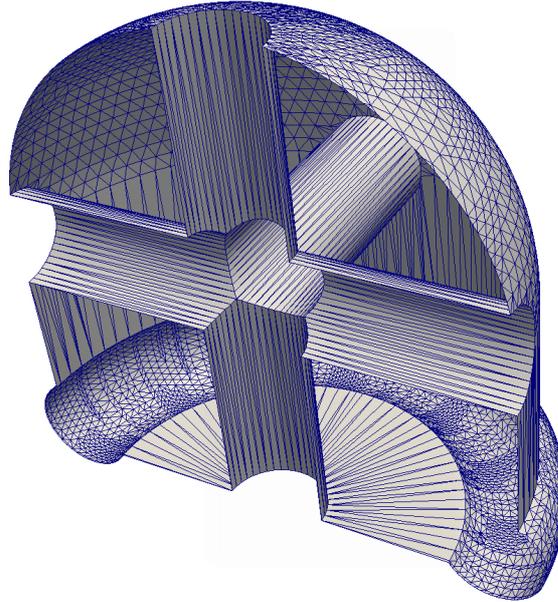
De très nombreux travaux portent sur les maillages triangulaires, qui forment à eux seuls un champ entier de la géométrie algorithmique, via notamment les triangulations de Delaunay ou diagrammes de Voronoï, les maillages adaptatifs, les mesures de qualité de maillages, etc. Ils sont aussi très utilisés en reconstruction de modèles 3D à partir de nuages de points.

La pertinence de la description systématique des surfaces par les maillages est aujourd'hui discutable. Il est toutefois plus facile, voire parfois indispensable, de raisonner sur une surface discrétisée plutôt que sur une surface continue.

### 1.1.2 Représentation volumique discrète

Le concept de discrétisation de surface s'étend tout naturellement à la discrétisation de volumes. La discrétisation volumique trouve son application principale dans la simulation physique des milieux continus. La méthode des éléments finis par exemple, requiert un découpage fin en éléments de l'espace pour permettre la résolution d'équations aux dérivées partielles (EDP) définies sur l'espace continu, que l'on retrouve typiquement dans les équations d'écoulement des fluides ou d'échanges thermiques. La finesse de la discrétisation

<sup>2</sup>Le maillage dual est un maillage dont les sommets correspondent aux faces du maillage primal (d'origine), et dont les faces correspondent aux sommets du maillage primal. On constate ainsi que le dual du dual est le maillage primal.



**Figure 1.2:** Exemple de maillage surfacique (triangulation). La coupe du maillage montre que seule l'interface entre l'objet et l'extérieur est effectivement encodée par le maillage surfacique.

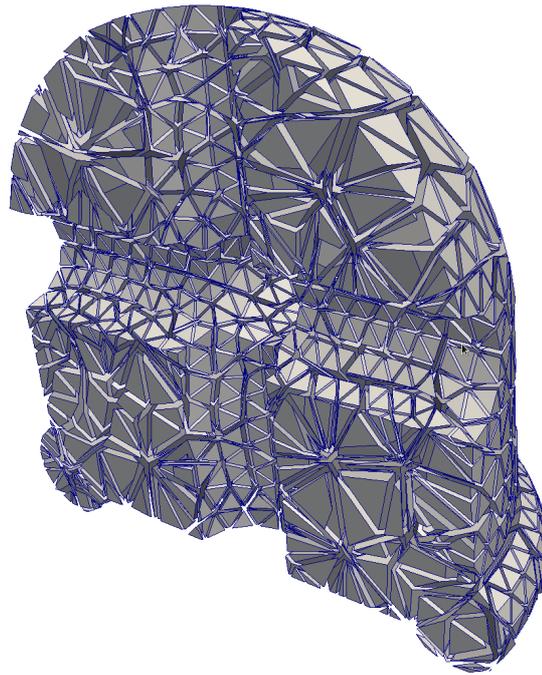
est un élément critique de la méthode, puisqu'il contribue à l'exactitude de la solution calculée. Les systèmes masses-ressorts quant à eux reposent sur une discrétisation du volume de l'objet étudié pour permettre la simulation de son comportement lorsqu'il est soumis à diverses forces. Ils proposent une représentation basée sur des particules échantillonnant le volume, reliées de proche en proche par des ressorts régissant le comportement interne de la forme. Plus l'échantillonnage de l'objet est fin, plus la simulation du comportement est exacte.

Certaines méthodes informatiques reposent sur le découpage de l'espace en une grille régulière, proposant ainsi un équivalent tridimensionnel des images. Par analogie avec les pixels des images, on parle d'ailleurs de voxel pour désigner un élément de volume auquel est rattaché une information (intensité, couleur, densité, énergie, etc.).

De la même manière que les maillages surfaciques, les maillages volumiques se définissent comme des assemblages de primitives volumiques adjacentes (généralement des tétraèdres ou hexaèdres). Deux éléments volumiques adjacents partagent une face commune, les faces étant délimitées par des arêtes et les arêtes par des sommets définis par leur position dans l'espace. Il existe différentes approches traitant de la génération de maillages volumiques, en fonction de la nature des données à partir desquelles la discrétisation doit être calculée et de l'application dans laquelle le maillage doit être utilisé. Les calculs physiques sont par exemple très contraignants quant à la qualité des maillages sur lesquels ils sont effectués.

### 1.1.3 Représentations continues

Contrairement aux représentations discrètes reposant sur un échantillonnage des surfaces ou volumes traités, les représentations continues permettent une description exacte de surfaces parfois complexes à l'aide d'une faible quantité d'informations. Bien que les surfaces ainsi définies soient caractérisées formellement, elle peuvent toutefois ne constituer qu'une approximation de la réalité traitée, puisque cette dernière ne se résume souvent



**Figure 1.3:** Exemple de maillage volumique (tétraédrisation). La coupe du maillage montre que l'intérieur de l'objet est décomposé en tétraèdres adjacents, et que les parties creuses du solide ne sont pas maillées, respectant ainsi la topologie initiale de l'objet.

pas à des objets analytiques: la représentation d'un tuyau par un cylindre ne permet typiquement qu'une représentation inexacte de la réalité du terrain, car les tuyaux sont soumis localement à des déformations d'amplitude variable (chocs, distorsions, etc.). En ce sens, les objets définis dans le continu n'ont pas un pouvoir de représentation supérieur aux maillages, puisqu'ils ne sont pas plus exacts.

Les modèles 3D issus de Conception Assistée par Ordinateur (C.A.O.) s'inscrivent dans un schéma de représentation continue des surfaces et volumes modélisés. Un objet complexe se décrit alors comme un assemblage d'une quantité très limitée d'éléments atomiques qui sont potentiellement de natures diverses. Cet assemblage peut établir une hiérarchie dans la scène, permettant un enrichissement sémantique de la représentation qu'encode le modèle 3D CAO. Prenons l'exemple des environnements industriels: il est courant de regrouper les composants selon le rôle qu'ils occupent dans l'environnement (tuyauteries, machineries, charpentes, etc.), puis de regrouper ces structures en fonction de la salle à laquelle ils appartiennent, puis de regrouper les salles par bâtiment, etc.

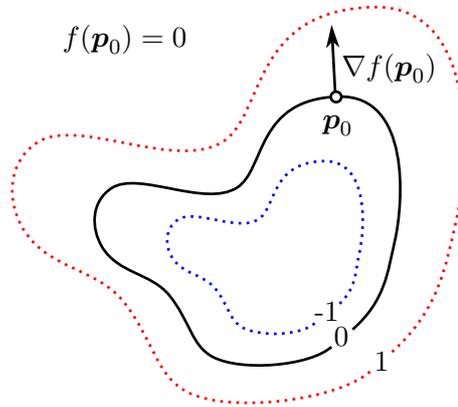
Cette section présente certains outils couramment utilisés pour permettre une description CAO des objets représentés.

### Primitives CAO: solides et surfaces

Les éléments atomiques qui composent un modèle CAO peuvent s'exprimer de manières diverses. Certaines représentations permettent de définir l'objet comme un solide, tandis que d'autres ne définissent que sa surface.

**Surfaces implicites** Les surfaces implicites (introduites en CAO par A. Ricci [Ric73]) reposent sur la définition d'un champ de potentiel engendré par une fonction définie en tout point de l'espace  $\mathbb{R}^3$ . En évaluant cette fonction en un point donné, on sait alors

dire, selon la valeur retournée (le "potentiel"), si le point se trouve à l'intérieur de l'objet représenté, sur sa surface, ou bien à l'extérieur. Habituellement, la surface est considérée comme étant l'ensemble des points de potentiel nul, tandis que l'intérieur correspond aux potentiels négatifs. Les propriétés différentielles de la fonction utilisée permettent en outre de définir les normales à la surface de l'objet représenté (correspondant au gradient de la fonction).



**Figure 1.4:** Surface implicite en 2D définie par une fonction  $f$ . La courbe de niveau nul  $\{\mathbf{p} \in \mathbb{R}^2 | f(\mathbf{p}) = 0\}$  correspond à la surface de l'objet représenté. L'intérieur se caractérise par des valeurs négatives de la fonction tandis que l'extérieur est positif.

Exemple: La représentation implicite du cylindre (infini)  $\mathcal{C}$  de rayon  $r_{\mathcal{C}}$  et dont l'axe est défini par un point  $\mathbf{c}_{\mathcal{C}}$  et un vecteur directeur unitaire  $\mathbf{a}_{\mathcal{C}}$  est engendrée par la fonction de distance à l'axe:

$$f(\mathbf{p}) = \|(\mathbf{p} - \mathbf{c}_{\mathcal{C}}) \times \mathbf{a}_{\mathcal{C}}\| - r_{\mathcal{C}}$$

**Surfaces paramétrées** Cette alternative aux surfaces implicites consiste en la définition de fonctions définies sur un espace paramétrique  $\mathbb{P}$  différent de l'espace tridimensionnel géométrique auquel appartiennent les primitives. Les points  $\mathbf{p}$  décrivant la surface d'un solide s'expriment alors sous la forme:

$$\mathbf{p} = (f_X(\mathbf{t}), f_Y(\mathbf{t}), f_Z(\mathbf{t}))^T \quad \text{où } \mathbf{t} \in \mathbb{P}$$

Le balayage de l'espace paramétrique permet alors l'exploration de la surface de l'objet par le biais des fonctions  $f_X$ ,  $f_Y$  et  $f_Z$ .

En adaptant  $\mathbb{P}$ , il est possible d'étendre cette description pour permettre la représentation du volume du solide. Le balayage de l'espace paramétrique permet alors l'exploration de l'intérieur de l'objet.

Exemple: La représentation de la surface du cylindre (infini)  $\mathcal{C}$  mentionné précédemment est paramétrée par l'angle  $\phi$  et la hauteur  $h$ :

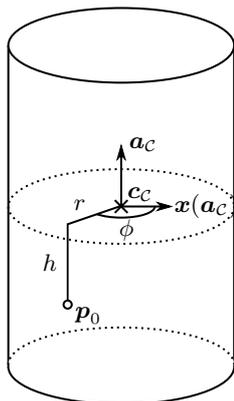
$$\begin{pmatrix} f_X(\phi, h) \\ f_Y(\phi, h) \\ f_Z(\phi, h) \end{pmatrix} = \mathbf{c}_{\mathcal{C}} + r_{\mathcal{C}} (\cos(\phi)\mathbf{x}(\mathbf{a}_{\mathcal{C}}) + \sin(\phi)(\mathbf{a}_{\mathcal{C}} \times \mathbf{x}(\mathbf{a}_{\mathcal{C}}))) + h\mathbf{a}_{\mathcal{C}}$$

avec  $\mathbb{P} = [0, 2\pi[ \times \mathbb{R}$  (où  $\phi \in [0, 2\pi[$  et  $h \in \mathbb{R}$ ).  $\mathbf{x}(\mathbf{a}_{\mathcal{C}})$  est une fonction renvoyant un vecteur orthogonal à  $\mathbf{a}_{\mathcal{C}}$ .

En ajoutant un paramètre  $r$  permettant de balayer les rayons, et en bornant le domaine de définition de  $h$ , on peut alors définir paramétriquement le volume du cylindre tronqué  $\mathcal{C}$  de longueur  $l_{\mathcal{C}}$ :

$$\begin{pmatrix} f_X(\phi, h, r) \\ f_Y(\phi, h, r) \\ f_Z(\phi, h, r) \end{pmatrix} = \mathbf{c}_{\mathcal{C}} + r(\cos(\phi)\mathbf{x}(\mathbf{a}_{\mathcal{C}}) + \sin(\phi)(\mathbf{a}_{\mathcal{C}} \times \mathbf{x}(\mathbf{a}_{\mathcal{C}}))) + h\mathbf{a}_{\mathcal{C}}$$

avec  $\mathbb{P} = [0, 2\pi[ \times [-\frac{l_{\mathcal{C}}}{2}, \frac{l_{\mathcal{C}}}{2}] \times [0, r_{\mathcal{C}}]$  (où  $\phi \in [0, 2\pi[$ ,  $h \in [-\frac{l_{\mathcal{C}}}{2}, \frac{l_{\mathcal{C}}}{2}]$  et  $r \in [0, r_{\mathcal{C}}]$ ).



**Figure 1.5:** Représentation paramétrique d'un volume cylindrique tronqué. Tout point  $\mathbf{p}_0$  à l'intérieur du cylindre peut être déterminé à partir de son azimut  $\phi$ , de sa hauteur  $h$  et de sa distance radiale  $r$ .

**Surfaces à pôles** Les surfaces à pôles correspondent à des surfaces paramétrées sur  $\mathbb{P} = [0, 1]^2$  définies à partir d'une combinaison linéaire de points  $(\mathbf{q}_0, \dots, \mathbf{q}_{s-1})$  que l'on nomme points de contrôle. Ces points de contrôle n'appartiennent généralement pas à la surface qu'ils caractérisent, mais participent à la définition de sa géométrie. Les surfaces à pôles sont de plus caractérisées par des degrés  $m$  et  $n$  permettant principalement d'ajuster les oscillations de la surface (plus  $m$  et  $n$  sont faibles, plus la surface est "tendue" entre les points de contrôle). Tout point  $\mathbf{p}$  appartenant à une telle surface s'exprime de la manière suivante:

$$\mathbf{p} = \begin{pmatrix} f_X(u, v) \\ f_Y(u, v) \\ f_Z(u, v) \end{pmatrix} = \frac{\sum_{i=0}^n g_i^{m,n}(u, v) w_i \mathbf{q}_i}{\sum_{i=0}^n g_i^{m,n}(u, v) w_i}$$

Les facteurs  $w_i$  sont des poids associés aux points de contrôle, pour permettre d'en définir l'importance dans la définition de la surface. La nature de la surface dépend principalement de l'expression de la fonction  $g_i^{m,n}(u, v)$  définie sur  $[0, 1]^2$ . Les surfaces B-Splines Rationnelles Non Uniformes (NURBS), qui sont les plus utilisées, sont définies à partir de fonctions de base B-Splines  $N_i^d$ :

$$g_i^{m,n}(u, v) = N_i^m(u) \times N_i^n(v)$$

où la fonction B-Spline  $N_i^d(x)$  est définie récursivement:

$$N_i^0(x) = \begin{cases} 1 & \text{si } x_i \leq x < x_{i+1} \\ 0 & \text{sinon} \end{cases}$$

$$N_i^k(x) = \frac{x - x_i}{x_{i+1} - x_i} N_i^{k-1}(x) + \frac{x_{i+k+1} - x}{x_{i+k+1} - x_{i+1}} N_{i+1}^{k-1}(x)$$

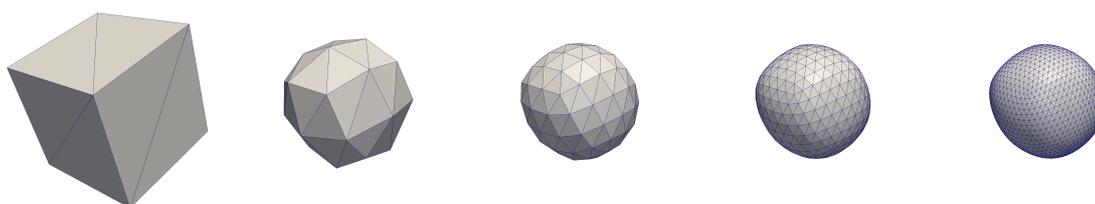
Les NURBS nécessitent, en plus des degrés  $m$  et  $n$  et de la liste de  $s$  points de contrôle, la définition de deux séquences croissantes,  $(u_0, \dots, u_{m+s})$  associée à  $N_i^m(u)$  et  $(v_0, \dots, v_{n+s})$  associée à  $N_i^n(v)$ , que l'on nomme séquences nodales et dont le rôle est la détermination de l'influence de chaque point de contrôle sur la surface résultante (chaque point ne participe réellement qu'à la définition d'une partie de la surface). Un élément  $u_i$  ou  $v_j$  d'une de ces séquences est appelé "nœud", et intervient dans l'équation précédente sous la notation " $x_i$ " selon que l'on considère le paramètre  $u$  (via  $N_i^m$ ) ou  $v$  (via  $N_i^n$ ).

Les fonctions de base B-Spline  $N_i^d$  peuvent être remplacées par des polynômes de Bernstein de degré approprié pour définir des carreaux de Bézier. Cependant, les possibilités offertes par les surfaces de Bézier sont très limitées par rapport à ce que permettent les NURBS.

Le succès des NURBS tient au fait qu'elles permettent de représenter une grande variété de surfaces, mais aussi et surtout au fait qu'elles peuvent être éditées très facilement et intuitivement en manipulant les points de contrôle qui les définissent.

**Surfaces de subdivision** Les surfaces de subdivision permettent théoriquement d'obtenir une surface lisse à partir d'un maillage surfacique initial grossier. Le calcul permettant d'aboutir à la surface finale procède par subdivisions successives. La subdivision est une procédure définissant la construction d'un nouvel ensemble de faces, sommets et arêtes calculés à partir des éléments du maillage courant, augmentant à chaque étape le nombre de faces du maillage. Appliquée récursivement, la subdivision augmente la finesse du maillage, ainsi que sa précision vis-à-vis de la surface finale qui constitue le point fixe vers lequel converge le processus si l'on applique les subdivisions à l'infini. Habituellement, les subdivisions sont encodées sous forme de matrices qui, lorsqu'elles sont appliquées aux sommets du maillage courant, permettent de calculer les coordonnées des sommets du maillage à l'étape suivante.

Le célèbre schéma de subdivision proposé par E. Catmull et J. Clark [CC78] par exemple permet d'obtenir une surface B-Spline à l'issue du processus récursif de subdivision d'un maillage rectangulaire initial. Ce résultat peut être étendu aux surfaces non rectangulaires, avec l'apparition potentielle d'un nombre fini de points singuliers sur la surface. Le schéma de subdivision proposé par C. Loop [Loo87] est lui aussi très utilisé, et propose une méthode portant uniquement sur les maillages triangulaires.



**Figure 1.6:** Schéma de subdivision de Loop appliqué à un maillage triangulaire plongé dans  $\mathbb{R}^3$ . Le maillage initial considéré est ici un cube (vignette de gauche), chaque vignette représentant ensuite une étape supplémentaire de subdivision.

**Primitives géométriques paramétrées** Les primitives géométriques paramétrées proposent un ensemble de formes simples définies chacune par une faible quantité de paramètres, parmi lesquels on distingue les paramètres de position et d'orientation définissant le placement de la primitive dans l'espace, et les paramètres géométriques définissant la forme de la primitive. Ce type de représentation rassemble une famille d'objets pouvant être décrits par des surfaces implicites ou paramétrées mentionnées précédemment (voire même avec

les surfaces à pôle ou de subdivision), sans toutefois tenir compte du formalisme mathématique permettant la description des formes. L'annexe A liste un certain nombre de primitives géométriques utilisables en CAO.

### Assemblage CAO

Pour permettre la représentation de surfaces ou solides complexes, les modèles CAO reposent sur une méthode d'assemblage de primitives. Les modèles CAO doivent typiquement permettre la représentation de solides percés, fraisés, chanfreinés, etc. et l'agrégation des pièces pour former le composant final. Deux principales approches se distinguent en la matière: les *Constructive Solid Geometries* et les *Boundary Representations*.

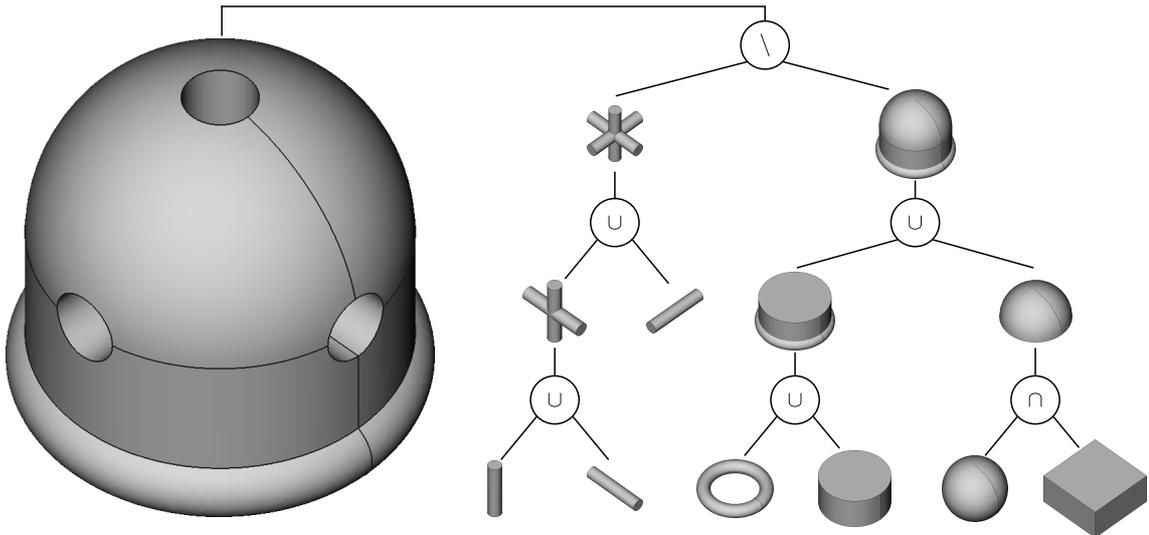
**Constructive Solid Geometry (CSG)** L'approche CSG est essentiellement associée à la représentation de volumes (solides). Elle procède par construction d'un arbre binaire d'assemblage de solides basé sur l'utilisation d'opérateurs ensemblistes. Les trois opérateurs utilisables sont en général l'union  $\cup$ , l'intersection  $\cap$  et la différence ensembliste  $\setminus$  dans  $\mathbb{R}^3$ , chaque solide étant perçu comme un fermé dans  $\mathbb{R}^3$ . Les feuilles de l'arbre CSG sont des primitives géométriques représentées sous une certaine forme, typiquement des solides paramétrés (ou implicites). Chaque nœud interne de l'arbre CSG représente un solide correspondant au résultat de l'application d'un opérateur sur ses fils. La remontée dans l'arbre correspond donc à l'assemblage des formes, et le solide construit se complexifie ainsi en approchant de la racine qui correspond à l'objet modélisé.

La modélisation par CSG possède des limites inhérentes à la quantité finie de primitives à partir desquelles les modèles sont construits. Tout objet ne peut pas nécessairement se représenter comme une combinaison d'un nombre limité d'objets atomiques, aussi complexe que soit cette combinaison. L'ensemble des primitives dont on dispose à l'origine détermine donc les capacités de représentation de la modélisation CSG. Toutefois, la simplicité de cette approche la rend très intuitive et ne nécessite pas d'expertise spécifique de la part de l'utilisateur, qui peut construire un modèle CAO élaboré en quelques étapes sur la seule base de sa connaissance (même intuitive) de ce que sont l'union, l'intersection et la différence de solides.

**Boundary Representation (B-Rep)** Cette approche repose sur la notion de bordure à différents niveaux. Ainsi, un solide est considéré comme étant limité par des patches surfaciques "collés" les uns aux autres. Une surface est elle-même délimitée par une collection d'arêtes adjacentes. Pour finir, chaque arête se termine en deux sommets. La continuité de la représentation est assurée par le fait qu'une même bordure de dimension donnée puisse être partagée par plusieurs bordures de dimension supérieure: deux volumes adjacents partagent une face commune, deux faces adjacentes partagent une arête commune et deux arêtes adjacentes partagent un sommet commun.

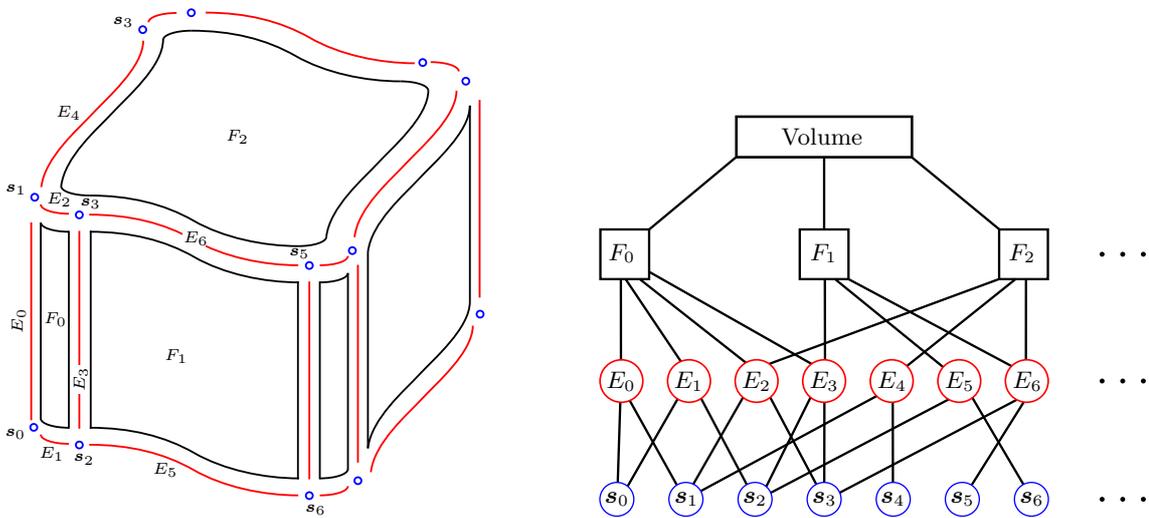
Dans le formalisme B-Rep, on distingue ainsi la géométrie de la topologie. Alors que la géométrie est portée par les bordures quelle que soit la dimension considérée, la topologie supervise la manière dont les bordures se greffent les unes aux autres, permettant ainsi d'assurer "l'étanchéité" de la représentation à la jonction des éléments.

Géométriquement, les faces B-Rep sont généralement représentées sous forme de surfaces à pôles (NURBS, Bézier, etc.), mais peuvent s'appuyer sur d'autres représentations. Par extension, les arêtes sont habituellement représentées par des courbes à pôles. Les sommets sont quant à eux spécifiés par leurs coordonnées cartésiennes définissant leur position.



**Figure 1.7:** Arbre de construction CSG du modèle CAO à partir duquel on a été générés les maillages en figure 1.2 et 1.3. L'opérateur majoritaire est ici l'union, bien qu'apparaissent une intersection pour créer l'hémisphère (en bas à droite) et une différence pour permettre de percer le solide (en haut).

La topologie quant à elle peut être encodée sous forme d'un treillis, dont chaque niveau correspond à une dimension (0 pour les sommets, 1 pour les arêtes, 2 pour les faces et 3 pour les volumes). Chaque sommet du treillis représente une bordure, et les arêtes représentent la relation de délimitation: une bordure de dimension  $D+1$  est reliée à une bordure de dimension  $D$  si cette dernière participe à la délimitation de la première.



**Figure 1.8:** Exemple de modèle CAO en B-Rep. Le schéma de gauche représente la géométrie du solide. Les faces y apparaissent en noir, les arêtes en rouge et les sommets en bleu. Le schéma de droite présente une partie du treillis décrivant la topologie du solide, en s'appuyant sur le même code couleur.

Les possibilités de modélisation offertes par les B-Rep sont très supérieures à celles des CSG. Mais l'implémentation du formalisme B-Rep est complexe, et son utilisation requiert une meilleure maîtrise des notions géométriques sous-jacentes.

**En pratique** les approches CSG et B-Rep peuvent être complémentaires. Il n'est ainsi pas rare que le résultat d'une construction à base de CSG soit représentée, *in fine*, à

l'aide du formalisme B-Rep. Les arêtes visibles dans la figure 1.7, par exemple, indiquent que le modèleur utilisé procède au calcul d'une représentation B-Rep du modèle CSG pour en permettre la visualisation. De plus, certains éléments ne pouvant être définis par CSG, tels que les soudures, congés, bossages, etc. peuvent être représentés par des surfaces prédéfinies, le résultat étant ensuite décrit sous forme de modèle B-Rep.

Ces formalismes ne constituent pas non plus les seules alternatives permettant de représenter ou stocker les données CAO. Il arrive que les primitives géométriques soient simplement listées, éventuellement sous la forme d'une arborescence représentant la hiérarchie de la scène. Rien ne permet de garantir la cohérence de tels modèles, puisque les relations entre primitives n'y sont pas représentées, contrairement aux B-Rep et CSG reposant sur une structuration forte de la scène. Cependant, les problématiques d'échange de données CAO et les possibilités restreintes offertes par les éditeurs de logiciel quant à l'export des modèles ne permettent pas nécessairement d'adopter une modélisation de haut niveau. Finalement, cette solution simpliste peut s'avérer être un des seuls choix envisageable en matière d'échange de données CAO.

## 1.2 Nuages de points

Le nuage de points représente une collection de mesures 3D permettant de décrire la surface<sup>3</sup> d'une scène telle qu'elle est en réalité. La structure d'un nuage de points dépend de la source dont proviennent les mesures, et les possibilités permettant l'acquisition de mesures 3D sont variées. Le nuage de points fraîchement acquis est couramment soumis à un certain nombre de prétraitements destinés à corriger ou compléter les données, et déterminer certaines de ses caractéristiques.

### 1.2.1 Acquisition

On distingue deux principales techniques d'acquisition de grandes quantités de mesures 3D dans des scènes de grande échelle.

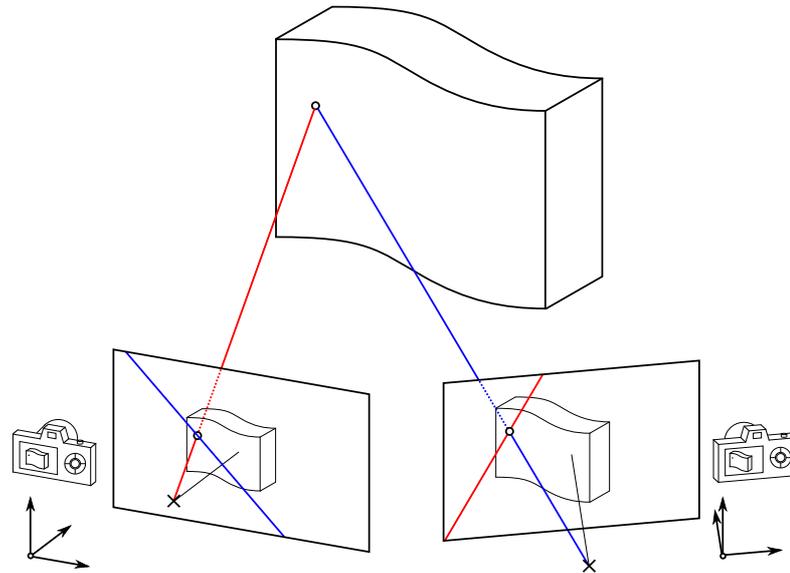
**Photogrammétrie** La photogrammétrie permet la reconstruction de données tridimensionnelles à partir de photographies. La technique photogrammétrique la plus utilisée repose sur l'analyse des dissimilarités induites par la parallaxe lors de l'acquisition d'images d'une même scène à partir de points de vue distincts. Après avoir identifié avec précision la position d'un point d'intérêt de la scène étudiée dans différentes images, on sait retrouver la position 3D de ce point si l'on connaît précisément les mécanismes géométriques d'acquisition de chaque image ainsi que les positions et orientations des appareils lors des prises de vue.

Pour permettre la stéréorestitution, c'est-à-dire la reconstruction d'une information tridimensionnelle à partir de plusieurs informations issues d'images 2D, il est donc avant tout nécessaire de se donner un modèle décrivant le capteur ayant servi pour l'acquisition des photographies (typiquement le modèle de chambre à sténopé). Le capteur est défini dans ce modèle par un certain nombre de paramètres "intrinsèques" décrivant son état au moment de l'acquisition ( focale, distorsion, centre optique, etc.), ainsi que des paramètres "extrinsèques" décrivant sa position et son orientation dans un repère choisi, commun à toutes les prises de vue. La donnée du modèle et des paramètres des capteurs permet la mise en place d'une description algébrique du phénomène d'acquisition, dont l'inversion

---

<sup>3</sup>Les nuages de points peuvent aussi représenter le volume solide d'une scène dans certaines applications, mais nous ne considérons ici que les échantillonnages surfaciques

permet de déduire l'expression algébrique du calcul des données tridimensionnelles à partir des données bidimensionnelles (via les matrices essentielle et fondamentale).



**Figure 1.9:** Principe de la stéréorestitution en photogrammétrie. La position d'un point sur l'objet photographié sous deux points de vue peut être recalculée si ce point est identifié dans chaque photographie. On note que les droites épipolaires, correspondant à la projection du rayon optique issu d'une photo dans la seconde, peuvent aider à retrouver le point d'une image à l'autre.

La redondance des points de vue lorsque le nombre de photographies est supérieur à trois permet de renforcer l'exactitude des données reconstruites.

La tâche la plus complexe dans la reconstruction de nuages de points à partir de photographies est la mise en correspondance automatique des points dans les différentes images. Cette procédure peut être guidée grâce à la géométrie épipolaire, afin d'en augmenter l'efficacité mais aussi de limiter les erreurs d'appariement des points. La géométrie épipolaire stipule qu'un point d'une image ne peut se trouver que sur une ligne dans une seconde image, et explicite le calcul de cette ligne.

**Light Detection And Ranging (LIDAR)** La technologie LIDAR [PB06] propose des capteurs fonctionnant sur le principe d'émission de faisceaux laser, permettant l'acquisition précise de quantités énormes de mesures. Les appareils laser de génération actuelle proposent par exemple<sup>4</sup> une précision de 6 millimètres sur des distances inférieures à 50 mètres, avec 1 point tous les 2 millimètres à une distance de 10 mètres et une vitesse d'acquisition de 500000 points par seconde.

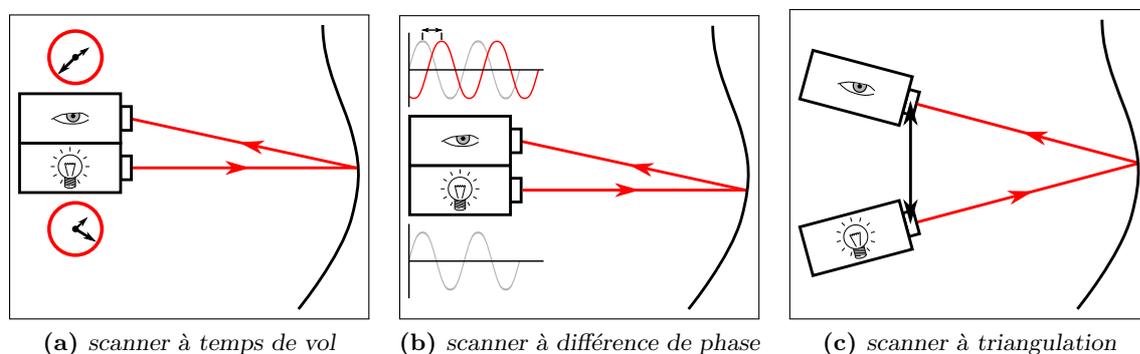
De manière générale, l'acquisition laser procède par estimation de la distance séparant le capteur de l'obstacle le plus proche dans une direction donnée. La collecte des données 3D s'effectue alors en balayant l'ensemble des directions accessibles au capteur (pour les scanners laser terrestre, selon un angle vertical d'une part, et horizontal d'autre part). Le procédé de calcul de la distance varie selon la technologie employée:

**temps de vol:** la mesure du temps de vol consiste en la détermination du temps séparant l'émission et la réception d'un signal lumineux. Connaissant la vitesse du signal lumineux dans le milieu dans lequel il se propage lors de l'acquisition, on peut alors calculer simplement la distance qu'il aura parcourue.

<sup>4</sup>Les données numériques citées sont issues des spécifications du Leica HDS 6000

**différence de phase:** la mesure de la différence de phase entre le signal émis et le signal reçu permet, étant données la fréquence du signal et sa vitesse dans le milieu de propagation, de déduire son temps de vol et donc la distance qu'il aura parcourue. La principale limitation de cette technique tient au fait que la détermination de la différence de phase est par nature ambiguë pour un signal périodique. On peut toutefois contourner cette difficulté en modulant les fréquences des signaux émis. Les technologies basées sur la mesure de la différence de phase sont celles présentant les plus grandes vitesses d'acquisition des données.

**triangulation:** cette technologie s'apparente à la photogrammétrie. Un capteur matriciel est positionné à une distance connue de l'émetteur laser. Lorsque le signal est reçu par le capteur, on peut alors calculer la position à partir de laquelle le signal a été réfléchi, puisque l'on connaît les positions relatives de l'émetteur et du récepteur.



**Figure 1.10:** Principe d'acquisition par lasergrammétrie (LIDAR). Les trois technologies illustrées ici ne se distinguent que par la manière dont elles interprètent la réception du signal lumineux émis par le scanner.

En tous les cas, le capteur doit interpréter un signal analogique se présentant comme un pic de luminosité correspondant au retour du signal lumineux. Ce pic est plus ou moins concentré selon les conditions d'acquisition, notamment selon l'étendue de l'empreinte laser à la surface de l'obstacle. Moins cette tâche est dispersée, plus le retour du signal est énergétique, donc plus le pic est concentré et plus l'évaluation de la distance est précise. Or plus l'angle d'incidence du faisceau laser sur la surface est rasant, plus l'empreinte laser est dispersée. Les travaux de S. Soudarissanane et al. [SLMT09] proposent un étude poussée de ce phénomène.

Certains capteurs sont en mesure de traiter des retours multi-échos, correspondant à la réflexion du faisceau laser par de multiples surfaces (arbres, fonds océaniques, etc.). Au lieu de fournir des nuages de points, d'autres capteurs fournissent un retour d'onde complète (*full waveform*) pour chaque faisceau émis, *i.e.* un échantillonnage de la puissance lumineuse reçue au cours du temps, permettant une étude poussée de la structure de la scène. Ces données peuvent alors faire l'objet de prétraitements pour identifier les pics du signal analogique original [MLB<sup>+</sup>09].

**À petite échelle** L'acquisition de surfaces sur des objets de taille moyenne (taille humaine) fait habituellement appel à des technologies plus adaptées aux dimensions des scènes traitées. On pourra mentionner dans cette catégorie les appareils tactiles permettant, à l'aide d'un bras mécanisé muni d'un capteur, d'effectuer des mesures de position par contact avec la surface à acquérir. En outre, il existe des appareils LIDAR mobiles légers qui,

lorsqu'ils sont couplés à des dispositifs de suivi de la position de l'appareil d'acquisition porté à bout de bras, permettent des relevés extrêmement précis<sup>5</sup>.

Les scanners à lumière structurée permettent d'approfondir l'approche photogrammétrique pour permettre des relevés précis (l'idée peut aussi s'appliquer à grande échelle, quoique certaines difficultés apparaissent rapidement, en extérieur typiquement). Leur principe consiste en la projection de motifs lumineux connus sur la surface de l'objet relevé, par exemple une alternance de franges lumineuses sombres et claires. Les déformations de ces motifs tels que nous les percevons à la surface de l'objet après photographie révèlent son relief, qui peut ainsi être reconstruit en 3D.

**2D et demi ou 3D** Selon la technologie employée, les données résultantes peuvent se présenter sous forme "d'images de profondeur" (*range image* en anglais, pour un relevé laser à écho unique typiquement) ou de mesure réellement tridimensionnelles (stéréophotogrammétrie, fusion de relevés, ou relevé laser multi-échos). La nuance tient au fait que les images de profondeur ne permettent de décrire qu'une unique information par direction de visée, et les données résultantes ne sont finalement qu'une collection de profondeurs organisées en une grille, à la manière d'une image dont l'intensité des pixels correspondrait à une distance. Chaque mesure peut être caractérisée par l'identifiant du pixel représentant la direction de visée, et la profondeur mesurée. On parle alors de données en "deux dimension et demi", par opposition aux données tridimensionnelles qui ne peuvent être décrites dans un tel formalisme, puisqu'une même direction de visée à partir d'un appareil peut fournir une multitude de mesures. Les points ne peuvent, dans ce dernier cas, être décrits autrement que par trois paramètres, et sont naturellement désordonnés.

En pratique, cette différence permet de nuancer le traitement des données selon que l'on traite des images de profondeur ou non. En effet, les images de profondeur induisent une structure forte parmi les données, puisqu'elles définissent naturellement une notion de voisinage entre les mesures. Le fait de relier simplement les points voisins (en traitant éventuellement les sauts de profondeur) permet ainsi d'obtenir à moindre effort un maillage de bonne qualité topologique. Cette opération n'est évidemment pas aussi simple pour les données tridimensionnelles désorganisées.

**Données complémentaires** Il n'est pas rare que les données tridimensionnelles soient accompagnées de données complémentaires liées à la texture des surfaces acquises. La photogrammétrie permet par exemple d'associer directement à chaque point la couleur de la surface qu'il échantillonne. Les couleurs peuvent aussi être obtenues par acquisition LIDAR, lorsque le scanner laser est couplé à un appareil photographique.

L'intensité du retour laser dans le cadre d'acquisition LIDAR peut aussi être fournie avec les données tridimensionnelles. Ces mesures renseignent notamment la nature des surfaces rencontrées par le faisceau laser.

### 1.2.2 Recalage

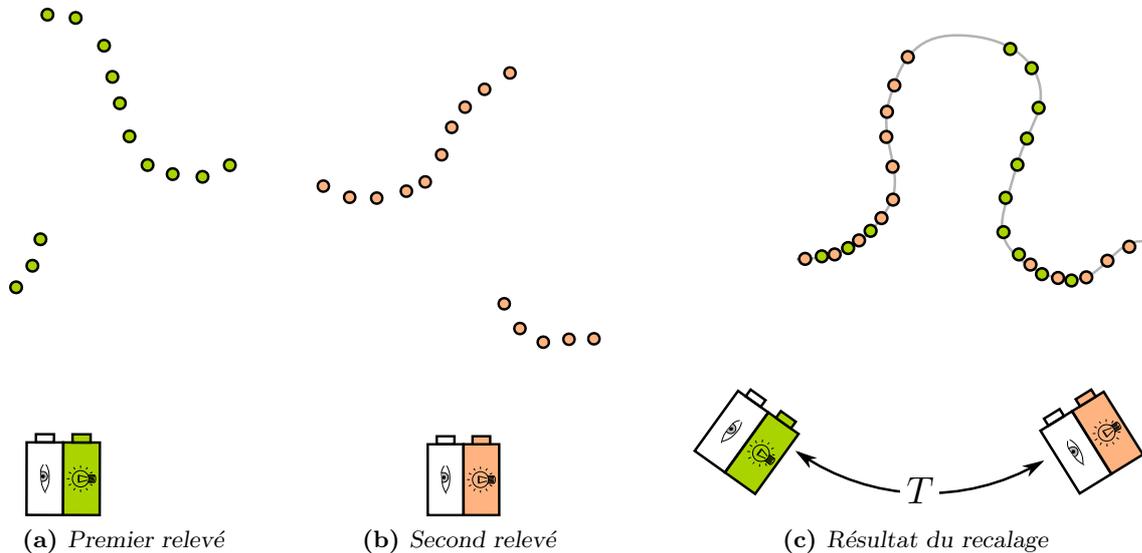
Le traitement des données 3D (LIDAR ou photogrammétrique) est différente selon la procédure d'acquisition, en particulier selon que les données ont été acquises à partir de points de vue fixes ou en mouvement.

Les acquisitions statiques nécessitent de multiplier les prises de vue pour assurer une couverture optimale de la surface d'intérêt, et éviter les absences d'information dues aux

---

<sup>5</sup>Les spécifications du T-Scan de Leica affichent une précision de mesure des distances inférieure au dixième de millimètre

occultations entre objets. Se pose alors le problème du recalage des différentes prises de vues, consistant en la recherche de la transformation rigide (rotation, translation et homothétie) permettant d'ajuster les nuages issus des différents relevés afin qu'ils correspondent à la surface d'origine lorsqu'ils sont tous exprimés dans un repère commun. Cette transformation dépend de la position et de l'orientation relative des appareils d'acquisition lors du relevé.



**Figure 1.11:** Recalage de deux prises de vue statiques. Le recalage trouve la transformation  $T$  permettant de positionner chaque station d'acquisition à sa position correcte dans un repère commun, de manière à ce que la fusion de vues permette de reconstituer la surface d'origine.

Ce problème du recalage se pose sous une forme différente lorsque l'acquisition est mobile. Il existe en effet de plus en plus de systèmes d'acquisition embarqués dans des véhicules (avion ou voitures), permettant le relevé 3D de scènes à l'échelle géographique. Typiquement, les modèles d'élévation numériques utilisés dans les systèmes d'information géographiques sont réalisés à partir de campagnes d'acquisition aéroportées. Chaque survol d'avion permet le relevé d'une bande de terrain de quelques centaines de mètres de large sur quelques kilomètres de long. Quant aux scènes urbaines, elles sont acquises grâce à des appareils mobiles le plus souvent intégrés à des voitures, permettant un relevé des façades de bâtiments sur le trajet du véhicule. En tous les cas, les données d'acquisitions mobiles sont généralement accompagnées de données de positionnement et d'orientation, collectées à partir d'appareillages spécifiques. Le plus souvent, il s'agit d'une centrale inertielle (accéléromètres et gyroscopes) couplée à un GPS pour limiter la propagation de l'erreur de positionnement. Le recalage des données d'acquisition mobile consiste en la correction des erreurs s'accumulant lors de la mesure des positions et orientations des appareils sur le trajet du relevé, ou en l'alignement des bandes issues des acquisitions aéroportées.

Nous ne nous intéresserons ici qu'au recalage des données issues d'acquisitions LIDAR terrestres statiques. La plupart des approches existantes dans le domaine se basent sur les travaux de B.K.P Horn [Hor87]. L'auteur y propose une formule calculant la transformation rigide alignant deux ensembles composés de plus de trois points au sens des moindres carrés.

Le problème du recalage de relevés entiers "se résume" alors à la recherche d'ensembles de plus de trois points qui se correspondent dans les différents relevés, puisque l'alignement de ces ensembles fournit la transformation permettant le recalage des nuages de points partiels. Ceci suppose notamment qu'il existe des recouvrements entre les différents ensembles de mesures, sans quoi le recalage est impossible. Plus ces recouvrements sont importants, plus les sous ensembles de points communs aux relevés sont étendus dans l'espace et plus le calcul de la rotation est contraint, renforçant ainsi la précision du calcul.

Lorsque les relevés sont initialement proches du résultat recherché, c'est-à-dire lorsque l'on dispose d'une estimation initiale des poses des appareils lors de l'acquisition, le recalage "fin" est généralement calculé à l'aide de l'algorithme Iterative Closest Point (ICP) [CM91, BM92], ou l'une de ses nombreuses variantes. Il consiste en l'alignement de deux nuages de points par appariement de chaque point d'un nuage avec le point du second nuage dont il est le plus proche. À chaque itération, cet appariement doit être recalculé, et le processus converge au fil des répétitions vers le minimum de l'erreur quadratique moyenne entre les deux nuages (si l'estimation initiale est suffisamment proche du résultat).

L'estimation grossière des poses des appareils peut être réalisée par différents moyens. Le plus immédiat consiste en la mesure des positions des scanners par des appareils de géopositionnement, ce qui n'est malheureusement pas toujours possible, en particulier en intérieur dans les environnements industriels. Par ailleurs, les géomètres topographes réalisant les relevés disposent d'une méthodologie permettant, par des mesures d'angles et de distance, de conserver une trace du cheminement topographique de l'acquisition, fournissant ainsi l'estimation initiale grossière des poses des scanners lors de la campagne de relevé. Une méthode alternative très utilisée en pratique consiste en la pose de cibles sur les surfaces acquises, permettant d'identifier précisément des points communs entre les différents relevés après que l'utilisateur ait identifié ces cibles parmi les données et spécifié leurs correspondances d'un relevé à l'autre. L'alignement de ces cibles permet alors d'obtenir un recalage plus ou moins précis de la scène (en fonction notamment du nombre de cibles), généralement affiné ensuite à l'aide de l'ICP. Il existe en outre un certain nombre d'approches automatiques mettant en œuvre des algorithmes cherchant les correspondances entre les différents relevés, comme par exemple l'algorithme *4-Points Congruent Sets* [AMCO08] basé sur la recherche stochastique des sous-ensembles de 4 points similaires entre nuages de points.

Une approche alternative émerge face au recalage à partir d'appariement de points: le recalage à partir de primitives géométriques [HGLT11, RDvV07]. L'intérêt de cette démarche repose sur la complexité des objets manipulés, qui permet d'introduire une grande quantité de contraintes et d'obtenir un recalage de grande précision avec peu d'objets identifiés dans les nuages de points. Elle permet notamment de s'affranchir des biais introduits par l'échantillonnage des surfaces que l'on cherche à recalcer. Cette approche repose toutefois sur la détection et l'appariement préalable de primitives géométriques dans les nuages de points à recalcer.

### 1.2.3 Normales

Un grand nombre d'algorithmes de traitement de nuages de points nécessitent d'associer à tout point du nuage un vecteur unitaire supposé normal à la surface échantillonnée. Ce vecteur est généralement fourni avec les données d'acquisition<sup>6</sup>, au même titre que les points. Toutefois, lorsque ce n'est pas le cas, il peut être estimé par le calcul à

<sup>6</sup>Les vecteurs normaux fournis sont calculés au moment de l'acquisition par le scanner, à l'aide d'une méthode qui n'est pas forcément communiquée par le fabricant de l'appareil

l'aide d'ajustement de plans locaux, comme le préconisent initialement H. Hoppe et al. [HDD<sup>+</sup>92]. Considérons un point  $\mathbf{p}$  du nuage de points dont on souhaite évaluer le vecteur normal associé, et  $\mathcal{N}(\mathbf{p})$  un ensemble de points du nuage voisins de  $\mathbf{p}$  (ses  $k$  plus proches voisins, ou les voisins contenus dans une sphère centrée sur  $\mathbf{p}$  de rayon donné). Si le voisinage de  $\mathbf{p}$  considéré est suffisamment petit et si la surface échantillonnée en  $\mathbf{p}$  est lisse, elle peut être approximée par un plan. Le vecteur normal associé à  $\mathbf{p}$  peut alors être calculé comme étant le vecteur normal au plan s'ajustant au mieux sur les points de  $\mathcal{N}(\mathbf{p})$ .

L'ajustement de plan au sens des moindres carrés se calcule par analyse en composantes principales (ACP), une méthode issue des statistiques permettant d'identifier les trois axes orthogonaux dont l'un maximise et un autre minimise la répartition des points après projection sur les axes. L'axe minimisant la répartition correspond au vecteur normal recherché. Concrètement, les axes se calculent comme étant les vecteurs propres de la matrice de variance-covariance de  $\mathcal{N}(\mathbf{p})$ , le vecteur normal étant associé à la plus petite valeur propre:

$$\Sigma(\mathcal{N}(\mathbf{p})) = \frac{1}{|\mathcal{N}(\mathbf{p})|} \sum_{\mathbf{q} \in \mathcal{N}(\mathbf{p})} (\mathbf{q} - \overline{\mathcal{N}(\mathbf{p})}) \cdot (\mathbf{q} - \overline{\mathcal{N}(\mathbf{p})})^T \quad (1.1)$$

$\overline{\mathcal{N}(\mathbf{p})}$  étant l'isobarycentre du nuage de voisins  $\mathcal{N}(\mathbf{p})$ :

$$\overline{\mathcal{N}(\mathbf{p})} = \frac{1}{|\mathcal{N}(\mathbf{p})|} \sum_{\mathbf{q} \in \mathcal{N}(\mathbf{p})} \mathbf{q} \quad (1.2)$$

L'ACP fournit de plus une mesure de la dispersion (assimilable à un écart type) le long de chaque axe via les valeurs propres de la matrice de variance-covariance. Ces valeurs permettent d'évaluer la planarité effective du voisinage considéré, et donc la fiabilité des normales calculées. Pour être plan, un ensemble de points doit minimiser la plus petite des trois dispersions tout en maximisant les deux restantes. Un ratio de la plus petite dispersion sur le produit des deux plus grandes permet par exemple d'estimer une confiance (adimensionnelle) pour le vecteur normal calculé: plus ce ratio est faible, plus la normale calculée est fiable.

La problématique majeure de l'approche par ajustement de plans réside dans le choix du voisinage, qui détermine la qualité des normales estimées comme le soulignent N. Mitra et A. Nguyen [MN04]. Un voisinage trop restreint ne permet pas un ajustement robuste de plan, puisque l'impact du bruit du nuage n'est pas compensé par la taille de la zone étudiée. À l'inverse, le choix d'un voisinage trop grand permet effectivement d'atténuer l'effet du bruit, mais ne satisfait généralement pas l'hypothèse selon laquelle les voisinages étudiés sont assimilables à des plans. De plus, le choix d'une taille unique de voisinage pour le calcul de toutes les normales du nuage n'est pas un choix pertinent, puisque les surfaces étudiées présentent habituellement des courbures très variées. Les approches adaptatives permettant de choisir localement la taille la plus appropriée [MN04, DMDV11] sont donc préférables, bien que plus coûteuses en temps de calcul.

Il est de plus possible de pondérer les points du voisinage de  $\mathbf{p}$  dans l'expression de la matrice  $\Sigma(\mathcal{N}(\mathbf{p}))$  en fonction décroissante de leur distance à  $\mathbf{p}$  [ABCO<sup>+</sup>01]. On attribue ainsi aux points les plus éloignés une importance moindre dans l'évaluation du plan tangent. Cette approche possède toutefois l'inconvénient de privilégier les points les plus proches de  $\mathbf{p}$  indépendamment de leur pertinence réelle, au risque d'amplifier l'importance du bruit sur le calcul du vecteur normal.

Les normales calculées par ajustement de plans locaux souffrent d'un effet de lissage, inhérent au processus de "moyennage" induit par l'ajustement au sens des moindres carrés. Ainsi, le calcul de normales ne permet pas d'obtenir deux points voisins ayant des normales radicalement différentes: s'ils sont voisins, deux points partagent un voisinage commun significatif et possèdent donc des plans moyens très peu différents. Les plus fortes courbures sont les principales victimes de ce filtre passe-bas, dont l'importance grandit avec la taille des voisinages considérés.

Finalement, l'inconvénient principal des normales calculées vis-à-vis des normales fournies tient au fait que les vecteurs calculés ne sont pas orientés. Les calculs fournissent en effet la direction des vecteurs, sans en donner le sens. La détermination de l'orientation est un problème particulièrement complexe à résoudre. Les normales issues de l'acquisition quant à elles sont orientées: la position du scanner vis-à-vis du point acquis permet en effet de déterminer simplement le sens du vecteur normal.

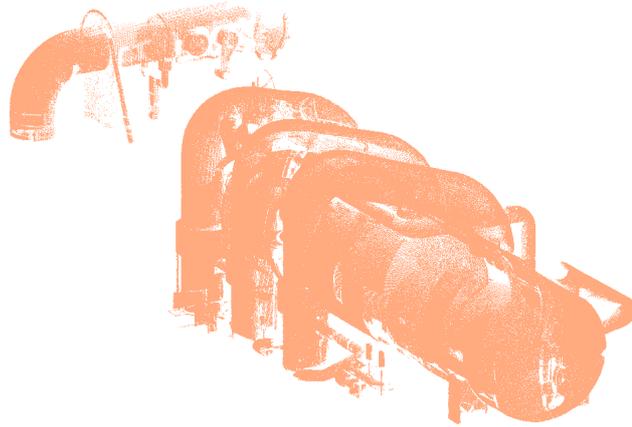
### 1.2.4 Défauts

#### Bruit

Comme tout processus de mesure, l'acquisition (laser ou photogrammétrique) est entachée d'erreurs. Il existe en effet un aléa lié à différents facteurs (conditions atmosphériques, nature de la surface échantillonnée, albédo etc.) induisant une erreur sur l'estimation de la distance séparant l'appareil d'acquisition de la surface étudiée. De plus, il peut exister un biais systématique, dû par exemple à une imprécision lors du calibrage de l'appareil, ou plus couramment à l'angle d'incidence entre le faisceau laser et la surface acquise. Sur un matériau diffus, plus cet angle est proche de  $90^\circ$ , plus l'acquisition est précise puisque l'empreinte lumineuse à la surface est plus concentrée.

Outre les incertitudes introduites par la physique du processus d'acquisition, il peut exister une autre source d'erreur systématique. Un nuage de points est en effet construit, dans la plupart des cas, comme un assemblage de relevés. Comme nous le disions en section 1.2.2, une acquisition seule ne permet généralement d'avoir qu'une très vue partielle des surfaces étudiées, du fait des occultations. Pour recueillir une quantité satisfaisante d'informations, il est donc souvent nécessaire d'agrèger les données issues de différents points de vue permettant une couverture optimale de la surface étudiée. Or le recalage des différents nuages partiels en un nuage complet est une source non négligeable d'erreurs. Le recalage nécessite en effet d'estimer avec la plus grande précision possible la position et l'orientation des appareils d'acquisition de manière à déterminer la position relative des nuages partiels. Si elles sont issues de mesures, les positions et orientations des appareils sont par nature erronées. Si elles sont issues de calculs, les méthodes de recalage existantes ne garantissent qu'une précision très limitée. Quoiqu'il en soit, le recalage des nuages partiels introduit une incertitude supplémentaire sur les données (erreur de consolidation - cf. figure 1.12c).

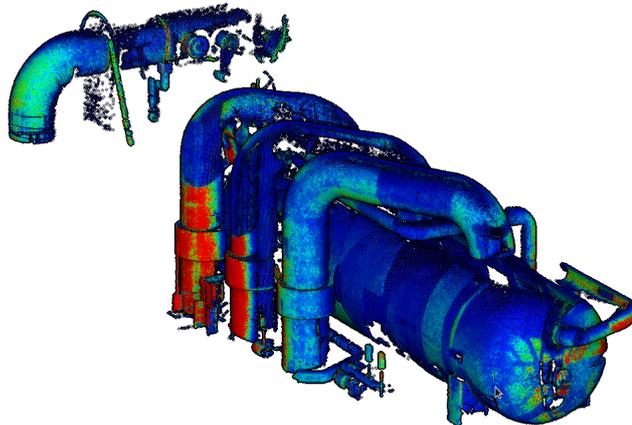
La position de chaque point d'un nuage est donc soumise à une incertitude dont la modélisation est très complexe, du fait du nombre et de la nature diverse de ses causes. Elle n'est de plus pas homogène sur le nuage de points: certaines portions sont soumises à des erreurs plus conséquentes que d'autres. Toutes ces erreurs constituent le bruit du nuage de points. Ce bruit est très souvent (quasi systématiquement) approximé dans la littérature par un modèle gaussien d'erreur sans réelle justification. Cette hypothèse est contestée par X. Sun et al. [SRML09] qui montrent qu'elle ne propose qu'une estimation grossière



(a) Nuage de points



(b) Éclairage calculé grâce aux normales estimées sur un voisinage de 40 points



(c) Incertitudes sur les plans locaux, estimées via la dispersion de long du plus petit axe de l'ACP. Les incertitudes sont représentées sur une échelle de couleur allant du bleu (fiable, car faible dispersion) au rouge (incertain car dispersion élevée).

**Figure 1.12:** Calcul de normales sur un nuage de points par ajustement de plans locaux (ACP pondérée par la distance au point). Ces normales peuvent être utilisées pour la visualisation des données grâce au calcul de réémission lumineuse en tout point du nuage, via le modèle lambertien. Les incertitudes obtenues sur le calcul des normales sont révélatrices du bruit dans le nuage de point, dont l'hétérogénéité s'explique ici vraisemblablement par un défaut de recalage, révélé par les contours nets dans les motifs d'incertitude.

de la réalité, puisque le bruit n'est en général pas gaussien. Toutefois, la représentation gaussienne constitue un modèle simple du bruit malgré son caractère très approximatif. Au bruit s'ajoutent de potentielles aberrations de mesure (que l'on nomme couramment *outliers*). Il s'agit de points qui n'échantillonnent manifestement aucune surface, qui sont erronés dans des proportions qui ne s'expliquent pas par le biais de l'incertitude sur la position des points (trop faible pour expliquer de telles aberrations).

Les normales, qu'elles soient fournies ou calculées, sont elles aussi soumises à un bruit qui dépend notamment de l'incertitude sur la position des points (cf. figure 1.12c).

### Densité

Le processus d'échantillonnage angulaire régulier de l'espace de visualisation des capteurs introduit une hétérogénéité significative dans la densité des données 3D. En effet, plus les surfaces relevées sont éloignées du capteur, plus les mesures acquises sont distantes les unes des autres. Les surfaces les plus proches du capteur présentent ainsi une densité d'échantillonnage élevée, tandis que les plus éloignées peuvent souffrir d'un sous-échantillonnage.

Cette hétérogénéité dans la densité des nuages de points 3D est généralement accentuée par le processus de recalage des différentes vues. En effet, les zones de recouvrement des relevés bénéficient d'un sur-échantillonnage, du fait de la multiplication des capteurs intervenant dans leur acquisition. On peut ainsi observer des sauts dans la densité locale d'un nuage de points, imputables à la fusion des relevés (cf. figure 1.11).

Pour finir, les phénomènes d'occultations lors de l'acquisition constituent la plus importante source de biais dans la densité des nuages de points. Il existe en effet des pans entiers de la scène qui ne peuvent apparaître dans les relevés 3D, malgré les efforts réalisés pour multiplier les vues, du fait du masquage de certaines parties de l'environnement relevé par d'autres parties de ce même environnement. Certains endroits sont par exemple inaccessibles aux capteurs, d'autres ne peuvent être relevés faute de temps en particulier en certains endroits de centrales nucléaires (limitation des temps d'arrêt de tranches, environnement irradiés, etc.), le géomètre topographe effectuant les relevés peut ne pas s'apercevoir lors de l'acquisition qu'une partie de la scène n'a pas été relevée, etc. Il est par exemple très courant de n'obtenir qu'une portion des tuyaux dans les relevés de lignes de tuyauterie le long de murs, puisque les surfaces faisant face aux murs ne peuvent être acquises (réciproquement, les parties du mur faisant face aux tuyaux sont inaccessibles).

Les défauts d'homogénéité dans la densité des données peuvent être gênants pour bon nombre de traitements sur les nuages de points. Cette diversité peut être "lissée" en échantillonnant le nuage de points de manière à uniformiser la distance séparant un point de ces plus proches voisins. Ce processus permet ainsi de diminuer la densité des zones suréchantillonnées en provoquant une perte de l'information redondante, mais ne permet en aucun cas d'ajouter une information initialement absente. Seuls les algorithmes de reconstruction permettent éventuellement de retrouver les données manquantes [PMG+05, SDK09].

## 1.3 Reconnaissance et ajustement de formes

Le problème de la reconnaissance de formes se pose comme la recherche de parties correspondant à des formes cibles parmi de grands volumes de données. Sa formulation peut être très variée selon l'application visée, notamment selon la nature des données traitées. Il

peut s'agir de la recherche d'une instance au sein d'une base de données qui soit similaire à une forme requête. Certains s'intéressent à la recherche de correspondances entre les parties de maillages surfaciques, basée sur l'étude géométrique de ces surfaces. D'autres étudient la segmentation, consistant en l'identification de formes distinctes au sein des données, qu'il s'agisse de nuages de points, de maillages ou même souvent d'images.

En tous les cas, la reconnaissance de formes nécessite la mise en place de mesures et outils permettant d'identifier les formes, de les ajuster et de valider la mise en correspondance. Ces outils dépendent bien entendu de la nature des données traitées.

### 1.3.1 Ajustement de surfaces

Pour reconnaître des objets dans un nuage de points, il est avant tout nécessaire de se donner des outils permettant de calculer une surface approximant un ensemble de points. Ce calcul se pose la plupart du temps comme un problème de minimisation d'une mesure de qualité d'ajustement: il s'agit de retrouver la surface minimisant l'écart aux données. Nous donnons en section 2.3.1 plusieurs types de mesures couramment utilisées pour quantifier la qualité d'ajustement d'une surface sur un nuage de points.

#### Quelques surfaces usuelles

La formulation du problème d'ajustement dépend de la nature de la surface étudiée. L'ajustement de carreaux de Bézier par exemple, consiste en la recherche des points de contrôle définissant la surface la plus proche algébriquement des données. Le problème se pose alors comme la résolution d'un système linéaire surdéterminé dont les inconnues sont les coordonnées des points de contrôle, et dont les coefficients sont déterminés en évaluant le polynôme de Bernstein aux points du nuage. La principale difficulté consiste en la détermination des coordonnées paramétriques associées aux points du nuage, pour le calcul des coefficients du système linéaire. Le système étant surdéterminé, il n'accepte en général pas de solution exacte satisfaisante. Toutefois, il peut être résolu au sens des moindres carrés à l'aide des méthodes habituelles d'algèbre linéaire (calcul de pseudo-inverse, décomposition matricielle, etc.), fournissant ainsi une solution optimale.

Les quadriques sont aussi couramment utilisées [WbFAR99, AS05, DLP08, FEF97] en reconnaissance de formes, puisque très adaptées à la description de surfaces manufacturées. Une quadrique est une surface définie algébriquement comme l'ensemble des points satisfaisant une équation polynomiale de second degré:

$$\underbrace{a_0x^2 + a_1y^2 + a_2z^2 + a_3xy + a_4xz + a_5yz}_{\text{forme quadratique}} + \underbrace{2(b_0x + b_1y + b_2z)}_{\text{forme linéaire}} + c = 0 \quad (1.3)$$

Soit sous forme matricielle (en posant  $\mathbf{x} = (x, y, z)^T$ ):

$$\mathbf{x}^T \cdot \underbrace{\begin{pmatrix} a_0 & \frac{a_3}{2} & \frac{a_4}{2} \\ \frac{a_3}{2} & a_1 & \frac{a_5}{2} \\ \frac{a_4}{2} & \frac{a_5}{2} & a_2 \end{pmatrix}}_{\mathbf{A}} \cdot \mathbf{x} + 2 \underbrace{\begin{pmatrix} b_0 \\ b_1 \\ b_2 \end{pmatrix}}_{\mathbf{b}} \cdot \mathbf{x} + c = 0 \quad (1.4)$$

Les cônes, cylindres et ellipsoïdes font par exemple partie de la famille des quadriques (les plans peuvent aussi être considérés comme des quadriques dégénérées, dont la forme quadratique correspond à la matrice nulle).

Pour déterminer les paramètres de la forme canonique associée à la quadrique, il faut mettre cette équation sous sa forme normalisée 1.5. La normalisation calcule le changement de repère approprié de sorte que l'équation de la quadrique dans le nouveau repère soit sous la forme :

$$s_0x^2 + s_1y^2 + s_2z^2 = s_3 \quad (1.5)$$

Si l'on considère uniquement les quadriques à centre de symétrie (dont font partie les cylindres, cônes et ellipsoïdes), le centrage de quadrique est l'opération consistant, par changement de variable, à supprimer la forme linéaire de l'équation 1.3. Le centre  $\mathbf{o}$  de la quadrique est tel que [Cai87]:

$$\mathbf{A} \cdot \mathbf{o} + \mathbf{b} = \mathbf{0}$$

Une fois  $\mathbf{o}$  connu, l'équation de la quadrique centrée se reformule :

$$\mathbf{x}'^T \cdot \mathbf{A} \cdot \mathbf{x}' = -c - \mathbf{b}^T \cdot \mathbf{o} \quad (1.6)$$

On vérifie aisément que la quadrique définie par 1.6 est centrée, puisque si le point  $\mathbf{p}$  appartient à la quadrique, alors  $-\mathbf{p}$  appartient lui aussi à la quadrique. On remarque ensuite que si  $\mathbf{A}$  est diagonale, alors l'équation est sous sa forme normalisée. Sinon, la diagonalisation de  $\mathbf{A}$  permet de trouver un repère (et la matrice de changement de repère associée) dans lequel cette matrice est diagonale. La décomposition en valeurs propres permet d'effectuer cette diagonalisation, et les vecteurs propres de la matrice  $\mathbf{A}$  constituent ainsi les axes principaux de la quadrique. Dans le repère défini par ces vecteurs (unitaires et orthogonaux deux à deux), l'équation de la quadrique est sous sa forme normalisée, et les valeurs propres fournissent les coefficients  $s_0$ ,  $s_1$  et  $s_2$  de l'équation 1.5. D'après l'équation 1.6, on a de plus  $s_3 = -c - \mathbf{b}^T \cdot \mathbf{o}$ .

En fonction des coefficients de la forme normalisée, on peut déduire la nature et les paramètres de la quadrique selon le tableau 1.1.

Équation	Description
$\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1$	Ellipsoïde de rayons $a$ , $b$ et $c$ . En cas d'égalité de deux de ces paramètres, il s'agit alors d'une sphéroïde, et en cas d'égalité des trois paramètres, il s'agit d'une sphère.
$\frac{x^2}{a^2} + \frac{y^2}{b^2} - \frac{z^2}{c^2} = 0$	Cône à base elliptique, aligné sur l'axe "Z". Lorsque $a = b$ , le cône est à base circulaire. Dans ce cas, l'angle d'ouverture $\alpha$ du cône est tel que $\tan(\alpha) = \frac{a}{c}$ .
$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$	Cylindre à base elliptique, aligné sur l'axe "Z". Lorsque $a = b$ , le cylindre est à base circulaire. Dans ce cas, le rayon du cylindre est $a$ .
$\frac{x^2}{a^2} - \frac{y^2}{b^2} = 1$	Cylindre hyperbolique, aligné sur l'axe "Z".
$\frac{x^2}{a^2} + \frac{y^2}{b^2} = z$	Paraboloïde elliptique, aligné sur l'axe "Z". Lorsque $a = b$ , le paraboloïde est circulaire.
$\frac{x^2}{a^2} - \frac{y^2}{b^2} = z$	Paraboloïde hyperbolique, aligné sur l'axe "Z"
$\frac{x^2}{a^2} + \frac{y^2}{b^2} - \frac{z^2}{c^2} = 1$	Hyperboloïde à une nappe, aligné sur l'axe "Z".
$\frac{x^2}{a^2} + \frac{y^2}{b^2} - \frac{z^2}{c^2} = -1$	Hyperboloïde à deux nappes, aligné sur l'axe "Z".

**Table 1.1:** Classification des quadriques en fonction de leur équation normalisée.

Le principe de l'ajustement algébrique de quadrique consiste en l'approximation des données par l'équation générale de la quadrique 1.3. En remplaçant  $x$ ,  $y$  et  $z$  par les coordonnées des points du nuage, on obtient un système linéaire surdéterminé. La résolution de ce système au sens des moindres carrés (sous contrainte, typiquement  $\sum a_i^2 + \sum b_j^2 + c^2 = 1$ ) permet de trouver la quadrique générale minimisant la distance algébrique aux points du nuage (cf. section 2.3.1). On identifie ensuite la nature de la quadrique puis ses paramètres caractéristiques en la mettant sous sa forme normalisée via le processus décrit ci-avant.

En pratique, ce procédé ne fonctionne pas vraiment puisqu'il est difficile d'établir la nature d'une quadrique ajustée, dans la mesure où les paramètres calculés numériquement ne permettent pas de rattacher l'équation normalisée à un des cas du tableau 1.1. Pour reconnaître un cylindre par exemple, il faut trouver  $s_2 = 0$  ce qui n'arrive quasiment jamais du fait du bruit des erreurs numériques d'une part, et parce que la surface approximant au mieux les données n'est pas nécessairement un cylindre parfait d'autre part.

Il est plus fiable d'ajuster directement la forme appropriée en considérant les primitives géométriques paramétrées. Il faut alors tester toutes les formes recherchées (cylindre, tore, etc.) pour ne finalement garder que la plus pertinente. L'ajustement géométrique au sens des moindres carrés [Ahm08] se pose alors comme un problème d'optimisation non linéaire (sauf pour le plan). Il s'agit de trouver le paramétrage de la primitive qui minimise la moyenne des distances quadratiques séparant les points de la surface de la primitive. Il existe divers algorithmes [PTVF92] permettant de résoudre ce type de problème (descente de gradients, gradients conjugués, etc.). T. Chaperon [Cha02] préconise par exemple d'utiliser l'algorithme de Levenberg-Marquardt, et propose un paramétrage ainsi qu'une formulation des distances et dérivées permettant de réaliser l'ajustement de sphères, cylindres, cônes et tores. L'initialisation de l'algorithme constitue la principale difficulté de cette approche, puisque les algorithmes d'optimisation non linéaire ne garantissent généralement la convergence vers l'optimum que lorsqu'ils sont initialisés à partir d'une solution suffisamment proche de l'optimum recherché. Ce problème est abordé dans [MLM01, Cha02] notamment.

## Validation de la surface

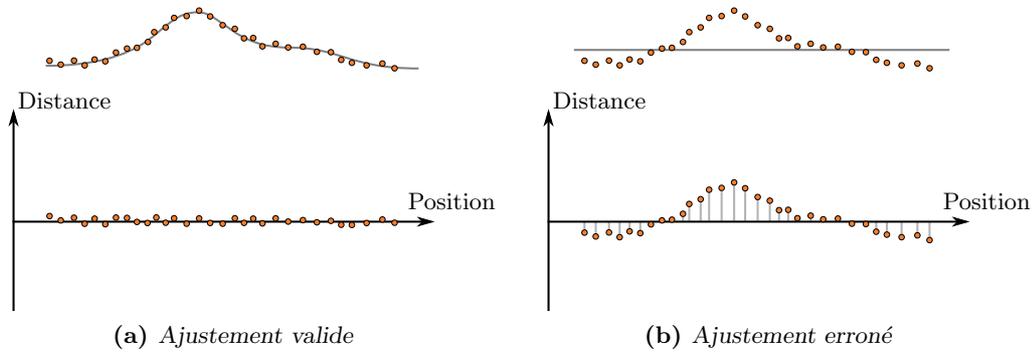
Une fois la surface ajustée, il est important de pouvoir confirmer la pertinence du modèle 3D proposé pour décrire le nuage de points. Il peut aussi s'agir de comparer l'ajustement de diverses surfaces sur un même nuage, de manière à déterminer laquelle de ces surfaces est la plus apte à décrire le nuage de points. Quoiqu'il en soit, il est nécessaire de se donner un critère permettant de discriminer les erreurs d'ajustement.

L'erreur quadratique moyenne (cf. section 2.3.1) est parfois utilisée pour permettre la comparaison de l'ajustement de surfaces. Cette approche n'est pas forcément pertinente, puisque toutes les surfaces n'ont pas la même capacité à minimiser l'erreur vis-à-vis des données, et les surfaces les plus simples sont ainsi pénalisées alors qu'elles sont souvent les plus pertinentes. La prise en compte de la "complexité" des surfaces est donc souhaitable dans ce type de raisonnement (nous reviendrons sur cette problématique par la suite pour le cas des primitives géométriques - cf. section 1.3.2, figure 1.21).

T. Chaperon [Cha02] propose une étude plus poussée de ce problème, en considérant deux situations distinctes, selon qu'un modèle de bruit du nuage de points est disponible ou non.

Si le modèle de bruit est connu, une analyse statistique permet de vérifier que la

distribution des résidus (distances entre les points du nuage et la surface) est conforme aux attentes portées par le modèle de bruit, via un test d'adéquation entre la distribution constatée pour les résidus et la distribution attendue. Cette démarche de validation statistique par un modèle de bruit se retrouve aussi dans les travaux de D. Girardeau-Montaut [GM06].



**Figure 1.13:** Analyse de la pertinence de l'ajustement par détection de tendances [Cha02]. En repérant un motif parmi les résidus dans la figure de droite, on peut conclure que les points ne sont pas aléatoirement répartis autour de la surface ajustée. Le bruit ne peut expliquer ce déterminisme dans les erreurs constatées, permettant ainsi de conclure que l'ajustement est faux.

Lorsque le modèle de bruit n'est pas disponible, T. Chaperon propose d'analyser les tendances parmi les résidus, et en particulier l'indépendance des distances "point-surface" vis-à-vis de la position des points. En effet, si la surface ajustée représente effectivement le nuage de points, on doit alors constater une répartition aléatoire des distances entre les points et la surface. La présence d'une dépendance entre la distance et la position des points sur la surface trahit une régularité dans l'erreur entre le modèle 3D et les données observées, permettant ainsi de conclure que la surface n'est pas appropriée pour décrire le nuage de points. L'hypothèse de l'ajustement valide peut être ainsi être rejetée si l'on constate que la surface polynomiale s'ajustant au mieux sur les résidus n'est pas l'hyperplan des distances nulles.

### Moindres Carrés Glissants

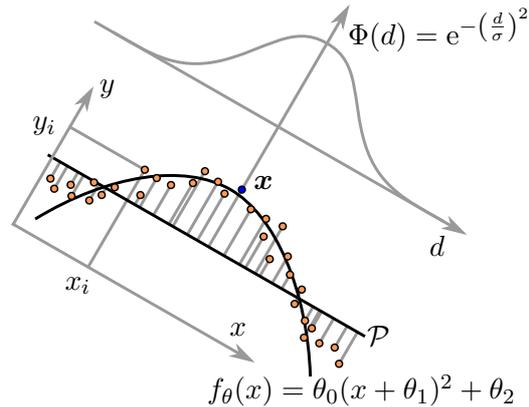
Les moindres carrés glissants (*Moving Least Squares* - MLS) constituent une méthode d'ajustement de surfaces continues basée sur une étude locale du nuage de points. Dans une fenêtre de taille donnée, le nuage est ainsi approximé par une surface polynomiale de faible degré. Cet ajustement au sens des moindres carrés est pondéré en fonction de l'éloignement au point étudié, de sorte que le poids des points du voisinage considéré pour l'ajustement diminue avec leur distance au point d'intérêt [Lev03].

Soit  $\{(\mathbf{x}_0, y_0), \dots, (\mathbf{x}_{n-1}, y_{n-1})\}$  un ensemble de mesures telles que  $\mathbf{x}_i \in \mathbb{R}^d$  et  $y_i \in \mathbb{R} \forall i \in [0, n[$ , et  $\mathbf{x}$  le point d'intérêt autour duquel on souhaite évaluer la surface. On se donne une fonction paramétrée  $f_\theta : \mathbb{R}^d \rightarrow \mathbb{R}$  destinée à approximer les mesures, et  $\Phi$  une fonction de pondération positive décroissante (typiquement une fonction gaussienne ou une simple exponentielle inverse). L'ajustement local des MLS consiste en la recherche du paramétrage  $\theta^*$  de  $f_\theta$  minimisant l'erreur quadratique pondérée:

$$\theta^* = \operatorname{argmin}_{\theta} \sum_{i=0}^{n-1} \underbrace{(f_\theta(\mathbf{x}_i) - y_i)^2}_{\text{Erreur quadratique}} \times \underbrace{\Phi(\|\mathbf{x} - \mathbf{x}_i\|)}_{\text{Pondération}} \quad (1.7)$$

Le fait que  $\Phi$  s'annule au delà d'une certaine valeur permet en particulier d'exclure du calcul tous les points trop éloignés de  $\mathbf{x}$ , et définit ainsi la taille du voisinage considéré dans les MLS.

Dans le cas de données 3D, cet ajustement peut se faire en deux temps: on estime tout d'abord le plan approximant localement le voisinage de  $\mathbf{x}$ . L'ajustement polynomial est ensuite effectué en considérant les projections  $x_i$  des points dans ce plan, et  $y_i$  hauteur des points vis à vis du plan. Ces deux approximations (le plan et la surface polynomiale) se formulent comme des ajustements au sens des moindres carrés pondérés.



**Figure 1.14:** Ajustement par moindres carrés pondérés glissants. Un premier ajustement permet d'approximer les données par un plan  $\mathcal{P}$  définissant une base dans laquelle sont exprimées les mesures  $\{(x_i, y_i)\}_i$ . Sur cet exemple, on ajuste une parabole sur les mesures. La fonction  $\Phi$  permet d'amoinrir l'influence des points les plus éloignés sur l'ajustement.

Ce processus est par exemple utilisé pour "lisser" le bruit du nuage de points [WPH<sup>+</sup>04, ABCO<sup>+</sup>01], en projetant chaque point du nuage sur la surface polynomiale ajustée localement par les MLS à partir de ce point.

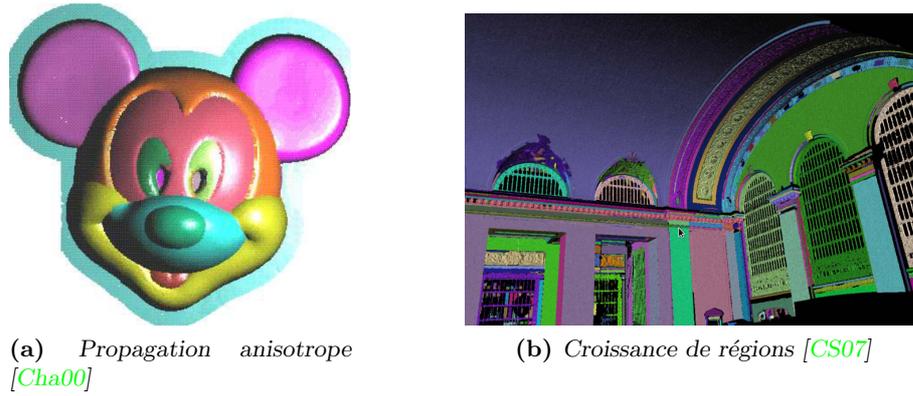
### 1.3.2 Segmentation

La segmentation est la tâche consistant en l'identification et la séparation, au sein de grands volumes de données, des parties possédant une caractéristique commune. Elle repose donc avant tout sur la mise en place d'un critère de segmentation, définissant la règle de regroupement des données. On se donne ainsi un prédicat  $C(\mathbb{A})$  vrai si et seulement si le sous-ensemble  $\mathbb{A}$  (on parlera de "segment" extrait des données) satisfait le critère de segmentation. Dans le cas du traitement d'images par exemple, on utilise typiquement l'uniformité en terme de teinte ou de luminosité des pixels composant un segment donné. Ainsi, pour une image en niveau de gris, on utilisera un prédicat  $C$  basé sur la répartition statistique des luminances associées aux pixels de  $\mathbb{A}$ :  $C(\mathbb{A})$  est vrai si et seulement si la dispersion (l'écart type statistique) des valeurs de luminances dans  $\mathbb{A}$  est inférieure à un seuil spécifié.

## Fondements et principaux algorithmes pour la segmentation d'images

Segmenter un ensemble de données  $\mathbb{D}$  sur la base d'un critère représenté par le prédicat  $C$ , c'est identifier la partition  $\{\mathbb{D}_0, \dots, \mathbb{D}_{s-1}\}$  telle que:

- par définition de la partition:



**Figure 1.15:** Quelques exemples de segmentation sur images de profondeur. Chaque couleur représente un segment.

$$\begin{aligned}
 & - \mathbb{D}_i \subset \mathbb{D} \quad \forall i \\
 & - \mathbb{D}_i \cap \mathbb{D}_j = \emptyset \quad \forall i, j \in [0, s[ \quad i \neq j \\
 & - \bigcup_{i \in [0, s[} \mathbb{D}_i = \mathbb{D}
 \end{aligned}$$

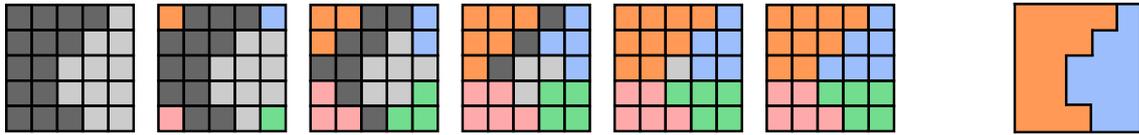
- le prédicat  $C(\mathbb{D}_i)$  est vrai pour tout segment  $\mathbb{D}_i$ , sauf un éventuellement. Le seul segment faisant exception constitue l'ensemble des données non segmentées.
- la segmentation est maximale, c'est-à-dire que la fusion de segments produit une partition qui ne satisfait plus le critère de segmentation. Formellement, le prédicat  $C(\mathbb{D}_i \cup \mathbb{D}_j)$  est faux pour tout couple  $(i \neq j)$  pris dans  $[0, s[$ .
- en général, on ajoute de plus une contrainte de connexité sur tous les segments (à l'exception des données non segmentées), reposant sur la définition d'un voisinage sur  $\mathbb{D}$ . Un segment  $\mathbb{D}_i$  est connexe si, pour toute paire d'éléments  $(x, y)$  pris dans  $\mathbb{D}_i$ , il existe un chemin de voisin en voisin qui relie  $x$  à  $y$ .

Cette définition se retrouve par exemple dans les travaux de A. Hoover et al. [HBJJ+96] traitant des images de profondeur.

La thématique de la segmentation trouve ses origines et ses plus nombreuses applications dans le domaine du traitement d'images, où il existe une multitude d'approches permettant d'aborder ce problème. Notons d'ores et déjà que la nature de grille des images permet de définir simplement un voisinage entre pixels, basé sur une 4-connexité (prise en compte des voisinages horizontaux et verticaux uniquement) ou une 8-connexité (4-connexité avec prise en compte des voisinages diagonaux), permettant d'établir aisément la connexité d'ensembles de pixels.

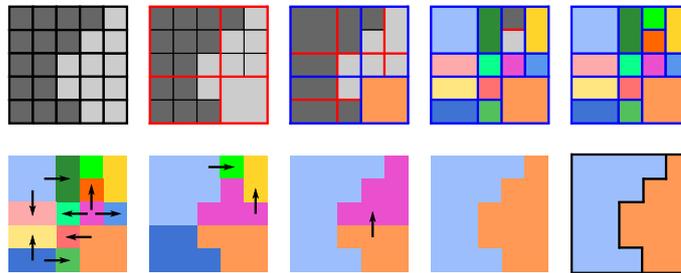
Une des approches les plus connues et plus utilisées est l'algorithme par croissance de régions (*region growing*). Il procède en dispersant initialement une certaine quantité de graines parmi les données à segmenter, selon une méthode à déterminer (en choisissant aléatoirement des données typiquement). Ces graines constituent des segments atomiques, qui satisfont le prédicat de segmentation  $C$  et que l'algorithme va ensuite faire grossir. La croissance d'une région procède itérativement, en considérant les éléments qui lui sont adjacents: un nouvel élément non encore affecté est intégré à la région si cette opération

conserve l'état de vérité de son prédicat de segmentation  $C$ . Cette opération de croissance est répétée pour toutes les régions, jusqu'à ce que plus aucune région ne puisse grossir. Si le nombre de graines initiales est suffisant, tous les pixels doivent alors être affectés à une région, mais il se peut que les données soient sursegmentées (le nombre de régions est plus important que le nombre de segments effectivement recherchés). La croissance de régions est donc systématiquement suivie d'une étape de fusion des régions adjacentes: deux régions sont réunies si cette opération permet de satisfaire le prédicat  $C$  pour la région résultante. La fusion est ensuite répétée jusqu'à ce qu'elle ne soit plus possible.



**Figure 1.16:** Segmentation d'image par croissance de régions. Le critère utilisé ici est l'uniformité parfaite des segments. La seconde vignette présente un choix de placement des graines (une dans chaque angle), tandis que les vignettes suivantes illustrent la croissance. La dernière vignette montre le résultat après fusion des régions adjacentes similaires.

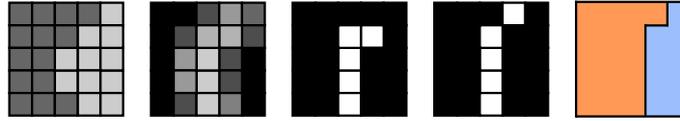
L'approche division-fusion (*split and merge*) procède selon des principes similaires. Dans un premier temps, la phase de division considère la totalité des données comme un unique segment, qui sera subdivisé récursivement jusqu'à ce que tous les segments résultants satisfassent le prédicat  $C$ . La subdivision s'appuie généralement sur une structure de découpage spatial tel qu'un *quadtree* (cf. annexe C.1.1). Cette division résulte en une sursegmentation des données, qui doit être corrigée par une phase de fusion telle que décrite précédemment.



**Figure 1.17:** Segmentation d'image par division-fusion. La première rangée illustre la phase de division par quadtree, tandis que la seconde présente la fusion. La subdivision d'une cellule de quadtree (rouge et bleu) est interrompue dès lors que les données contenues dans la cellule satisfont le critère de segmentation (ici, l'uniformité parfaite).

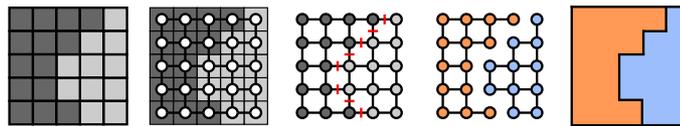
L'approche par détection de contours considère que les segments homogènes sont délimités par des frontières séparant nettement toute région de ses voisines. L'identification de frontières closes permet ainsi de déduire les régions qu'elle enferment. Cette détection peut être calculée en repérant les ruptures dans l'homogénéité des données vis-à-vis du critère de segmentation. Dans le cas du traitement d'images, une telle rupture correspond à une grande amplitude du gradient de la fonction d'intensité des pixels. Les gradients peuvent être calculés par convolution avec un filtre de Canny ou Sobel, fournissant une image dont l'intensité correspond à la magnitude des gradients de l'image d'origine. Par seuillage (éventuellement par hystérésis) de l'image de gradient obtenue, on peut ensuite isoler les pixels de forte magnitude correspondant aux contours recherchés. Généralement,

les contours obtenus sont incomplets et épais, nécessitant un post-traitement pour clore les contours d'épaisseur unitaire, et ainsi isoler les régions.



**Figure 1.18:** Segmentation d'image par détection de contours. La seconde vignette présente le résultat du calcul de gradients (filtre de Sobel), la troisième illustre le seuillage de l'image de gradients. La vignette suivante présente le résultat de la clôture du contour, avec amincissement (chaque pixel possède au plus 2 voisins en 8-connexité).

L'approche par coupe de graphe (*graph cut*) pose la segmentation comme un problème d'optimisation. L'image est représentée sous forme d'un graphe dont les sommets correspondent aux pixels, et les arêtes représentent l'adjacence des pixels. Les arêtes sont pondérées par une mesure de similarité entre les pixels. L'algorithme calcule ensuite les suppressions d'arêtes permettant de maximiser la similarité intra-classe, tout en maximisant la dispersion inter-classes. Ces "coupes" induisent la séparation du graphe initial en sous-graphes connexes correspondant aux segments recherchés.



**Figure 1.19:** Segmentation d'image par coupe de graphe. La seconde vignette présente le graphe associé à l'image. Dans la troisième vignette, l'algorithme sélectionne les arêtes dont la suppression optimise la fonction objectif. Ces coupes permettent d'isoler les segments distincts (dernières vignettes).

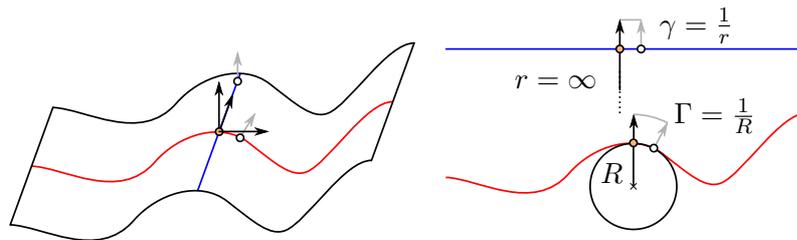
Il existe de nombreuses autres approches (lignes de partage des eaux, *clustering*, segmentation bayésienne, etc.) que nous ne détaillerons pas ici.

### Application à la segmentation de données 3D

Certaines des idées apparues dans la communauté du traitement d'images peuvent trouver une application dans le domaine de l'analyse des données 3D. Tout comme pour la segmentation d'images, il est avant tout nécessaire d'identifier le critère de segmentation définissant les caractéristiques selon laquelle on souhaite regrouper les données. Une fois, ce critère établi, il faut ensuite mettre en place l'algorithme de segmentation qui doit s'adapter à la nature des données.

**Critères de segmentation 3D** La plupart des travaux considèrent qu'une région homogène en 3D est une région représentant une surface "lisse", c'est-à-dire une 2-variété dans  $\mathbb{R}^3$  différentiable en tout point. Ce critère peut éventuellement être nuancé en étudiant la courbure de la surface [Ben00].

La courbure d'une surface est une propriété différentielle de second ordre et correspond intuitivement à l'amplitude de l'évolution locale du vecteur normal à la surface lorsqu'on la parcourt en "ligne droite" dans une direction donnée. A déplacement infinitésimal constant sur la surface, plus la normale se trouve modifiée lors du trajet, plus la surface est courbée. Ainsi un plan possède-t-il une courbure nulle (le plan tangent à un plan est invariant, quel que soit le point considéré), tandis qu'une sphère possède une courbure constante (où que



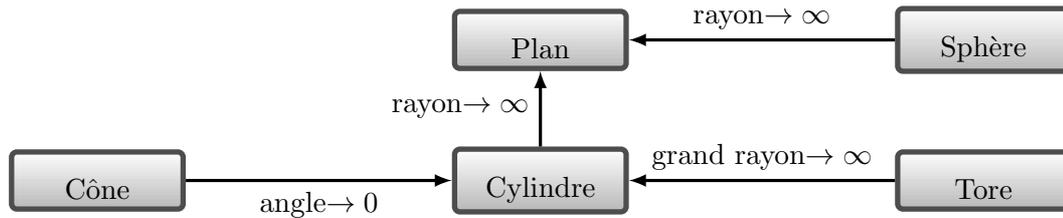
**Figure 1.20:** Courbures principales en un point d'une surface. La courbe bleue sur la surface engendre la courbure minimale au point étudié: la surface est plate dans cette direction, possédant ainsi un rayon de courbure  $r$  infini et par conséquent une courbure  $\gamma$  nulle (la normale -en gris- est constante). La courbe rouge possède un faible rayon de courbure  $R$ , et correspond à la courbure  $\Gamma$  maximale (grandes variations de la normale).

l'on soit sur la sphère, un même déplacement produira la même variation quantitative du vecteur normal). Quantitativement, la courbure dans une direction tangente à la surface s'interprète comme l'inverse du rayon du plus grand cercle tangent à la courbe décrite par la surface dans cette direction (cercle osculateur). Cette quantité peut être positive ou négative selon que le cercle osculateur se trouve "en dessous" ou "au-dessus" de la courbe, le sens étant dicté par l'orientation des vecteurs normaux à la surface. La courbure varie selon la direction de déplacement considérée. On distingue en particulier deux directions orthogonales produisant des courbures extrémales, que l'on appelle courbures principales. Bien qu'elle semble pertinente, l'information de courbure est difficilement utilisable en pratique puisque son calcul parmi des données bruitées et de densité hétérogène est en général très peu fiable (instabilité du calcul d'une différentielle d'ordre 2). D'autant plus que le signe de la courbure sur un nuage de points dont les normales ne sont pas orientées est indéterminé.

L'information de courbure est utilisée par R. Bénéière [B12] par exemple pour segmenter les surfaces de primitives distinctes sur un maillage surfacique.

Une quantité considérable de travaux de segmentation 3D [RvV06, Cha00, WKWL02, YY08] reposent sur les normales aux sommets, qui servent à définir l'homogénéité des régions. Un segment se définit ainsi comme un ensemble de points dont les normales vérifient un critère d'uniformité. Pour segmenter les régions planes, il s'agit de s'assurer que toutes les normales du segment sont similaires, à une tolérance près. En revanche, un segment lisse correspond à un ensemble de points tels que les points voisins possèdent une normale semblable (encore une fois, à une tolérance près). Un segment plan est ainsi représenté par une normale unique, alors qu'un segment lisse est potentiellement représenté par une multitude de normales, seule compte la stabilité locale de la normale. Ces normales peuvent être couplées à d'autres informations, comme l'intensité du retour laser [CS07].

D'autres descripteurs locaux peuvent être utilisés pour la segmentation. P.J. Besl et R. Jain [BJ88] proposent par exemple de calculer une première segmentation d'image de profondeur basée sur le signe de la courbure de la surface échantillonnée. N. Gelfand et J. Guibas [GG04] proposent un descripteur astucieux basé sur l'étude de l'autosimilarité de la surface selon certaines transformations. Une sphère est par exemple invariante par rotation autour de tout axe, tandis qu'un cylindre est invariant par translation le long de son axe, ou rotation autour de son axe. Les auteurs proposent un calcul de signature permettant d'identifier les comportements locaux de la surface vis-à-vis des diverses transformations. Les points sont ensuite segmentés sur la base de l'homogénéité de cette signature.



**Figure 1.21:** Diagramme de complexité des formes. Le plan est la forme la plus simple. Une surface plane peut être approximée par une portion de sphère infiniment grande, ou d'un cylindre infiniment grand. Une surface cylindrique peut elle-même être approximée par un cône d'angle quasi nul, ou un tore de grand rayon infini.

Une autre catégorie de critère de segmentation, certainement la plus intéressante dans le cadre de nos travaux, procède par reconnaissance de surfaces distinctes. La reconnaissance implique l'identification des surfaces, notamment leur nature ainsi que les paramètres qui caractérisent l'instance dérivant au mieux le segment considéré. La segmentation par reconnaissance de formes procède donc généralement par ajustement de surfaces. Il peut s'agir de la reconnaissance d'un unique type de primitive géométrique, par exemple de plans dans les applications de traitement de données d'acquisition en environnement urbain [BLGTK07, TKLG07]. Certains auteurs mêlent diverses primitives surfaciques [SWK07, MLM01, YY08, AF06], posant alors le problème de la distinction de la nature des surfaces. Ce problème est relativement complexe, dans le sens où il est initialement ambigu. Un ensemble de points ne décrit effectivement qu'une portion de la surface qu'il échantillonne. Il existe ainsi une multitude de surfaces décrivant cet ensemble, puisque les points ne contraignent qu'une partie de l'espace. La partie non échantillonnée est libre de toute contrainte et doit être "devinée", ce qui rend pertinente toute surface exprimée hors de l'ensemble des points acquis. Pour soulever cette ambiguïté, D. Marshall et al. [MLM01] préconisent de procéder par ordre de complexité de l'objet ajusté, la complexité étant définie par le nombre de paramètres définissant l'objet. On privilégiera ainsi l'ajustement d'un plan (3 paramètres) à une sphère (4 paramètres). Viennent ensuite, par ordre de complexité, le cylindre (5 paramètres), le cône (6 paramètres) et le tore (7 paramètres). L'idée de privilégier les formes les plus simples se retrouve dans d'autres travaux. L'approche concurrente consiste en la validation des formes en fonction de leur qualité d'ajustement sur le nuage de points. Or cette démarche tend à privilégier les formes les plus complexes puisqu'elles permettent un meilleur ajustement aux défauts de la surface.

**Algorithmes de segmentation de surfaces 3D** L'algorithme par croissance de région bénéficie d'un succès indéniable dans le domaine de la segmentation de données 3D [Gor07, MLM01, GG04, Tau91, CS07, RvV06, BJ88]. L'algorithme peut être appliqué directement dans les images de profondeur. Sur les maillages surfaciques, le voisinage est défini par la présence même d'arêtes reliant les sommets, et l'algorithme par croissance de région peut donc être appliqué directement. En revanche, son application aux nuages de points nécessite la mise en place d'un prédicat de voisinage. Deux approches sont principalement utilisées:

**$k$  plus proches voisins** ( $k$  Nearest Neighbors): chaque point est considéré comme étant un voisin des  $k$  points les plus proches dans le nuage de points.

**rayon limite** (**R**adius **B**ounded **N**earest **N**eighbors): deux points du nuage sont voisins si la distance qui les sépare est inférieure à un seuil  $r$ .

On note que le second voisinage (RBNN) est symétrique, contrairement au premier ( $k$ NN): le fait qu'un point  $\mathbf{p}$  fasse partie des  $k$  plus proches voisins d'un point  $\mathbf{q}$  ne garantit en aucun cas l'appartenance de  $\mathbf{q}$  aux  $k$  plus proches voisins de  $\mathbf{p}$ . On peut corriger ceci en considérant que deux points sont voisins si et seulement si la relation de "k-proximité" est réciproque. Ce critère est en particulier très utile pour permettre l'identification d'*outliers* [WPH<sup>+</sup>04], puisque ceux-ci se réclament voisins d'un grand nombre de points qui ne valident pas la réciprocity de la relation de voisinage.

Une fois le voisinage défini (en particulier lorsque les paramètres  $k$  ou  $r$  ont été spécifiés, selon la méthode choisie), la croissance de région peut être exécutée sur le nuage de points.

Certains auteurs s'inspirent d'autres approches issues de la segmentation d'images. Les contours détectés dans les images de profondeur [SD01, Ben00] révèlent par exemple les sauts de profondeur trahissant les changements de surface. L'approche de division-fusion trouve elle aussi son application dans le traitement de données 3D, en remplaçant le *quadtrees* par un *octree* [WKWL02]. En arrêtant la subdivision des cellules lorsque la dispersion des normales qu'elles contiennent est suffisamment faible, on isole ainsi les parties planaires qui pourront être fusionnées ensuite. Entretemps, la suppression des plus petites cellules de l'*octree* aura permis d'écarter les points représentant des courbures trop importantes (identifiés comme représentant des arêtes vives). P. Benko et al. [BMV01] proposent eux aussi une approche basée sur la division de régions, mais s'appuient sur l'analyse des formes correspondant aux régions pour guider la subdivision.

L. Yu et X. Youlun [YY08] proposent une approche particulièrement intéressante, propre au traitement de données 3D. Leur analyse repose sur l'application de Gauss du nuage de points. L'application de Gauss associe à tout point de la surface le vecteur unitaire normal à la surface en ce point. L'image d'une surface par cette application est donc incluse dans la sphère unitaire centrée  $\mathbb{S}^2$ . En particulier:

**plan** l'image d'un plan par l'application de Gauss est un point de  $\mathbb{S}^2$ ,

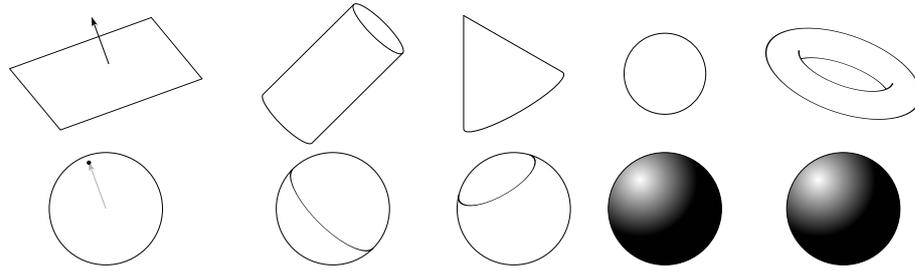
**cylindre/cône** l'image d'un cylindre ou d'un cône par l'application de Gauss est un cercle correspondant à l'intersection de  $\mathbb{S}^2$  et d'un plan orthogonal à l'axe de révolution de la forme. La hauteur du plan de coupe est fonction (sinus) de l'angle d'ouverture du cône (un cylindre étant assimilable à un cône d'angle nul).

**tore/sphère** l'image d'un tore ou d'une sphère par l'application de Gauss est  $\mathbb{S}^2$  toute entière.

L'étude de la trace laissée par un ensemble de points sur la sphère de Gauss permet de déduire la nature de la surface échantillonnée. Pour segmenter les données, les auteurs proposent ainsi d'isoler les différentes traces identifiables sur la sphère de Gauss<sup>7</sup> associée au nuage de points (par un algorithme de *clustering* sur  $\mathbb{S}^2$  de type *Mean-Shift*), permettant ainsi une première séparation des données selon leurs caractéristiques différentielles. Or il est fortement probable que deux surfaces distinctes possèdent des images par l'application de Gauss qui se recouvrent. Un second regroupement, parmi les points du nuage cette fois-ci, permet alors de segmenter correctement les surfaces selon le type des primitives

<sup>7</sup>La sphère de Gauss est le nuage de points obtenu en considérant les normales au points du nuage étudié. Ce "nuage de normales" est défini sur la surface  $\mathbb{S}^2$ .

qu'elles décrivent, sans n'avoir jamais effectué un ajustement de surface sur le nuage de points.



**Figure 1.22:** Sphères gaussiennes associées aux primitives géométriques usuelles. La rangée supérieure illustre les différentes primitives géométriques considérées sur cet exemple, tandis que la rangée inférieure illustre (en noir) l'image dans  $\mathbb{S}^2$  de chaque surface par l'application de Gauss.

L'analyse de la sphère de Gauss associée à un nuage de points s'avère extrêmement utile en pratique, puisqu'elle permet d'identifier de manière relativement fiable un certain nombre de caractéristiques de la surface étudiée: calcul de l'axe d'un cylindre, de l'angle d'ouverture d'un cône, identification de la surface, etc. T. Chaperon et F. Goulette [CG01] proposent par exemple une méthode d'extraction de cylindres basée sur la détection de plans dans la sphère de gaussienne du nuage de points.

### Discussion

Un certain nombre d'auteurs perçoivent la segmentation comme une étape préalable à la modélisation [WKWL02, CS07, BMV01]. L'identification des groupes de points cohérents vis-à-vis d'un critère de segmentation permet d'isoler plus ou moins grossièrement les sous-ensembles décrivant des surfaces considérées comme distinctes, et pourront par la suite faire l'objet d'analyses plus poussées (ajustement de surface, reconnaissance d'objets, etc.).

La méthode se prête effectivement au prétraitement des nuages de points si le critère de segmentation est relativement simple. En revanche, la démarche est différente lorsque la segmentation traite de tâches complexes telles que l'extraction de formes. La segmentation peut alors ne pas être réellement le problème que l'on cherche à résoudre, mais plutôt le résultat du processus de modélisation du nuage de points. Dans ce contexte, modélisation et segmentation sont intriquées, et la résolution d'un problème ne peut être menée à bien qu'en traitant l'autre comme le remarquent notamment T. Varady et al. [VMC97]. En effet, pour reconnaître une forme (typiquement par ajustement), il est nécessaire de considérer un sous-ensemble du nuage qui décrive effectivement cette forme. Or pour identifier un tel sous-ensemble, il faut savoir reconnaître la forme recherchée parmi les points. Ce cercle vicieux est la principale source de complexité de la tâche de l'extraction de formes dans les nuages de points.

Deux démarches se distinguent dans le domaine de la détection de formes:

**l'approche ascendante** (*bottom-up*), orientée segmentation, procède en considérant tout d'abord les portions locales dans le nuage de points pour tenter d'y reconnaître des formes. Ce processus est ensuite suivi d'éventuelles fusions. La plupart des approches présentées précédemment (en particulier celle basées sur les algorithmes par croissance de régions) font partie de cette catégorie.

**l'approche descendante** (*top-down*), orientée modélisation, considère la totalité du nuage pour en extraire des formes. L'identification des parties distinctes dans le nuage

de points est alors le résultat de la reconnaissance des objets.

L'approche ascendante souffre d'un certain nombre de lacunes inhérentes au processus de traitement local du nuage de points. En effet, l'ajustement de formes ne peut être fiable que si l'échelle d'étude est parfaitement adaptée. Le fait de considérer des voisinages trop étroits ne permet pas d'être précis. Au contraire, l'étude de voisinages trop larges introduit la présence de points ne correspondant pas à la forme recherchée, qui faussent l'ajustement. Les approches descendantes, telles que l'algorithme RANSAC ou les transformées de Hough (présentés par la suite), s'affranchissent de ce problème, et sont reconnues pour leur fiabilité et leur robustesse. En ce sens, nous pensons que les approches descendantes sont plus pertinentes pour la résolution des problèmes d'extraction de formes.

### 1.3.3 Détection de primitives géométriques

La reconnaissance de primitives géométriques telles que le plan, le cylindre, le cône, la sphère ou le tore au sein d'un nuage de points sans segmentation préalable trouve sa principale application dans le domaine de la rétro-ingénierie de nuage de points représentant des objets manufacturés, voire parfois même de scène architecturales ou de pièces archéologiques. Les approches plus connues traitant de ce problème sont certainement l'algorithme RANSAC et les Transformées Généralisées de Hough.

#### RANdom SAMple Consensus (RANSAC)

**Principe** L'algorithme du RANSAC, initialement proposé par M. Fishler et R. Bolles [FB87], propose une méthode stochastique permettant de calculer un modèle (au sens générique) expliquant les données observées. La raison d'être de cet algorithme réside dans sa robustesse aux *outliers*, qui sont une gêne considérable pour les approches de modélisation au sens des moindres carrés. Le RANSAC est un outil permettant d'identifier un modèle correspondant aux seules données pertinentes, sans tenir compte des *outliers*, sous l'hypothèse que les observations pertinentes sont suffisamment nombreuses pour permettre cette opération. Cet algorithme n'est pas seulement applicable à la détection de formes, et trouve ses principales applications en recalage d'images ou de données 3D (c'est d'ailleurs l'exemple développé par les auteurs de l'algorithme [FB87]).

Avant tout, on suppose qu'une instance de modèle peut être construite à partir d'un quorum de  $k$  données. Ce quorum contraint les paramètres libres du modèle, permettant ainsi de définir l'unique instance qui décrit ces  $k$  données observées. Pour le cas de la reconnaissance de primitives géométriques, C. Beder et W. Forstner [BF06] expliquent par exemple comment construire un cylindre à partir de 5 points, bien qu'il soit plus simple d'utiliser 6 ou 7 points.

Lors de chaque tentative, l'algorithme du RANSAC construit donc un candidat à partir de  $k$  données distinctes choisies aléatoirement. La pertinence de chaque candidat est ensuite évaluée à l'aide d'un score simple comptabilisant, parmi la totalité des observations, les données qui décrivent effectivement le candidat construit (que les auteurs nomment "consensus"). L'algorithme du RANSAC original [FB87] considère qu'une donnée décrit une instance du modèle si l'erreur observée est inférieure à un seuil  $\epsilon$  spécifié. Dans le cas de la reconnaissance de forme, on utilise typiquement l'erreur géométrique, c'est-à-dire la distance euclidienne entre un point et la forme candidate, de sorte que chaque primitive générée soit associée à l'ensemble des points du nuage dont la distance à la forme est inférieure au seuil d'erreur.

Ces tentatives sont ensuite répétées un certain nombre  $N$  de fois. À l'issue des ces  $N$  essais, seul le candidat ayant obtenu le meilleur score est conservé. Si ce score est supérieur à un seuil  $t$ , le candidat est validé et constitue l'instance de modèle décrivant les données. Dans le cas contraire, il est rejeté et on considère alors que la modélisation a échoué.

Pour le cas de la reconnaissance de formes, une primitive seule ne suffit pas nécessairement pour représenter l'ensemble des données. Le cycle de  $N$  tentatives est donc répété jusqu'à ce que le RANSAC échoue, faute de candidat satisfaisant pour représenter au minimum  $t$  points. A l'issue de chaque cycle couronné de succès, les points associés au meilleur candidat obtenu sont retirés du nuage. Notons que cette suppression permet d'assurer la convergence de l'algorithme, puisqu'il arrivera fatalement un moment où le nombre de données restante sera inférieure à  $t$ , entraînant ainsi l'échec de l'algorithme après  $N$  tentatives infructueuses.

---

**Algorithme 1** : Algorithme RANSAC pour la détection de formes.

---

```

Données :  $\mathcal{P}$ : nuage de point observé
 $\epsilon$ : tolérance à l'erreur
 $t$ : seuil de consensus
 $N$ : nombre de tentatives
 $k$ : nombre de points nécessaire à l'instanciation d'une forme
Résultat : Modèle  $\mathcal{X}$  en adéquation avec les données observées
1  $\mathcal{X} \leftarrow \emptyset$ ;
2 Tant que VRAI faire
3    $T \leftarrow t$ ;
4    $\mathcal{C}_{\text{best}} \leftarrow \emptyset$ ;
5   Pour  $i \leftarrow 1$  à  $N$  faire                                /* Cycle de tentatives */
6     Tirer aléatoirement un quorum de  $k$  points dans  $\mathcal{P}$ ;
7     Construire une forme candidate  $\mathcal{S}$  à partir de ce quorum;
8      $\mathcal{C} = \{\mathbf{p} \in \mathcal{P} | d(\mathcal{S}, \mathbf{p}) \leq \epsilon\}$ ;          /* Points décrivant  $\mathcal{S}$  ("consensus") */
9     Si  $|\mathcal{C}| > T$  alors                                    /* Sauvegarder la meilleure forme */
10       $\mathcal{S}_{\text{best}} \leftarrow \mathcal{S}$ ;
11       $T \leftarrow |\mathcal{C}|$ ;
12       $\mathcal{C}_{\text{best}} \leftarrow \mathcal{C}$ ;
13   Si  $T > t$  alors                                       /* Traiter la meilleure forme trouvée */
14      $\mathcal{X} \leftarrow \mathcal{X} \cup \{\mathcal{S}_{\text{best}}\}$ ;                /* Acceptation de la forme */
15      $\mathcal{P} \leftarrow \mathcal{P} \setminus \mathcal{C}_{\text{best}}$ ;             /* Retirer les points associés à  $\mathcal{S}_{\text{best}}$  */
16   Sinon
17     Retourner  $\mathcal{X}$ ;

```

---

**Analyse de complexité** Le paramètre clé lorsque l'on s'intéresse à l'algorithme du RANSAC pour la détection de formes (algorithme 1) est le nombre  $N$  de tentatives permettant d'assurer statistiquement la détection d'une forme au sein du nuage de points  $\mathcal{P}$ . Une analyse proposée par R. Schnabel et al. [SWK07] repose sur l'hypothèse selon laquelle tout quorum de  $k$  points pris à la surface d'une primitive permet d'engendrer effectivement cette primitive (ce n'est pas tout à fait exact en pratique, mais cette approximation permet une première estimation de  $N$ ). Dans ce cas, la probabilité  $P(S_1)$  de générer une primitive échantillonnée par  $n$  points au sein du nuage  $\mathcal{P}$  lors d'une tentative est égale à

la probabilité de sélectionner aléatoirement  $k$  points parmi ces  $n$  données:

$$P(S_1) = \frac{C_n^k}{C_{|\mathcal{P}|}^k} \approx \left( \frac{n}{|\mathcal{P}|} \right)^k$$

La probabilité  $P(S_{N_0})$  de succès après  $N_0$  tentatives est ainsi:

$$P(S_{N_0}) = 1 - (1 - P(S_1))^{N_0}$$

On peut ensuite déterminer la valeur de  $N$  au delà de laquelle la probabilité  $P(S_N)$  est supérieure à un seuil  $p$ :

$$N \geq \frac{\ln(1-p)}{\ln(1-P(S_1))} \stackrel{\text{Taylor}}{\approx} \frac{-\ln(1-p)}{P(S_1)} \quad (1.8)$$

À titre d'exemple, il faut plus de 30 milliards de tentatives pour garantir à plus de 95% la détection d'un cylindre (créé à partir de  $k = 5$  points) échantillonné par 10000 points au sein d'un nuage d'un million de points. Le paramètre  $k$  joue en particulier un rôle critique dans cette complexité. Plus le quorum nécessaire à la création d'un candidat est grand, plus il est improbable de sélectionner les points sur une même primitive, donc plus le nombre de tentatives permettant d'assurer le succès de la détection est élevé.

Il faut aussi tenir compte, en plus du nombre de tentatives, du coût de l'évaluation de chaque candidat généré. Avec une structure d'accélération appropriée (*octree* ou *KD-tree* - cf. annexe A), le calcul du score d'un candidat échantillonné par  $n$  points au sein du nuage  $\mathcal{P}$  requiert  $O(n \log(|\mathcal{P}|))$  opérations. Si l'on reprend l'exemple précédent, la détection d'un unique cylindre dans le nuage de points aura coûté, dans le pire des cas,  $200000 \times N$  opérations, soit  $6 \times 10^{15}$  opérations élémentaires.

Comme un cylindre occupe un centième de la scène, on peut raisonnablement supposer que ce nuage de points contiendra une centaine de cylindres. Pour simplifier, la reconnaissance de toutes les formes du nuage aura donc nécessité, au pire des cas,  $10^{17}$  opérations environ. Cette complexité démesurée ne permet évidemment pas d'envisager la résolution de ce problème à l'aide de l'algorithme proposé.

**Applications et améliorations** L'algorithme du RANSAC est très utilisé pour la détection de plans dans les scènes urbaines par exemple [TKLG07, BLGTK07, NSZCO10, BR05]. En effet, un plan peut n'être instancié qu'à partir de 3 points, et le traitement de gros volumes de données est donc envisageable. On note en particulier l'idée proposée par F. Bretar et M. Roux [BR05], basée sur le tirage de triplets de points possédant des normales similaires pour permettre la génération de plans candidats pertinents.

En revanche, les formes les plus complexes nécessitent de modifier l'algorithme 1 pour en améliorer l'efficacité. R. Schnabel et al. [SWK07] d'une part, puis C. Cheng et S. Lai [CL09] d'autre part, mettent en avant deux critères à améliorer pour permettre le traitement efficace des données:

**L'échantillonnage** des données doit permettre le tirage de quorums pertinents, et une probabilité uniforme sur l'ensemble des données n'est pas une approche appropriée (M. Fishler et R. Bolles faisaient déjà cette remarque dans leur article d'origine [FB87]). Plus les quorums sélectionnés sont pertinents, plus le nombre de tirages garantissant la détection d'une forme est faible et plus l'algorithme converge rapidement.

**Le calcul de score** doit être rapide, puisque cette étape doit malgré tout être répétée un grand nombre de fois (même avec une méthode intelligente pour le choix des quorums). Il n'est bien entendu pas envisageable de parcourir la totalité du nuage de points et le calcul du score doit *a minima* s'appuyer sur une structure d'accélération spatiale comme nous le disions précédemment.

La méthode proposée par R. Schnabel et al. [SWK07] apporte des améliorations judicieuses permettant d'améliorer considérablement l'efficacité de la méthode, sans la changer radicalement.

La première idée mise en avant par les auteurs est le tirage des points de chaque quorum au sein d'une même cellule d'*octree* choisie aléatoirement à une profondeur donnée. Ce processus permet ainsi de privilégier le choix de points appartenant à un même voisinage dont la taille dépend de la profondeur considérée dans l'*octree*, et augmente ainsi significativement la probabilité pour que ces points appartiennent à une même surface. Les candidats générés sont ainsi plus pertinents, et le nombre de tentatives permettant la détection de formes diminue en conséquence. Le choix de la profondeur d'*octree* est effectué en fonction de la propension de chaque niveau à générer des primitives pertinentes: plus un niveau fournit de bons candidats, plus il est probable qu'il soit choisi à nouveau.

La seconde idée des auteurs consiste en la prédiction du score d'une primitive à partir d'un échantillon de points du nuage. Le décompte des points décrivant un candidat au sein de cet échantillon correspond à une expérience de loi hypergéométrique, et on sait alors estimer un intervalle de confiance pour le score du candidat dans la totalité du nuage de points. Plus la taille de l'échantillon considéré est élevée, plus cet intervalle est resserré. Or le calcul exact du score d'un candidat n'est pas indispensable, puisque la seule opération requise par le RANSAC est la comparaison de candidats pour ne retenir que le meilleur. Ainsi, lorsque les intervalles de score de deux candidats s'intersectent, on ne peut décider lequel surclasse l'autre et il est alors nécessaire d'augmenter la taille de l'échantillon pour le calcul du score. Dans le cas contraire (si les intervalles de score sont disjoints), la conclusion est immédiate, et seuls quelques points auront suffi à déterminer le meilleur candidat.

La dernière modification apportée par les auteurs consiste en la prise en compte des normales dans le calcul du consensus. Pour décrire la forme candidate, un point doit être non seulement suffisamment proche de la forme, mais l'angle entre sa normale et la normale théorique de la forme doit aussi être inférieur à un seuil donné.

### Transformées Généralisées de Hough (TGH)

**Principe** P. Hough [Hou62] propose à l'origine une méthode à base de votes permettant la reconnaissance de lignes parmi des données 2D. Tout repose sur le constat suivant: si l'on considère qu'une droite est paramétrée par son angle vis-à-vis de l'axe des abscisses et sa distance à l'origine, l'ensemble des droites passant par un point correspond à une sinuséide dans l'espace paramétrique des droites. Ainsi chaque point d'un nuage en deux dimensions peut-il exprimer un vote dans l'espace paramétrique, sous la forme d'une courbe sinuséidale. Un point de l'espace paramétrique où se rencontrent un grand nombre de sinuséides représente un pic de votes, et correspond alors à une droite plébiscitée par le nuage de points. La détection de ces pics de votes en espace paramétrique permet alors la détection des droites échantillonnées par le nuage de points.

Cette approche peut être généralisée pour permettre la détection de formes plus complexes dans les images [DH72], et dans les données tridimensionnelles.

En pratique, l'espace paramétrique est divisé en une grille régulière dont les cellules sont destinées à recueillir les votes. Ce système permet de simplifier le décompte de votes par simple incrément d'un compteur associé à chaque cellule. En particulier, la recherche de maxima locaux est triviale du fait du voisinage défini par la grille. Mais l'inconvénient majeur de cette approche tient au fait que la discrétisation de l'espace paramétrique engendre une perte de précision dans l'espace géométrique, et l'exactitude de la détection des droites est fortement dépendante de la taille des cellules: plus les cellules sont petites, plus la détection est précise mais plus les temps de calculs sont longs.

---

**Algorithme 2** : Détection de formes par transformées généralisées de Hough.
 

---

**Données** :  $\mathcal{P}$ : nuage de point observé  
 $t$ : nombre minimal de votes  
 $n$ : échantillonnage de la grille paramétrique  
 $k$ : nombre de paramètres définissant la forme recherchée  
 Une fonction de passage entre les cellules paramétriques et les formes géométriques.

**Résultat** : Modèle  $\mathcal{X}$  en adéquation avec les données observées

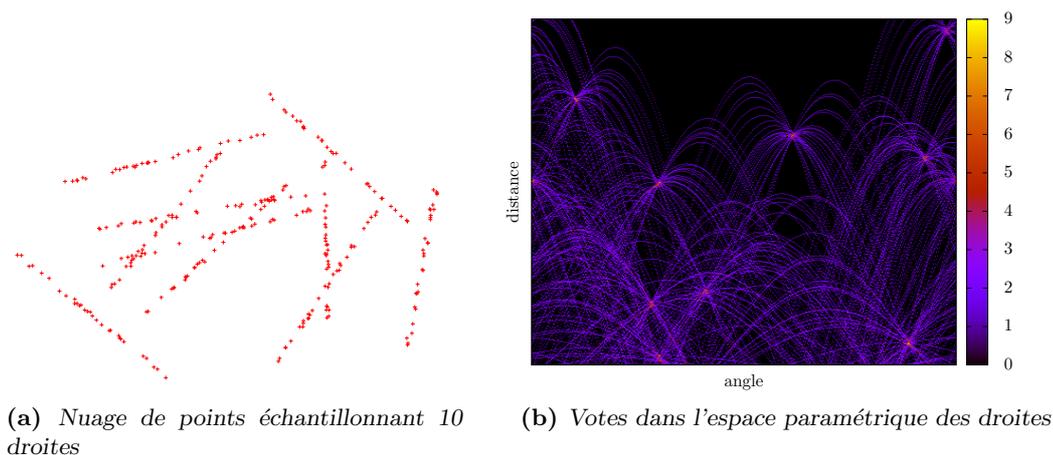
```

1  $\mathcal{X} \leftarrow \emptyset$ ;
2 Construire la grille paramétrique de  $n^k$  cellules initialisées à 0;
3 Pour tout  $p \in \mathcal{P}$  faire /* Calcul de la TGH */
4   Pour tout  $(i_1, \dots, i_{k-1}) \in [0, n-1]^{k-1}$  faire /* Balayage de la grille */
5     Calculer l'indice  $i_k$  tel que la forme engendrée par la cellule d'indices
6      $(i_1, \dots, i_{k-1}, i_k)$  passe par le point  $p$ ;
7     Incrémenter la valeur de la cellule  $(i_1, \dots, i_{k-1}, i_k)$ ;
8 Pour tout  $(i_1, \dots, i_k) \in [0, n-1]^k$  faire /* Identification des pics de votes */
9   Si la cellule  $(i_1, \dots, i_k)$  possède une valeur supérieure à toutes ses voisines et si
10  cette valeur est supérieure à  $t$  alors
11      $\mathcal{S} \leftarrow$  forme engendrée par la cellule  $(i_1, \dots, i_k)$ ;
12      $\mathcal{X} \leftarrow \mathcal{X} \cup \{\mathcal{S}\}$ ;
  
```

---

**Analyse de la complexité** Ce système de vote nécessite, pour tout point, d'explorer une partie de l'espace paramétrique. Si le modèle est défini par  $k$  paramètres, il est alors nécessaire de balayer l'ensemble des valeurs réalisables pour  $k-1$  de ses paramètres, le dernier pouvant ensuite être calculé grâce à la contrainte fixée par le point votant. Si l'on considère que la grille paramétrique est divisée en  $n$  cellules pour chacun des paramètres, un vote requiert donc  $n^{k-1}$  opérations. Le calcul de la transformée de Hough (la grille voxelique des votes) du nuage de points  $\mathcal{P}$  présente ainsi une complexité en  $O(|\mathcal{P}| \times n^{k-1})$ . S'ajoute à cela la détection des maxima dans la grille paramétrique ( $O(n^k)$ ).

À titre d'exemple, la détection de cylindres (5 paramètres) dans un nuage d'un million de points nécessite  $10^{14}$  opérations pour le calcul de la transformée de Hough divisée en  $n = 100$  cellules le long de chaque axe (ce qui est très peu), puis plus de  $10^{10}$  opérations pour identifier les pics de votes. Le nombre de paramètres définissant la forme recherchée est ici un paramètre critique pour la complexité, et la méthode ne semble pas applicable telle quelle pour la détection de formes 3D autres que le plan, ou éventuellement la sphère. En outre, la méthode requiert de petites cellules à des fins de précision. Or plus les cellules sont petites, plus elles sont nombreuses dans la grille paramétrique et il est souhaitable



**Figure 1.23:** Transformée de Hough d'un nuage de points. On considère qu'une droite est définie par l'angle qu'elle forme avec l'axe des abscisses, et sa distance à l'origine.

de limiter ce nombre pour éviter une explosion de la complexité de la méthode, tant concernant les temps de calcul que la mémoire requise pour le stockage de la grille.

**Applications et améliorations** Tout comme pour l'algorithme RANSAC, la détection de formes par transformées de Hough se prête parfaitement à la détection de plans dans les données 3D de scènes urbaines typiquement [TKLG07, VD01, OTDS04, OBKI04] (façades, toitures, etc.), mais ne peut être appliquée telle quelle pour la reconnaissance de formes plus complexes.

L'intérêt de la méthode apparaît lorsque sa complexité excessive est réduite [Kho07, RV05, Rab06]. Les travaux de T. Rabbani et F. Van Den Heuvel [RV05] sont particulièrement intéressants dans ce domaine. Les auteurs proposent en effet une méthode permettant la détection de cylindres par une méthode à base de transformée de Hough. Les auteurs proposent de briser la complexité de ce problème en séparant le traitement des paramètres définissant le cylindre: au lieu de calculer une transformée de Hough en 5 dimensions, on procède en calculant une transformée en 2 dimensions, puis une seconde en 3 dimensions.

La première étape consiste en l'estimation des axes de cylindres (2 paramètres) par transformée de Hough de la sphère gaussienne du nuage de points, de manière à en extraire les plans (cette idée se retrouve dans [CG01], où les plans sont extraits par RANSAC, et dans [PP03]). Le vote d'un point de la sphère de Gauss s'exprime comme l'intersection d'un plan et de la sphère unitaire centrée  $\mathbb{S}^2$ , de sorte que la détection d'un plan dans la sphère gaussienne par transformée de Hough se formule comme la recherche du point sur la sphère unitaire de Hough correspondant à un pic de votes. Ce processus permet ainsi d'isoler les différents axes de cylindres présents dans la scène.

Cette première étape permet de traiter efficacement une partie du problème de la reconnaissance de cylindres. Il reste toutefois à identifier les positions et rayons des cylindres (3 paramètres). Les auteurs proposent pour ce faire d'étudier chaque direction d'axe détectée, puis de projeter les points du nuage dont la normale est approximativement orthogonale à cet axe dans le plan perpendiculaire à la direction étudiée. Une fois projetés dans ce plan, les points d'un même cylindre décrivent un cercle commun dont la position et le rayon renseignent directement sur la position et le rayon du cylindre recherché. Une

transformée de Hough des points projetés est donc calculée pour permettre la détection de cercles. Les cylindres reconnus peuvent ensuite être reconstruits à partir des axes et des cercles détectés.

### 1.3.4 Appariement de formes similaires

Un champ entier du domaine de la reconnaissance de formes traite de l'appariement de tout ou partie de scènes, en s'appuyant sur la détection des similarités parmi les formes qu'elles représentent. Ce problème peut par exemple se poser comme la recherche, au sein d'une base de données, de formes qui ressemblent à une forme requête [WK10, FS06, FMA<sup>+</sup>10, FO09] (*shape retrieval*). Un tout autre problème consiste en la recherche de symétries ou de régularités au sein d'une même scène [MLMM01, MGP06, NSZCO10, SSS08].

Une part significative des travaux dans ce domaine s'appuie sur une démarche commune. Il s'agit dans un premier temps de proposer un calcul de descripteur de la surface échantillonnée, définissant sa géométrie locale en un point donné. Ces descripteurs doivent, dans la mesure du possible, être invariants vis-à-vis des translations, rotations et changements d'échelle pour permettre la reconnaissance de formes indépendamment du repère dans lequel elles sont exprimées. La comparaison de ces descripteurs permet dans un second temps l'évaluation quantitative de la similarité entre parties issues des scènes étudiées. La dernière partie de la démarche consiste en la mise en place d'une méthode efficace de recherche de similarités basée sur la "distance" proposée pour la comparaison de descripteurs. Une approche commune consiste typiquement en la mise en place d'algorithmes de regroupement ou de traitements avancés dans l'espace des descripteurs.

Parmi les divers descripteurs utilisés dans la littérature, on notera par exemple les images de rotation proposées par A. Johnson et M. Hebert [JH99], ou les motifs saillants proposés par R. Gal et D. Cohen-Or [GCO06].

L'image de rotation (*spin image*) en un point  $\mathbf{p}$  est issue d'un processus de collecte des points voisins de  $\mathbf{p}$  lorsqu'un plan tourne sur lui-même autour de l'axe défini par  $\mathbf{p}$  et sa normale. Lors de cette rotation, des accumulateurs organisés en grille sur la surface du plan de rotation décomptent les points qu'ils rencontrent lors de la collecte, fournissant ainsi une image décrivant la géométrie au voisinage du point étudié. Pour permettre le traitement et la comparaison de ces images, les auteurs proposent de les exprimer comme des vecteurs, puis de les compresser pour permettre une représentation compacte et ainsi accélérer les traitements.

Les motifs saillants (*saillant features*), quant à eux, reposent sur l'analyse de la courbure de la surface (les auteurs traitent des maillages surfaciques). Chaque descripteur est ici un patch surfacique décrivant une quadrique. La saillance de chaque patch est ensuite évaluée sur la base notamment de sa courbure et de la variance de la courbure sur la région qu'il représente (le patch et son voisinage). Les auteurs préconisent ensuite de stocker les descripteurs jugés comme étant saillants dans une table de hachage dont les clés sont basées sur les valeurs ayant permis de mesurer la saillance. La fonction de hachage détermine ainsi une signature géométrique pour chaque patch permettant ainsi de retrouver rapidement des formes similaires au sens de la saillance.

Comme toutes les approches entreprenant l'étude locale des données, les traitements à base de descripteurs sont par nature dépendants de l'échelle considérée. La prise en compte de voisinages trop étroits ne permet pas de discriminer les descripteurs, puisque la plupart des points dans les données représentent des surfaces lisses, assimilables localement

à des plans (on obtiendrait alors uniquement des descripteurs de plans). Au contraire, si les voisinages considérés sont trop larges, il devient peu probable d'établir une correspondance entre descripteurs à moins que les surfaces comparées soient rigoureusement équivalentes, ce qui est rarement le cas dans les applications visées.

De plus, les approches à base de descripteurs supposent une certaine variété dans la géométrie des surfaces, de manière à pouvoir distinguer les parties entre elles. Certains objets manufacturés présentent une surface très régulière, et il devient alors complexe d'établir une correspondance sur la seule base de l'étude locale des données, puisque tous les descripteurs risquent alors d'être semblables.

## 1.4 Reconstruction de scènes 3D

La reconstruction de scènes 3D est le processus de recherche de la surface à partir de laquelle les données 3D ont été acquises (on parle aussi de modélisation géométrique). Ce problème recoupe très largement celui de la reconnaissance de formes et d'ajustement de surfaces. Toutefois, nous faisons ici une très légère distinction entre ces champs disciplinaires dans la mesure où un certain nombre de travaux de reconstruction intègrent, pour être fiables, des connaissances *a priori* plus ou moins évoluées quant aux scènes traitées. Ces connaissances permettent de contraindre la recherche de surface de manière à intégrer certaines spécificités des scènes traitées qui ne sauraient être retrouvées à partir des seules données.

La reconstruction se divise en deux domaines principaux, selon que l'on cherche à reconstruire une surface quelconque ou un assemblage de primitives. Dans le premier cas, le résultat de la reconstruction est typiquement une surface maillée ou une surface implicite approximant ou interpolant, selon les travaux, les données portées par le nuage de points. Dans le second cas, les primitives géométriques apportent un certain niveau de contrainte et permettent généralement de représenter les objets manufacturés ou des scènes architecturales, puisqu'une part considérable des réalisations humaines peut se décrire à l'aide d'une poignée de formes seulement (plans, ellipsoïdes, cylindres, cônes et tores).

### 1.4.1 Surfaces libres

La reconstruction de surfaces quelconques fait l'objet de très nombreux travaux, et la thèse de R. Allègre [All06] fournit un état de l'art très abouti dans ce domaine. On distingue deux approches visant à produire une surface représentant les données d'acquisition. La première, que l'on qualifie d'approche "combinatoire", vise à reconstruire un maillage surfacique en établissant les relations d'adjacence entre points du nuage. Le maillage ainsi obtenu s'appuie donc sur les points du nuage, et la reconstruction est une interpolation des données. La seconde démarche repose sur des méthodes d'approximation par ajustement de surfaces, et les surfaces qu'elle produit ne passent généralement pas exactement par les points du nuage. L'avantage de cette seconde approche sur la première tient au fait qu'elle permet d'estomper les défauts d'acquisition des données 3D (cf. section 1.2.4).

**Approche combinatoire** Les approches combinatoires permettent donc de calculer un maillage dont les sommets sont les points du nuage à reconstruire. Il s'agit donc "uniquement" de proposer des relations d'adjacence entre les points de manière à définir les arêtes et faces du maillage reconstruit. Or, il peut exister une multitude de maillages décrivant un même nuage de points [Ede03]. Pour permettre la reconstruction, il est donc nécessaire de définir une contrainte de régularisation portant sur des propriétés du maillage

de manière à bien définir le problème en restreignant l'ensemble des solutions acceptables (avec idéalement l'unicité de la solution au problème posé).

La contrainte de Delaunay est notamment très utilisée en géométrie algorithmique. Un complexe satisfait la contrainte de Delaunay si les hypersphères circonscrites à chacun de ses simplexes<sup>8</sup> ne contiennent en leur intérieur aucun autre point que les sommets du simplexe à partir duquel elles sont définies. Les maillages de Delaunay vérifient un certain nombre de propriétés intéressantes, et permettent notamment d'obtenir de "belles" triangulations au sens de l'uniformité géométrique des triangles qui les composent.

F. Cazals et J. Giesen [CG04] présentent un état de l'art des algorithmes permettant la construction de triangulations interpolant un nuage de points s'appuyant sur les complexes de Delaunay. On notera par exemple l'approche par suppression des tétraèdres de Delaunay de manière à ne conserver que les triangles à l'interface entre l'intérieur et l'extérieur de l'objet échantillonné [AS00], ou encore l'approche par expansion ajoutant à chaque étape un nouveau triangle de Delaunay choisi parmi ceux qui sont adjacents aux bordures du maillage courant [CSD04]. L'algorithme par pivotement de boules [BM99] permet, selon un principe similaire d'expansion de la triangulation, de reconstruire le maillage interpolant le nuage de points sans toutefois utiliser les contraintes des triangulations de Delaunay. L'algorithme par convection géométrique [Cha03, All06] permet quant à lui de reconstruire une triangulation par contraction itérative d'une surface d'origine enveloppant le nuage de points, jusqu'à l'obtention d'un maillage s'appuyant sur les données et ne pouvant être contracté de nouveau. Cette convection géométrique aboutit à une surface maillée satisfaisant la propriété de Gabriel orientée selon laquelle la demi sphère minimale orientée "vers l'intérieur" circonscrite à une face est vide de tout point du nuage à l'exception des sommets de la face qui la définit.

Mentionnons pour finir l'approche *power crust* [ACK01, MAVdF05], visant à calculer l'axe médian d'un nuage de points en s'appuyant sur son diagramme de Voronoï<sup>9</sup>, et permettant ainsi (en déterminant l'inverse de cette transformée) de reconstruire la surface qui lui correspond.

**Approche par ajustement** Les approches de reconstruction par ajustement reposent quant à elles sur la recherche de la surface décrivant au mieux les données sous certaines contraintes dépendant du problème traité. Ces approches résultent la plupart du temps en une surface approximant les données, mais peuvent aussi parfois les interpoler [TO02, OBS03].

Il existe de grandes quantités de travaux intégrant cette approche. Nous avons d'ores et déjà mentionné la méthode des moindres carrés glissants (cf. section 1.3.1), basée sur l'approximation locale du nuage par des surfaces simples. La pondération des points en fonction de leur éloignement au centre d'intérêt local permet au final l'obtention d'une surface lisse approximant les données [AA03, MVD03, SF05, PKKG03]. Des nuances peuvent toutefois être introduites dans la méthode pour permettre l'obtention d'arêtes vives

---

<sup>8</sup>Un simplexe dans un espace à  $d$  dimensions est l'enveloppe convexe définie par  $d + 1$  sommets de cet espace (segment dans  $\mathbb{R}$ , triangle dans  $\mathbb{R}^2$ , tétraèdre dans  $\mathbb{R}^3$ , etc.). Un complexe est un assemblage organisé de simplexes (triangulation dans  $\mathbb{R}^2$ , tétraédrisation dans  $\mathbb{R}^3$ , etc.).

<sup>9</sup>Le diagramme de Voronoï d'un ensemble de sommets dans un espace euclidien est une partition de l'espace en cellules convexes associées à chacun des sommets. Une cellule de Voronoï se définit comme l'ensemble des points de l'espace plus proches du sommet associé à cette cellule que de tout autre sommet. Le complexe de Delaunay construit à partir de l'ensemble des sommets correspond en particulier au dual du diagramme de Voronoï : une arête de Delaunay relie deux sommets si et seulement si les cellules de Voronoï correspondantes son adjacentes.

conformes à la géométrie portée par les données [FCOS05].

Les surfaces implicites sont très utilisées pour permettre la reconstruction d'ensemble de points [BC00, ZOF01]. Y. Ohtake et al. [OBA+03] proposent par exemple un algorithme reconstruisant une surface implicite définie par une fonction de distance à la surface échantillonnée, construite à partir d'un mélange pondéré de quadriques reconnues localement dans le nuage de points à différentes échelles. Parmi les surfaces implicites, on distingue notamment les fonctions à base radiale (*Radial Basis Function* - RBF), définissant en tout point de l'espace un potentiel fonction de la distance de ce point à un centre donné. Le problème de la reconstruction se pose alors comme la recherche des paramètres des fonctions à base radiale induisant un champ de potentiel dont on puisse extraire une surface implicite (surface de niveau) interpolant ou approximant les données [OBS03, SPOK95, TO02]. La formulation de ce problème dépend bien évidemment de l'expression choisie pour les RBF.

Dans un tout autre registre, l'approche par modèles déformables permet la reconstruction d'un maillage surfacique final à partir de la distorsion d'un maillage initial, de manière à ce que la surface résultante vienne se "coller" aux données. L'algorithme proposé par A. Sharf et al. [SLS+06] repose par exemple sur la déformation de fronts du maillage guidée par un champ scalaire défini à partir des distances aux points du nuage. Plus les fronts sont éloignés du nuage, plus il évoluent rapidement vers la surface cible.

La méthode proposée par Y. Ohtake et al. [OBS06] se situe à mi-chemin entre les approches par ajustement de surface et les approches combinatoires. Les auteurs proposent essentiellement une méthode établissant l'adjacence entre des points via le calcul de sphères couvrant entièrement la surface échantillonnée. Les rayons de ces sphères sont calculés de manière à ce qu'elles englobent des portions planes de surfaces, et permettent dans un second temps de définir ce que les auteurs appellent des "points auxiliaires" issus d'un processus de lissage des données au sein de chacune des sphères. Le maillage est ensuite calculé en construisant une face triangulaire pour chaque triplet de points auxiliaires dont les sphères associées s'intersectent. Le maillage n'interpole donc pas les données, puisqu'il s'appuie sur les points auxiliaires.

### 1.4.2 Primitives géométriques

La reconstruction de scènes 3D à base de primitives géométriques relève en partie du domaine de la reconnaissance de formes évoqué précédemment (section 1.3.3). Cependant, une partie des auteurs approfondit la "simple" détection de formes en intégrant au problème des considérations quant à la cohérence du modèle 3D reconstruit. Un agrégat de formes ne fournit en effet une description satisfaisante de la scène que s'il vérifie un certain nombre de contraintes, notamment la consistance de l'assemblage des formes [Fis04, BMV01, LGM+04].

Conscients de l'intérêt majeur de la prise en compte de la structuration des primitives reconstruites, R. Schnabel et al. [SWWK08] proposent une approche permettant, en guise de post-traitement, d'identifier des entités sémantiques parmi les groupes de primitives. La démarche des auteurs repose sur la mise en place d'un graphe topologique représentant les relations de voisinage entre objets reconnus dans un premier temps par RANSAC. L'utilisateur peut ensuite définir un graphe "requête" définissant la topologie des objets qu'il recherche, et l'algorithme sera ensuite chargé d'extraire des sous-graphes similaires à cette requête au sein du graphe topologique entier, fournissant ainsi des groupes de

primitives cohérents vis-à-vis de l'assemblage recherché. Le problème de l'extraction de sous-graphes est en théorie NP-complet, mais les auteurs proposent d'adopter une démarche force brute sous l'hypothèse que les graphes requêtes sont relativement simples.

Y. Li et al. [LWC<sup>+</sup>11] proposent une approche particulièrement intéressante centrée sur la structuration de l'assemblage des primitives. Les auteurs s'appuient sur l'ensemble des primitives extraites par RANSAC, pour y détecter des dépendances d'orientation (parallélisme, orthogonalité, etc.), d'alignement (coplanarité, coaxialité, etc.) et de forme (dimensions, rayons, etc.). Pour chacun des ces trois critères, un graphe reliant les formes est construit de manière à apparier les primitives proches de satisfaire les contraintes correspondantes. La régularisation est ensuite traitée en minimisant l'erreur moyenne quadratique des primitives vis-à-vis des données, sous les contraintes induites par chaque graphe. À l'issue de ce processus, l'algorithme fournit un ensemble de formes satisfaisant rigoureusement les contraintes *a priori* d'orientation, d'alignement et de similarité, et s'ajustant au mieux sur les données sous ces contraintes. Ce processus est ensuite répété sur les points non encore reconnus.

Les travaux de thèse de R. Bénére [Bí2] quant à eux s'articulent autour de la reconstruction de primitives géométriques simples à partir de maillages surfaciques. Dans un premier temps l'auteur propose d'identifier les primitives distinctes de l'objet représenté grâce à une approche de segmentation s'appuyant sur les courbures calculées dans le maillage. Par la suite, les relations d'adjacence entre les primitives détectées sont extraites, de manière à identifier les intersections entre les différents objets et en déduire un modèle B-Rep permettant une représentation cohérente de la scène d'un point de vue topologique.

Dans sa thèse, T. Chaperon [Cha02] s'intéresse à la détection de lignes de tuyauterie dans les nuages de points représentant des scènes industrielles, reconstruites à partir de cylindres principalement, puis de tores et de cônes. La structuration de la scène est intrinsèquement traitée par la méthode du fait même qu'elle porte sur la reconnaissance de lignes de tuyauterie. On suppose dans un premier temps que l'utilisateur indique un point appartenant à la surface d'un tuyau, fournissant ainsi une information fiable permettant d'éviter la recherche automatique (et en générale incertaine) de formes cylindriques dans la totalité du nuage. Partant de cet *a priori*, l'algorithme proposé se charge ensuite de propager la détection des tuyauteries par extraction des portions cylindriques de proche en proche dans le nuage de points. Ces portions sont suffisamment restreintes pour permettre une approximation linéaire par morceaux des parties non rectilignes de tuyauterie. La chaîne de cylindres ainsi détectée est ensuite analysée pour permettre la reconnaissance de formes tels que les tores ou cônes.

### 1.4.3 Hybride

F. Lafarge et al. [LKBH10] présentent une approche astucieuse et aboutie mêlant reconstruction de surfaces maillées et reconnaissance de primitives géométriques. Les auteurs posent ce problème de reconstruction comme l'optimisation d'une fonctionnelle embarquant divers termes, vérifiant notamment l'adéquation entre le modèle reconstruit et les données, les propriétés du maillage (qui doit être lisse) et la pertinence des primitives détectées basée sur leur placement respectif (les primitives parallèles et perpendiculaires sont favorisées). Ce problème d'optimisation est ensuite résolu à l'aide d'un algorithme

probabiliste *Jump-Diffusion* (détaillé en section 3.2.2), dans lequel certaines parties du maillage reconstruit se voient notamment transformées en cylindres lorsque cette opération est pertinente.

#### 1.4.4 Recours à l'*a priori*

Les travaux sur la reconstruction ont vu l'émergence du recours à des connaissances *a priori* de haut niveau afin de permettre le traitement des données en fonction de la nature des scènes qu'elles représentent. Par exemple, le simple fait de considérer que les scènes se décrivent comme des assemblages de primitives [RV05, SWK07] constitue en soit un premier niveau d'*a priori*. Nous avons vu précédemment (section 1.4.2) qu'il était possible d'incorporer à ce type de reconstruction une information d'ordre sémantique [Cha02, SWK07, LWC<sup>+</sup>11] basée sur la relation entre les primitives reconstruites afin de contraindre la reconstruction et ainsi fournir des résultats plausibles.

L'attache du modèle 3D aux données ne peut fournir à elle seule aucune garantie quant à la cohérence de la reconstruction. Deux démarches permettent d'intégrer la régularisation du modèle au problème de reconstruction:

**intégration des contraintes** dans l'algorithme de reconstruction lui-même, de manière à fournir par construction un modèle se conformant rigoureusement aux attentes *a priori*

**un terme de régularisation** quantifiant l'adéquation d'une solution vis-à-vis des attentes *a priori*, permet de rechercher un modèle 3D vérifiant un certain équilibre entre l'attache aux données et la satisfaction des contraintes *a priori*. L'approche Bayésienne [DTB06, JWB<sup>+</sup>06] ou les méthodes à base de minimisation de fonctionnelles [LKBH10] s'inscrivent typiquement dans cette catégorie.

L'approche de Y. Li et al. [LWC<sup>+</sup>11] correspond par exemple au premier cas de figure: bien qu'il existe une évaluation numérique de la satisfaction des contraintes, l'algorithme proposé garantit par construction que le modèle reconstruit est exactement conforme aux contraintes d'orthogonalité de parallélisme, d'alignement et de régularité des formes. L'ajustement aux données n'est assuré qu'une fois le modèle régularisé.

Les travaux de A. Chauve [CLP10] mêlent les deux démarches. Les auteurs proposent en effet une approche déterministe permettant d'intégrer des *a priori* visant à deviner les parties incomplètes du modèle ou renforcer la verticalité et l'orthogonalité des plans détectés. Les auteurs proposent dans un second temps une approche permettant de déterminer l'espace occupé par la scène dans une partition spatiale construite au préalable, par optimisation d'une fonctionnelle intégrant un terme numérique de régularisation du résultat.

R. Fisher [Fis04] met en avant l'intérêt de l'utilisation d'*a priori* pour permettre l'obtention de résultats fiables, en présentant les nombreux apports de ce type de démarche dans la cadre de la détection de primitives géométriques, via notamment la prise en compte des relations entre les formes détectées.

Dans les travaux de J. Schmittwilken et L. Plümer [SP10] par exemple, le recours à des connaissances *a priori* est utilisé pour guider l'algorithme RANSAC afin de permettre la détection efficace de fenêtres ou d'escaliers dans des scènes urbaines. R. Gal et al. [GSH<sup>+</sup>07] quant à eux utilisent une base de formes *a priori* pour permettre la reconstruction fiable de surfaces. Les auteurs utilisent en effet des patches surfaciques issus de cette base, qu'ils tentent ensuite de reconnaître parmi les données. La méthode proposée

par les auteurs permet d'identifier des nuances fines dans les courbures de la scène étudiée grâce à des informations géométriques précises embarquées dans les patches *a priori*. La plupart des approches d'ajustement sans *a priori* ne permettent pas un tel niveau de précision, puisqu'elles ne portent aucune connaissance quant au résultat recherché et tendent à lisser toutes les parties reconstruites (moindres carrés glissants par exemple).

Le recours à un modèle CAO existant n'intervient, à notre connaissance, que dans les travaux de F. Bosche [Bos10]. L'auteur propose en effet d'aborder la reconstruction comme un problème de recalage des primitives composant le modèle CAO disponible sur les parties du nuage qui leur correspondent. Après un recalage grossier de l'ensemble des données (modèle CAO existant et nuage de points), chaque primitive est ainsi recalée finement sur le nuage de points grâce à un algorithme de type ICP (cf. section recalage). Ces travaux sont très proches de la problématique de cette thèse. La formulation sous forme de problème de recalage ne permet toutefois pas de tenir compte des disparitions ou apparition d'objets entre les scènes, ni des changements significatifs pouvant survenir entre les scènes étudiées puisque chaque objet du modèle CAO n'est ajusté sur la partie du nuage dont il est le plus proche.

## 1.5 Conclusion

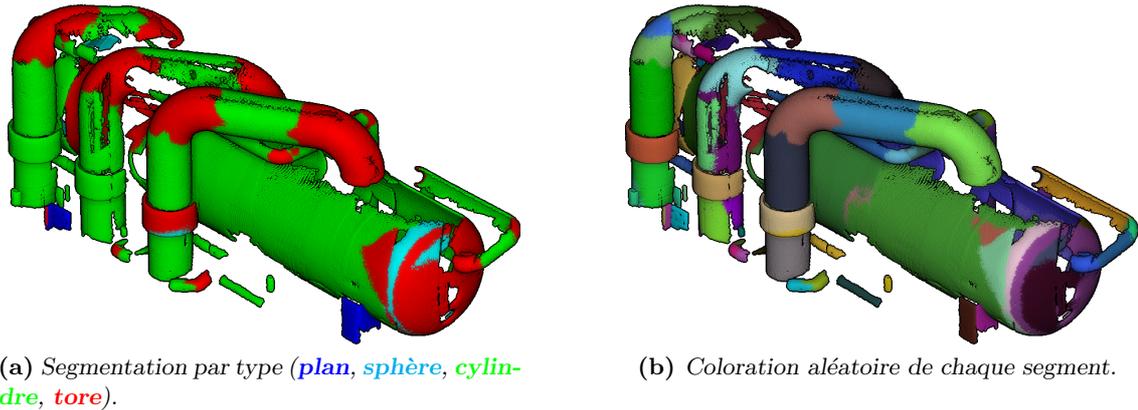
### 1.5.1 Positionnement par rapport à l'existant

Nous positionnons nos propres travaux dans le cadre de la reconstruction de modèles CAO à base de primitives géométriques, puisque la pertinence de cette représentation est reconnue pour le traitement des environnements industriels ou des objets manufacturés [RV05, Cha02, DLP08, KBK<sup>+</sup>04, LWC<sup>+</sup>11], et est d'ailleurs d'ores et déjà utilisée pour la représentation des scènes existantes.

Les approches d'ajustement de formes distinctes dans le nuage de points, telles que le RANSAC proposé par R. Schnabel et al. [SWK07] ou les transformées généralisées de Hough de T. Rabbani et F. Van Den Heuvel [RV05] permettent effectivement d'obtenir de premiers résultats intéressants (en particulier l'algorithme RANSAC dont l'efficacité en terme de temps de calcul est remarquable). Mais ces approches ne permettent pas tenir compte de la cohérence de la reconstruction globale, qui est pourtant un élément crucial du problème. L'étude des résultats de l'algorithme RANSAC [SWK07] par exemple (cf. figure 1.24) met en avant les lacunes de la méthode quant à la gestion de formes concurrentes, puisqu'il arrive que l'algorithme produise des primitives qui se chevauchent. En outre, l'algorithme tend à confondre les divers types de primitives (tores à la place de cylindres, cylindres à la place de plans, etc.) qu'il gère simultanément au cours du processus de reconnaissance de formes. Il nous semble ainsi plus prudent de ne pas chercher à mettre en concurrence l'ensemble des formes au sein d'un même algorithme, mais plutôt de hiérarchiser le traitement de chaque type de primitive en fonction du rôle qu'il peut jouer dans la scène traitée.

Parallèlement, certains auteurs tels que R. Fisher [Fis04], Y. Li et al. [LWC<sup>+</sup>11] ou F. Langbein [LGM<sup>+</sup>04] entre autres mettent en avant, à juste titre, la nécessité de la prise en compte de contraintes pour permettre la représentation cohérente des scènes reconstruites.

Des travaux existants [LWC<sup>+</sup>11, WK10, NSZCO10] s'appuient sur des algorithmes de détection fournissant une première estimation des formes, qui pourront par la suite être



**Figure 1.24:** Résultat de l’algorithme RANSAC [SWK07] sur une scène industrielle. On constate des chevauchements de formes au niveau de la cuve (cylindre horizontal central) ainsi que sur le second tuyau vertical d’arrivée (au centre des trois). On note également que les tores tendent à être privilégiés vis-à-vis des autres formes (extrémités de cuve).

utilisés dans un processus plus évolué. Pour notre part, nous pensons qu’il est pertinent d’intégrer la régularisation du modèle CAO au problème de reconstruction lui-même. Le respect des contraintes de modélisation n’est pas nécessairement indépendant du problème de la détection des parties qui composent le modèle, et il est préférable à ce titre de résoudre les deux problèmes simultanément et équitablement afin de ne pas privilégier arbitrairement l’un de ces deux aspects de la reconstruction. Dans cette optique, l’approche Bayésienne rencontrée par exemple pour la reconstruction de surfaces maillées [JWB<sup>+</sup>06,DTB06] est une excellente candidate à la formulation du problème de reconstruction intégrant des connaissances *a priori*.

### 1.5.2 La reconstruction en pratique

Malgré la multitude de travaux disponibles à l’heure actuelle en matière de reconstruction, la modélisation est aujourd’hui traitée quasi exclusivement par l’intervention humaine. En effet, aucun algorithme de reconstruction automatique existant ne semble en mesure d’offrir des résultats suffisamment fiables pour être utilisés directement dans un contexte industriel. Ceci rend nécessaire la vérification des résultats par l’utilisateur, qui s’avère être coûteuse puisqu’elle nécessite autant d’efforts que la reconstruction elle-même. Dans la mesure où la machine ne semble pas être en mesure de résoudre seule ce problème, cette tâche est donc laissée à l’expert humain et seuls les problèmes parfaitement maîtrisés par l’ordinateur, tels que l’ajustement de surfaces sur des portions de nuages isolées au préalable, sont automatisés.

Le processus de reconstruction interactif se déroule de manière séquentielle :

**Segmentation** l’utilisateur isole une partie du nuage de points représentant une primitive qu’il aura reconnue. La segmentation interactive procède par sélection des points à l’intérieur d’une région dont les contours sont définis à l’écran par l’utilisateur. Comme le volume correspondant à ces contours dans  $\mathbb{R}^3$  n’est pas délimité en profondeur, il est souvent indispensable de répéter cette opération sous divers points de vue, de manière à isoler correctement un ensemble de points représentant une primitive unique.

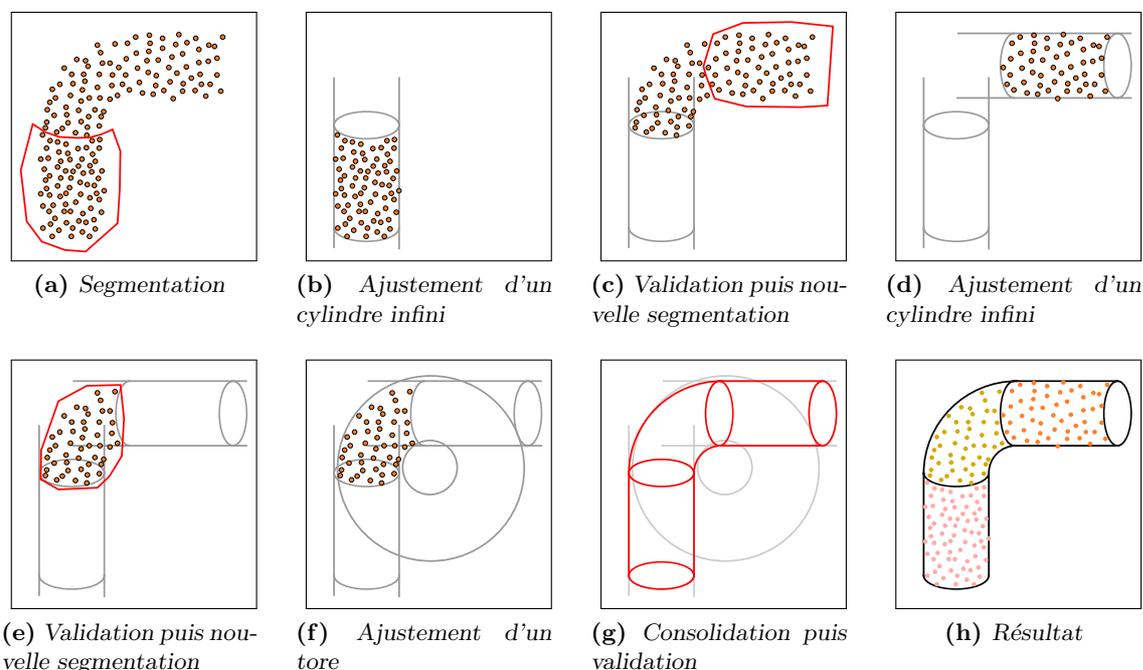
**Ajustement** l’utilisateur décide du type de primitive destiné à représenter le segment

isolé lors de la phase de segmentation. Un algorithme d'ajustement se charge ensuite d'estimer les paramètres de la primitive décrivant au mieux les données.

**Consolidation** lorsque la primitive ajustée s'avère être dépendante d'autres primitives déjà reconnues, l'utilisateur a la possibilité de notifier ces dépendances au logiciel de reconstruction (coaxialité, identité de rayons, orthogonalité, parallélisme, etc.). Le programme modifie ensuite la primitive de manière à ce qu'elle satisfasse les contraintes spécifiées. Cette démarche est par exemple utilisée pour déterminer les limites de cylindres, en s'appuyant sur des plans détectés au préalable.

**Validation** une fois les étapes précédentes réalisées, l'utilisateur peut valider la primitive reconstruite. Cette primitive est ainsi définitivement intégrée au modèle, et les points du segment utilisé pour sa reconstruction sont retirés du nuage de points.

Ces différentes étapes sont répétées jusqu'à ce que la quasi totalité des points aient été validés. Les points restant dans le nuage constituent les données non reconstruites.



**Figure 1.25:** Reconstruction manuelle d'un nuage de points. En alternant diverses phases de segmentation, d'ajustement et de validation, l'utilisateur isole et identifie les différentes primitives (tore et cylindres) et les points qui leur correspondent dans le nuage. Au moment de la validation du tore dont l'ajustement n'est pas suffisamment précis, l'utilisateur spécifie des contraintes (consolidation) permettant d'obtenir un assemblage cohérent des formes pour représenter le coude de tuyauterie de cette scène.

La segmentation de gros volumes de données est en pratique effectuée selon une démarche hiérarchique, puisque l'isolement de petites quantités de points au sein de la totalité d'un immense nuage est une activité très pénible. D'autant plus que la visualisation de nuages de points n'est pas forcément intuitive, et requiert un effort perceptif considérable de la part de l'utilisateur du fait de la superposition des points en profondeur notamment. L'expert isole donc dans un premier temps des bâtiments distincts, puis sépare ensuite les différentes salles, etc. et peut finalement isoler les primitives dans ces parties contenant une quantité raisonnable de données. Cette hiérarchie peut ensuite naturellement se retrouver

dans le modèle CAO reconstruit.

La reconstruction de certains bâtiments de centrale nucléaire peut ainsi nécessiter des mois de travail de la part d'équipes d'experts en reconstruction CAO (1 à 5), et l'apport d'outils semi-automatiques fiables tels que ceux proposés par T. Chaperon [Cha02] peut considérablement soulager les experts réalisant cette tâche.



# Chapter 2

## Formulation du problème

### Sommaire

---

<b>2.1</b>	<b>Données et connaissances <i>a priori</i></b>	<b>65</b>
2.1.1	Nuage de points	65
2.1.2	Modèle 3D CAO	65
2.1.3	Connaissances métier	69
<b>2.2</b>	<b>Formulation probabiliste du problème</b>	<b>70</b>
2.2.1	Formalisme Bayésien	70
2.2.2	Application au cas de la reconstruction	73
<b>2.3</b>	<b>Proposition d'énergies</b>	<b>75</b>
2.3.1	Attache aux données	75
2.3.2	Énergie géométrique	81
2.3.3	Énergie topologique	85
2.3.4	Évaluation des paramètres de pondération	95

---

---

### Préambule

---

*Le problème de reconstruction, dans le contexte qui nous intéresse ici, vise à produire des modèles CAO construits à base de primitives géométriques simples (section 2.1.2) satisfaisant simultanément trois attentes. La première de ces attentes concerne évidemment le respect des mesures acquises sur site, puisque ces données constituent la vérité terrain que le modèle CAO reconstruit doit lui-même représenter. En outre, nous disposons dans nos travaux de deux sources d'a priori dont on souhaite tenir compte pour renforcer la fiabilité du résultat. Tout d'abord, le modèle CAO reconstruit doit se conformer, dans l'assemblage de ces primitives, à certaines règles issues de connaissances quant à la structure de la scène traitée (section 2.1.3). De plus, pour être fiable, le résultat de la reconstruction doit être semblable à un modèle CAO existant supposé proche de la solution recherchée.*

*Avant toute chose, il est nécessaire d'établir les fondements du problème que l'on cherche à résoudre. Il s'agit de proposer une formalisation de ce problème qui permette de gérer la satisfaction simultanée de ces trois différentes contraintes. Or il apparaît que le formalisme Bayésien se prête particulièrement bien au traitement des problèmes d'adéquation aux observations sous contraintes d'a priori quant au résultat recherché, et nous choisissons donc d'adopter ce paradigme pour la formulation du problème de reconstruction (section 2.2). Dans ce formalisme, chaque contrainte s'exprime sous la forme d'une probabilité quantifiant la conformité d'un modèle CAO à son encontre, et la solution recherchée est alors simplement le modèle CAO permettant de maximiser le produit de ces trois probabilités. L'adéquation aux données est mesurée en tenant compte notamment de la distance entre les points et les primitives du modèle CAO (section 2.3.1), tandis que la conformité aux règles d'assemblage s'évalue en étudiant l'ensemble des connections entre les éléments du modèle (section 2.3.3). La ressemblance au modèle CAO a priori est quant à elle mesurée en comparant chaque forme du modèle reconstruit à l'élément du modèle CAO dont elle semble être le plus proche (section 2.3.2).*

---

## 2.1 Données et connaissances *a priori*

Le problème de reconstruction consiste en la recherche d'un modèle CAO décrivant fidèlement un nuage de points donné. Pour permettre de guider cette reconstruction, nous supposons que nous disposons d'un modèle CAO *a priori* représentant approximativement la scène que l'on cherche à reconstruire. En outre, pour tenir compte de la nature fortement structurée des environnements industriels, de par leur construction, on se donne un ensemble de règles (règles "métier") que l'on souhaite voir satisfaites dans tout modèle CAO représentant une scène industrielle.

Cette section introduit les diverses données et connaissances que nous serons amenées à manipuler par la suite.

### 2.1.1 Nuage de points

Un nuage de points 3D se définit comme une collection désorganisée de points tridimensionnels. On notera  $\mathcal{P}$  le nuage de points échantillonnant la scène à reconstruire:

$$\mathcal{P} = \left\{ \mathbf{p}_0, \dots, \mathbf{p}_{|\mathcal{P}|-1} \right\} \quad \text{avec} \quad \mathbf{p}_i \in \mathbb{R}^3 \quad \forall i \in [0, |\mathcal{P}| - 1] \quad (2.1)$$

Par défaut, on suppose que ce nuage provient d'acquisitions multiples (cf. section 1.2.1), et qu'il est donc désorganisé dans le sens où la liste de points n'obéit pas à un ordonnancement spatial. En pratique, le nuage de points est tout de même associé à une structure de découpage spatial (*octree* ou *KD-Tree* typiquement) permettant un accès rapide aux données qui le composent. En particulier, toute requête concernant les  $k$  plus proches voisins d'un point  $\mathbf{r} \in \mathbb{R}^3$  s'effectue ainsi en  $O(k \log(|\mathcal{P}|))$ .

À tout point  $\mathbf{p} \in \mathcal{P}$ , on associe un vecteur unitaire  $\mathbf{n}_{\mathcal{P}}(\mathbf{p})$  normal à la surface échantillonnée par  $\mathcal{P}$ . Les détails quant à l'obtention des vecteur normaux se trouvent en section 1.2.3.

Le nuage de points  $\mathcal{P}$  est soumis à un bruit dont on suppose *a minima* (faute de modèle de bruit) que l'amplitude peut être évaluée par l'utilisateur. Du fait du processus d'acquisition, la densité des points du nuage est hétérogène, et il peut notamment exister certaines zones souffrant d'une absence totale de données (phénomènes d'occultations). La nature et la provenance des divers défauts présents dans les nuages de points est expliquée en section 1.2.4.

### 2.1.2 Modèle 3D CAO

Nous considérons dans cette thèse des modèles CAO décrits comme des assemblages de primitives géométriques paramétrées, telles que les cylindres, plans, tores, cônes, etc. Les primitives auxquelles nous nous intéressons (cf. annexe A) permettent en effet de décrire la plupart des installations industrielles de manière concise et précise. En plus des primitives géométriques, le modèle intègre les connexions entre les éléments qui le composent (elles peuvent être éventuellement calculées si elles ne sont pas explicitement représentées). Des représentations similaires sont dores et déjà adoptées dans certains outils élaborés de conception de sites industriels (*Plant Design Management System* - PDMS- RealWorks ou SolidWorks par exemple).

### Restriction aux plans, cylindres et tores

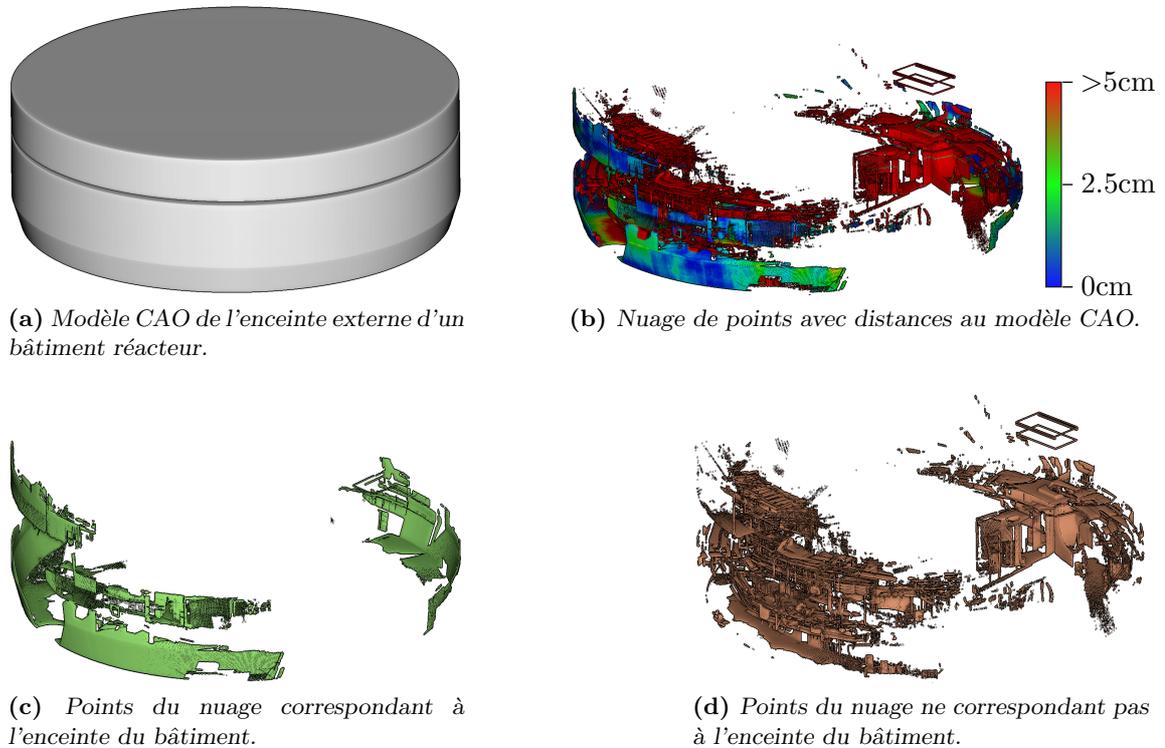
En plus de permettre la description de la plupart des composants de scènes industrielles (cf. tableau 2.1), les plans et cylindres sont fondamentalement structurants : ils permettent d'établir un squelette à partir duquel certains autres composants peuvent être déduits. Typiquement, les tores et cônes représentent pour la plupart des éléments de jonction (respectivement des coudes et réducteurs) reliant des éléments de tuyauterie représentés par des cylindres. Ainsi peut-on raisonnablement supposer que les tores et cônes peuvent n'être traités qu'une fois les cylindres convenablement détectés : il "suffirait" alors de tester toutes les jonctions entre cylindres pour voir si la présence d'une de ces formes est appropriée ou non.

		Boîtes	Cylindres	Tores circulaires	Tores rectangulaires	Cônes	Calottes sphéroïdales	Extrusions
Scène 1	Quantité	1747	1815	576	52	32	14	1026
	Surface (%)	78.6	16.3	2	0.2	0.2	1.3	1.3
Scène 2	Quantité	153	401	201	18	5	11	270
	Surface (%)	6.4	49	0.9	17.5	23.8	0.04	2.4
Scène 3	Quantité	5801	4257	2137	148	92	106	5615
	Surface (%)	31.2	17.9	3.8	0.08	1	2.5	43.5
Scène 4	Quantité	318	735	124	0	6	16	0
	Surface (%)	3.2	74.3	10.3	0	0.9	11.2	0

**Table 2.1:** Constitution en primitives géométriques de modèles CAO de quelques scènes industrielles. Ces statistiques sont établies à partir de la lecture de données PDMS de modèles CAO plus ou moins complets représentant des bâtiments de centrales nucléaires. Elles présentent un biais puisque les plans ne peuvent y pas être stockés en tant que tels. Les parties planaires sont donc noyées dans le décompte des boîtes.

Hormis les représentations très classiques des cylindres, cônes et tores rectangulaires en tant que composants de conduits (tuyauterie et ventilation), il existe quelques utilisations plus exotiques de ces trois types de primitives géométriques. En effet, dans des bâtiments présentant une structure annulaire, ce qui est typiquement le cas du bâtiment réacteur d'une centrale nucléaire, elles peuvent être utilisées pour décrire les murs des différentes enceintes, et peuvent à ce titre représenter une proportion non négligeable des données LIDAR (cf. Scène 2 dans le tableau 2.1). Cependant, on choisira de ne pas tenir compte de ce type d'utilisation pour deux raisons principales. Tout d'abord, il est raisonnable de considérer que les éléments du bâti (en l'occurrence, les murs d'enceinte) sont statiques d'une scène à l'autre : l'enceinte du modèle CAO *a priori* doit donc, selon toute vraisemblance, correspondre rigoureusement à ce qui représente l'enceinte dans les données LIDAR. Il semble donc pertinent, en guise de prétraitement, de laisser l'utilisateur exclure ces données des scènes considérées (modèle *a priori* et nuage de points - cf. figure 2.1). D'autre part, contrairement à la plupart des équipements représentés par le nuage de points, ces éléments contiennent en leur intérieur d'autres composants. À l'exception des structures d'enceinte, l'intérieur des équipements est en général inaccessible à l'acquisition LIDAR.

On se limitera donc aux situations où les primitives considérées sont des volumes ne pouvant contenir d'autres volumes.



**Figure 2.1:** Procédure permettant d'isoler le génie civil supposé commun entre les scènes. Dans un premier temps, on établit en tout point du nuage la distance au modèle CAO. Puis, dans cette carte de distance, on isole les parties inchangées par seuillage statistique (travaux de D. Girardeau-Montaut [GM06]). Ceci permet d'isoler du reste du nuage les parties correspondant effectivement à l'enceinte du bâtiment

A la lumière de ces constats, il semble pertinent de ne s'intéresser dans un premier temps qu'à la détection des cylindres et des plans, et de proposer dans un second temps une solution permettant d'affiner le modèle reconstruit en y intégrant les tores (les cônes, de par leur faible quantité ne semblent présenter qu'un intérêt très faible). D'autant plus que le traitement simultané de plusieurs types de primitives n'est pas sans poser de problèmes dans les approches existantes (cf. figure 1.24 de la section 1.5). La restriction à ces quelques types de primitives peut ainsi permettre de limiter les phénomènes de mise en concurrence des diverses formes, et donc de réduire les erreurs de reconstruction.

### Définition de l'ensemble des modèles CAO

Comme nous le disions précédemment, un modèle CAO est avant tout un ensemble de primitives géométriques paramétrées, parmi lesquelles nous choisissons de ne retenir que les cylindres et les plans dans un premier temps, puis les tores dans un second temps.

Les cylindres tronqués sont définis comme des points dans un espace  $\Psi_C$  à 7 dimensions, tandis que les carreaux rectangulaires plans sont définis dans un espace  $\Psi_P$  à 8 dimensions

et les tores à section circulaire dans un espace  $\Psi_T$  à 9 dimensions (cf. annexe A).

$$\Psi_C = \overbrace{[0, \pi] \times [0, 2\pi[}^{\text{Direction de l'axe } \mathbf{a}_C} \times \overbrace{\mathbb{R}^3}^{\text{Position } \mathbf{c}_C} \times \overbrace{\mathbb{R}}^{\text{Rayon } r_C} \times \overbrace{\mathbb{R}}^{\text{Longueur } l_C} \quad (2.2)$$

$$\Psi_P = \overbrace{[0, \pi] \times [0, 2\pi[}^{\text{Normale } \mathbf{n}_P} \times \overbrace{\mathbb{R}^3}^{\text{Position } \mathbf{c}_P} \times \overbrace{[0, 2\pi[}^{\text{Orientation } \mathbf{d}_P} \times \overbrace{\mathbb{R}}^{\text{Largeur } w_P} \times \overbrace{\mathbb{R}}^{\text{Longueur } l_P} \quad (2.3)$$

$$\Psi_T = \overbrace{[0, \pi] \times [0, 2\pi[}^{\text{Direction de l'axe } \mathbf{a}_T} \times \overbrace{\mathbb{R}^3}^{\text{Position } \mathbf{c}_T} \times \overbrace{[0, 2\pi[}^{\text{Orientation } \mathbf{d}_T} \times \overbrace{\mathbb{R}}^{\text{Rayon } r_T} \times \overbrace{\mathbb{R}}^{\text{Rayon } R_T} \times \overbrace{[0, 2\pi[}^{\text{Angle } \alpha_T} \quad (2.4)$$

L'ensemble des primitives géométriques pouvant s'assembler en un modèle CAO se définit donc simplement :

$$\Psi = \Psi_C \cup \Psi_P \cup \Psi_T \quad (2.5)$$

Un modèle CAO est un ensemble de primitives géométriques: une même primitive géométrique ne peut (en théorie) y figurer plusieurs fois, et l'ordre des éléments qui composent le modèle n'importe pas. L'espace des modèles CAO se définit comme l'ensemble des ensembles de primitives géométriques, c'est-à-dire l'ensemble des parties de  $\Psi$ :

$$\Omega = \mathcal{P}(\Psi) = \{\mathcal{M} | \mathcal{M} \subset \Psi\} \quad (2.6)$$

### Modèle CAO reconstruit / *a priori*

On distinguera par la suite deux modèles CAO, satisfaisant tous deux la description faite précédemment:

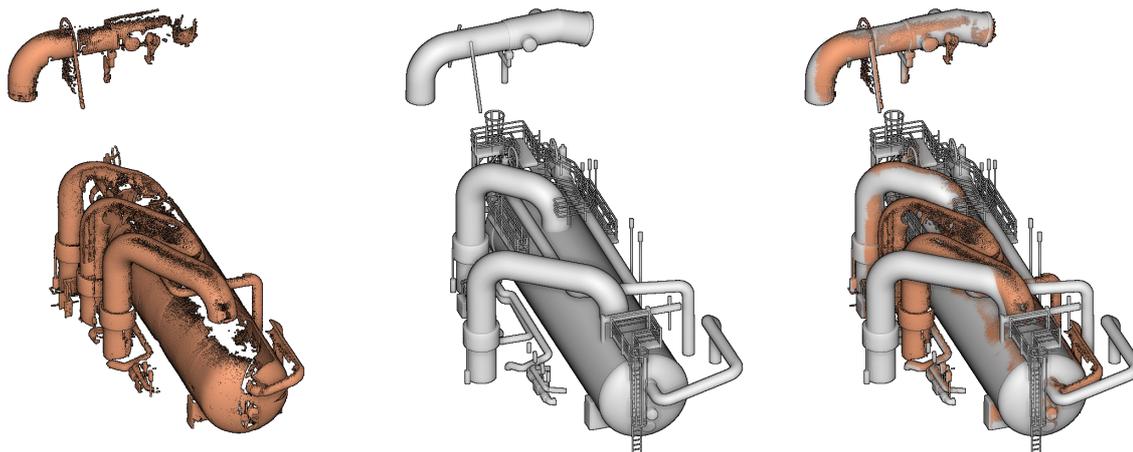
**le modèle CAO reconstruit**, que l'on notera  $\mathcal{X} = \{\mathcal{S}_0, \dots, \mathcal{S}_{|\mathcal{X}|-1}\}$ , est issu des calculs visant à reconstruire la scène représentée par le nuage de points.

**le modèle CAO *a priori***, que l'on notera  $\mathcal{M}_0 = \{\mathcal{S}^0, \dots, \mathcal{S}^{|\mathcal{M}_0|-1}\}$ , est un modèle existant supposé représenter approximativement la scène à reconstruire.

On notera la distinction dans les notations: alors que les primitives reconstruites  $\mathcal{S}_i \in \mathcal{X}$  sont représentées à l'aide d'indices, les primitives *a priori*  $\mathcal{S}^j \in \mathcal{M}_0$  sont représentées à l'aide d'exposants. Quoiqu'il en soit,  $\mathcal{X}$  et  $\mathcal{M}_0$  sont tous deux des éléments de  $\Omega$ , ce qui implique qu'ils sont tous deux composés uniquement de plans, cylindres et tores. Par construction,  $\mathcal{X}$  vérifie nécessairement cette affirmation puisqu'on ne choisit de ne traiter que ces trois types de primitives. Concernant  $\mathcal{M}_0$ , on supposera qu'il aura été filtré au préalable afin de ne retenir que les plans, cylindres et tores.

Le modèle CAO *a priori* doit représenter la scène étudiée, sans être forcément rigoureusement équivalent. On supposera que les parties communes aux scènes représentées par  $\mathcal{P}$  et  $\mathcal{M}_0$  sont identiques, bien qu'il puisse exister des différences potentiellement significatives entre les deux environnements traités. Certains équipements peuvent par exemple ne figurer que dans une des deux scènes, d'autres peuvent se trouver déplacés, les rayons apparents d'une tuyauterie peuvent être différents, etc.

Le nuage de points  $\mathcal{P}$  et le modèle *a priori*  $\mathcal{M}_0$  sont initialement recalés. Ce recalage initial peut être induit par l'expression des deux scènes dans un référentiel commun, propre à chaque site industriel étudié. L'origine et l'orientation du repère dans les centrales nucléaires, par exemple, sont standardisées, de sorte que deux mêmes bâtiments considérés sur deux sites distincts sont bien recalés s'ils sont exprimés dans ce repère. Si les informations de passage vers ce repère ne sont pas disponibles, l'utilisateur pourra recalculer



**Figure 2.2:** Comparaison d'un modèle a priori  $\mathcal{M}_0$  (milieu) et du nuage de points  $\mathcal{P}$  (gauche) de la scène étudiée. On constate des différences de position et d'orientation de tuyaux, ainsi que l'absence dans  $\mathcal{M}_0$  du tuyau central connecté à la cuve. Il existe en outre une erreur de recalage de l'ordre de 5 à 10 centimètres (il s'agit en fait d'un défaut angulaire de quelques degrés) que l'on peut constater en étudiant finement la superposition de  $\mathcal{P}$  et  $\mathcal{M}_0$  au niveau de la cuve supposée identique dans les deux environnements.

grossièrement les deux scènes via un outil interactif quelconque à base de sélection de points communs entre les deux scènes [Bos10].

Outre les différences de nature géométrique pouvant exister entre les deux scènes considérées, la mise en correspondance entre les deux scènes représentées par  $\mathcal{P}$  et  $\mathcal{M}_0$  doit tenir compte du fait que tous les éléments figurant dans un environnement ne se trouvent pas nécessairement dans le second. Il peut par exemple exister des tuyauteries qui, pour diverses raisons, ne se trouvent que dans une des deux scènes (cf. figure 2.2). La comparaison entre les deux scènes ne repose donc pas sur une "simple" bijection entre les primitives géométriques qui les composent, et la reconstruction guidée par le modèle *a priori* ne peut être traitée comme un problème de recalage. Il s'agit d'un problème plus complexe, se posant plutôt comme la recherche dans le nuage de points  $\mathcal{P}$  des primitives qui correspondent, à des changements géométriques près, à des primitives de  $\mathcal{M}_0$ .

### 2.1.3 Connaissances métier

Afin d'obtenir un modèle CAO fiable, il est important de tenir compte d'un certain nombre de règles qui régissent leur composition ainsi que l'assemblage des éléments en une description cohérente de la scène. Ces règles sont liées à la construction du site. Elles peuvent être issues de pratiques évidentes inhérentes à la nature des composants manufacturés (par exemple les axes concourants des éléments de tuyauterie), ou de connaissances expertes spécifiques à la scène étudiée (diamètres nominaux des lignes de tuyauterie, angles nominaux des coudes).

Ces règles n'ont pas forcément une valeur universelle : il peut exister des situations où elles se voient enfreintes dans les modèles CAO existants. Elles décrivent toutefois la majorité des situations, et on considérera qu'il est souhaitable que le modèle CAO reconstruit les satisfasse dans la mesure du possible.

Comme évoqué précédemment, nous avons fait le choix de nous concentrer exclusivement sur les cylindres, tores et plans. Les cylindres et tores servent essentiellement à la description des éléments de tuyauterie, et les premiers peuvent parfois représenter de gros composants (cuves). Quant aux plans, ils représentent les sols, plafonds et murs, ou divers

types de boîtiers électriques. Voici la liste décrivant les contraintes observables sur ces composants en environnement industriel que nous retenons dans le cadre de nos travaux:

- les composants de tuyauterie connexes possèdent généralement des axes concourants. Il n'est pas nécessaire que les diamètres de tuyaux soient tous égaux sur une même ligne de tuyauterie (piquages, recours aux réducteurs).
- les composants de tuyauterie sont produits selon des nomenclatures. À ce titre, leurs paramètres (angle, diamètre, longueur) sont régis par des valeurs nominales connues. Toutefois, le diamètre observable des lignes de tuyauterie peut être altéré par des couches d'isolation (calorifuges, plomb), qui elles ne sont pas standardisées.
- les sols et plafonds sont horizontaux, et les parois sont verticales. Ce constat n'est pas rigoureusement exact puisque certains sols doivent être légèrement inclinés pour permettre l'évacuation des eaux, mais c'est une bonne approximation. Les composants planaires connexes sont donc (quasi)perpendiculaires. On constate aussi cette contrainte pour les boîtiers, qui peuvent être décrits comme des assemblages de 6 plans orthogonaux deux à deux.
- les composants de tuyauterie sont principalement orientés selon la direction verticale ou dans un plan horizontal. Encore une fois, il peut exister un très léger biais puisque certains tuyaux ne sont pas rigoureusement horizontaux, pour permettre la vidange des fluides. Il découle de ce constat que, pour une grande partie des cas, deux éléments rectilignes de tuyauterie connexes sont perpendiculaires ou parallèles (voire colinéaires).
- les tuyaux peuvent traverser les murs, le sol, ou le plafond. Dans ces situations, l'axe du tuyau est le plus souvent perpendiculaire à la surface planaire traversée.

Ces quelques constats, illustrés par la figure 2.3, constitueront par la suite la base des règles que l'on se donnera pour définir la qualité intrinsèque d'un modèle CAO reconstruit vis-à-vis des connaissances métier.

## 2.2 Formulation probabiliste du problème

### 2.2.1 Formalisme Bayésien

#### Introduction à la recherche du Maximum A Posteriori (MAP)

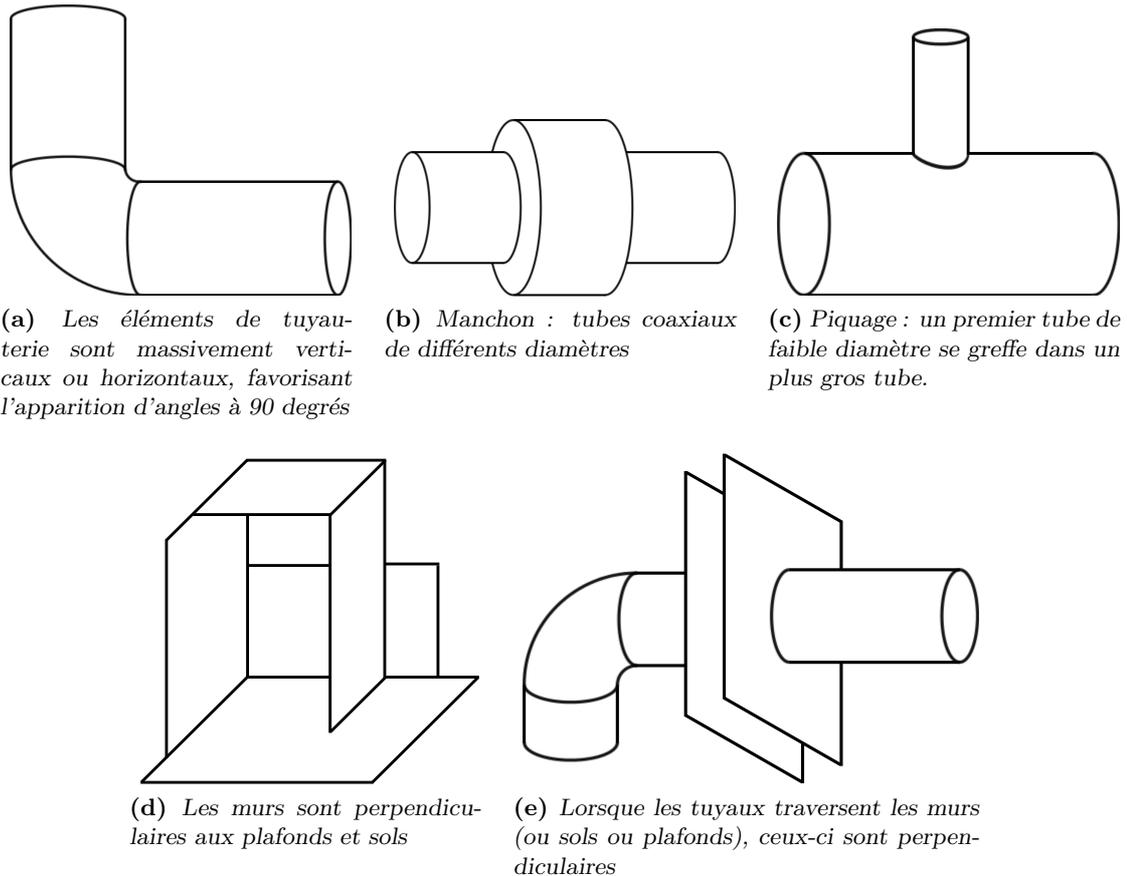
Le formalisme Bayésien est le principal outil permettant de tirer parti de la disponibilité d'informations/données *a priori*. La méthodologie repose sur la célèbre formule de Bayes issue de la théorie des probabilités. Cette équation stipule que la probabilité d'un événement  $A$  conditionnée par la réalisation d'un événement  $B$  peut être calculée à partir de la probabilité de  $B$  conditionnée par la réalisation de  $A$ . Cette formule permet donc d'inverser, en terme de probabilités, les causes et les effets.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \quad (2.7)$$

Usuellement, on utilise les dénominations suivantes pour les termes de l'équation 2.7 :

$P(\mathbf{A}|\mathbf{B})$  : probabilité *a posteriori*

$P(\mathbf{B}|\mathbf{A})$  : vraisemblance de  $A$



**Figure 2.3:** Illustration des situations majoritairement rencontrées dans les modèles CAO d'environnement industriels, permettant de définir les connaissances métier.

$\mathbf{P(A)}$  : probabilité *a priori*

$\mathbf{P(B)}$  : probabilité marginale de  $B$  (terme de normalisation)

On se donne deux variables aléatoires  $X$  et  $Y$  définies sur deux univers quelconques, et l'événement  $A$  correspond à l'affectation d'une valeur à la variable  $X$ , tandis que l'événement  $B$  correspond à l'affectation d'une valeur à la variable  $Y$ .

Dans une des utilisations courantes de la formule de Bayes,  $X$  constitue l'inconnue, c'est-à-dire le modèle (au sens général) que l'on cherche à déterminer. Quant à  $Y$ , elle représente les observations: ce sont les données  $Y = y_0$  dont on dispose en entrée. On cherche alors à trouver l'instance  $X = x^*$  qui soit la plus probable *a posteriori*, c'est-à-dire qui soit la plus probable à la fois vis-à-vis des observations dont on dispose ainsi que vis-à-vis d'un certain nombre de critères purement *a priori*. L'équation 2.7 formalise exactement cette explication très intuitive: pour maximiser la probabilité *a posteriori*, il faut maximiser le produit de la vraisemblance, définissant l'attache du modèle  $X$  aux observations  $y_0$ , et de la probabilité *a priori*. Les observations étant invariantes quel que soit l'état de  $X$ , le terme de normalisation  $P(B)$  de l'équation 2.7 est constant et n'intervient donc pas dans le processus d'optimisation. Il peut donc être ignoré.

Cette démarche est dénommée recherche du Maximum A Posteriori (MAP), et consiste donc en la recherche de l'instance optimale  $X = x^*$  au sens des probabilités définies pour

modéliser le problème à résoudre:

$$\begin{aligned} x^* &= \operatorname{argmax}_x P(X = x | Y = Y_0) \\ &= \operatorname{argmax}_x P(Y = y_0 | X = x) P(X = x) \end{aligned} \quad (2.8)$$

Son intérêt majeur tient au fait qu'elle permet d'intégrer dans la modélisation du problème une mesure de la qualité intrinsèque d'une instance via le terme *a priori*.

### Quelques exemples d'application

Les travaux de J.R. Diebel et al. [DTB06] constituent une excellente référence en matière d'application du formalisme Bayésien pour la reconstruction de surfaces (modèle) à partir de nuages de points (observations). Le problème consiste en la recherche d'un maillage qui décrive fidèlement le nuage de points, tout en respectant un certain nombre d'attentes *a priori*. Tout d'abord, le maillage doit être, lorsque c'est nécessaire, aussi lisse que possible afin d'estomper l'aspect disgracieux induit par le bruit du nuage de points. En outre, les arêtes vives doivent être respectées. Toute la surface n'est pas forcément lisse, et il est donc important que la géométrie des observations soit conservée.

Les auteurs proposent donc une mesure de vraisemblance du maillage reconstruit basée sur une estimation de la distance entre le nuage et le maillage: plus les sommets du maillage s'éloignent du point avec lequel il sont appariés (le plus proche selon la distance de Mahalanobis), plus la vraisemblance du maillage diminue. Les auteurs intègrent à cette distance l'incertitude supposée Gaussienne sur la position des points du nuage. Quant à la probabilité *a priori*, elle repose sur une fonction de la courbure du maillage, estimée grâce à la dispersion des normales. Cette fonction très simple tend à favoriser les surfaces de faible courbure, tout en valorisant les angles de plus de 90°.

La recherche de la solution optimale est réalisée à l'aide d'un algorithme de gradients conjugués, ce qui est rendu possible par les bonnes propriétés différentielles des probabilités définies. Les résultats présentés sont d'excellente qualité. La méthode proposée permet effectivement d'obtenir des surfaces fidèles aux données, dont les défauts sont gommés tout en préservant les arêtes vives.

Dans le même article, les auteurs proposent une seconde méthode Bayésienne, pour la décimation des maillages cette fois-ci. Il s'agit de trouver un maillage le plus compact possible (contenant moins de sommets) tout en étant visuellement proche du maillage d'origine. Pour ce problème, la vraisemblance du maillage valorise la préservation des normales lors de la décimation, afin de garantir l'équivalence visuelle du maillage décimé et du maillage d'origine. Quant à la probabilité *a priori*, elle est simplement fonction du nombre de sommets du maillage, et favorise les maillages compacts. L'optimisation se pose ici comme un problème discret. Les auteurs proposent de calculer la solution optimale en alternant des fusions de faces du maillage et en gérant l'acceptation de chaque solution de manière gloutonne.

P. Jenke et al. [JWB+06] proposent une approche équivalente pour la reconstruction de maillages lisses à partir de nuages de points, avec préservation des arêtes vives. Les auteurs introduisent en plus un *a priori* portant sur la densité des points sur la surface scannée.

### 2.2.2 Application au cas de la reconstruction

#### Formulation probabiliste

Dans le cadre de cette thèse, il semble pertinent de formuler le problème de la reconstruction comme une recherche de maximum *a posteriori*. Il s'agit en effet de trouver le modèle CAO reconstruit  $\mathcal{X}^*$  qui soit le plus pertinent à la fois vis-à-vis du nuage de points (observations), des connaissances *a priori* et du modèle CAO *a priori*. Cette pertinence peut être quantifiée, pour tout modèle CAO  $\mathcal{X}$ , par une probabilité *a posteriori* proportionnelle au produit de la vraisemblance du modèle CAO et d'une probabilité *a priori*:

$$P(X = \mathcal{X} | Y = \mathcal{P}) = \frac{P(Y = \mathcal{P} | X = \mathcal{X})P(X = \mathcal{X})}{P(Y = \mathcal{P})}$$

La vraisemblance  $P(Y = \mathcal{P} | X = \mathcal{X})$  mesure la qualité de la description du modèle CAO  $\mathcal{X}$  par le nuage de points  $\mathcal{P}$ , "l'attache aux données".

Dans nos travaux, la probabilité *a priori*  $P(X = \mathcal{X})$  intègre deux critères: la qualité de l'assemblage des formes qui composent le modèle CAO  $\mathcal{X}$ , d'une part, et la ressemblance au modèle CAO *a priori*  $\mathcal{M}_0$ , d'autre part. On peut ainsi définir, vis-à-vis de chacune de ces deux attentes, une probabilité de voir survenir l'événement  $X = \mathcal{X}$ . Comme ces deux critères sont indépendants, on peut définir la probabilité *a priori* comme étant le produit des probabilités définies pour chaque critère:

$$P(X = \mathcal{X}) = P_G(X = \mathcal{X})P_T(X = \mathcal{X})$$

$P_G$  estime la qualité "géométrique" de  $\mathcal{X}$ , définie par rapport au modèle *a priori*: un modèle reconstruit est satisfaisant (donc probable) s'il ressemble au modèle *a priori*  $\mathcal{M}_0$ , autrement dit, s'il lui est géométriquement équivalent. Quant à  $P_T$ , elle estime la qualité "topologique" de  $\mathcal{X}$ : un modèle reconstruit est satisfaisant (donc probable) s'il satisfait les règles d'assemblage des éléments qui le composent, autrement dit s'il est topologiquement correct.

Finalement, nous définissons donc la probabilité *a posteriori* d'un modèle CAO  $\mathcal{X}$  comme étant:

$$P(X = \mathcal{X} | Y = \mathcal{P}) = \frac{P(Y = \mathcal{P} | X = \mathcal{X})P_G(X = \mathcal{X})P_T(X = \mathcal{X})}{P(Y = \mathcal{P})} \quad (2.9)$$

D'où la formulation du problème de reconstruction comme la maximisation *a posteriori*:

$$\boxed{\mathcal{X}^* = \operatorname{argmax}_{\mathcal{X} \in \Omega} P(Y = \mathcal{P} | X = \mathcal{X})P_G(X = \mathcal{X})P_T(X = \mathcal{X})} \quad (2.10)$$

puisque l'on peut ignorer le terme de normalisation constant  $P(Y = \mathcal{P})$ .  $\Omega$  désigne l'ensemble des modèles CAO réalisables (cf. section 2.1.2).

#### Formulation énergétique

Nous choisissons d'exprimer les probabilités sous une forme commune:

$$P(Y = \mathcal{P} | X = \mathcal{X}) = \frac{e^{-\lambda_D H_D(\mathcal{X}, \mathcal{P})}}{\int_{\mathcal{Z} \in \Omega} e^{-\lambda_D H_D(\mathcal{Z}, \mathcal{P})}} \quad (2.11)$$

$$P_G(X = \mathcal{X}) = \frac{e^{-\lambda_G H_G(\mathcal{X}, \mathcal{M}_0)}}{\int_{\mathcal{Z} \in \Omega} e^{-\lambda_G H_G(\mathcal{Z}, \mathcal{M}_0)}} \quad (2.12)$$

$$P_T(X = \mathcal{X}) = \frac{e^{-\lambda_T H_T(\mathcal{X})}}{\int_{\mathcal{Z} \in \Omega} e^{-\lambda_T H_T(\mathcal{Z})}} \quad (2.13)$$

où les facteurs  $\lambda_D$ ,  $\lambda_G$  et  $\lambda_T$ , tous trois positifs, permettent de moduler l'importance relative de chaque terme.

La fonction exponentielle inverse étant non négative, elle est fréquemment utilisée pour définir des probabilités. Sa tendance à décroître rapidement est de plus une propriété très intéressante: moins  $\mathcal{X}$  est pertinent, plus sa probabilité *a posteriori* tend vers 0.

Les "énergies"  $H_D$ ,  $H_G$  et  $H_T$  sont des fonctions décroissantes de la pertinence du modèle CAO  $\mathcal{X}$  vis-à-vis de chaque critère concerné:

- $H_D$  est l'énergie d'attache aux données. Elle diminue lorsque  $\mathcal{X}$  s'ajuste sur le nuage de points  $\mathcal{P}$ .
- $H_G$  est l'énergie géométrique. Elle diminue lorsque  $\mathcal{X}$  se rapproche du modèle *a priori*  $\mathcal{M}_0$ .
- $H_T$  est l'énergie topologique. Elle diminue lorsque les formes constituant  $\mathcal{X}$  s'assemblent de manière cohérente.

Le problème de maximisation *a posteriori* peut être reformulé en un problème de minimisation d'énergie, en considérant le logarithme de la probabilité *a posteriori*:

$$\begin{aligned} \operatorname{argmax}_{\mathcal{X} \in \Omega} P(X = \mathcal{X} | Y = \mathcal{P}) &= \operatorname{argmax}_{\mathcal{X} \in \Omega} \ln(P(X = \mathcal{X} | Y = \mathcal{P})) \\ &= \operatorname{argmax}_{\mathcal{X} \in \Omega} \ln \left( \frac{e^{-\lambda_D H_D(\mathcal{X}, \mathcal{P})} e^{-\lambda_G H_G(\mathcal{X}, \mathcal{M}_0)} e^{-\lambda_T H_T(\mathcal{X})}}{\omega} \right) \end{aligned}$$

avec

$$\omega = \int_{\mathcal{Z} \in \Omega} e^{-\lambda_D H_D(\mathcal{Z}, \mathcal{P})} \int_{\mathcal{Z} \in \Omega} e^{-\lambda_G H_G(\mathcal{Z}, \mathcal{M}_0)} \int_{\mathcal{Z} \in \Omega} e^{-\lambda_T H_T(\mathcal{Z})}$$

Comme  $\omega$  est une constante, elle n'intervient pas dans le processus d'optimisation et peut donc être ignorée.

$$\begin{aligned} \operatorname{argmax}_{\mathcal{X} \in \Omega} P(X = \mathcal{X} | Y = \mathcal{P}) &= \operatorname{argmax}_{\mathcal{X} \in \Omega} \ln \left( e^{-\lambda_D H_D(\mathcal{X}, \mathcal{P}) - \lambda_G H_G(\mathcal{X}, \mathcal{M}_0) - \lambda_T H_T(\mathcal{X})} \right) \\ &= \operatorname{argmax}_{\mathcal{X} \in \Omega} -(\lambda_D H_D(\mathcal{X}, \mathcal{P}) + \lambda_G H_G(\mathcal{X}, \mathcal{M}_0) + \lambda_T H_T(\mathcal{X})) \end{aligned}$$

Rechercher le maximum *a posteriori*  $\mathcal{X}^*$ , c'est donc rechercher la configuration qui minimise l'énergie globale  $H$ :

$$\boxed{\mathcal{X}^* = \operatorname{argmax}_{\mathcal{X} \in \Omega} P(X = \mathcal{X} | Y = \mathcal{P}) = \operatorname{argmin}_{\mathcal{X} \in \Omega} H(\mathcal{X}, \mathcal{P}, \mathcal{M}_0)} \quad (2.14)$$

où  $H$  s'exprime simplement comme la combinaison linéaire des 3 énergies mentionnées précédemment:

$$H(\mathcal{X}, \mathcal{P}, \mathcal{M}_0) = \lambda_D H_D(\mathcal{X}, \mathcal{P}) + \lambda_G H_G(\mathcal{X}, \mathcal{M}_0) + \lambda_T H_T(\mathcal{X}) \quad (2.15)$$

Cette reformulation permet de rendre plus stable la résolution numérique du problème. En effet, du fait de la représentation discrète des nombres dans les ordinateurs, il est préférable de manipuler des sommes de nombres définis sur  $\mathbb{R}$  plutôt que des multiplications de nombres définis sur  $[0, 1]$ . Non seulement la somme permet de limiter la propagation des erreurs, mais le fait que les quantités manipulées ne soit pas restreintes à un sous-ensemble des nombres représentables permet de tirer entièrement parti des capacités de représentation, diminuant ainsi l'impact des erreurs d'arrondis sur les nombres stockés en machine. En pratique, la formulation à base d'énergies (équation 2.14) est donc nettement préférable à la formulation probabiliste (équation 2.10).

## 2.3 Proposition d'énergies

Pour modéliser le problème de reconstruction, nous devons donc définir des énergies qui vérifient la propriété mentionnée précédemment: elle doivent décroître lorsque le modèle tend à satisfaire les attentes du problème. Cette section décrit les énergies que nous proposons à cette fin.

### 2.3.1 Attache aux données

L'énergie d'attache aux données  $H_D(\mathcal{X}, \mathcal{P})$  permet d'estimer la qualité de la description du modèle CAO  $\mathcal{X}$  par le nuage de points  $\mathcal{P}$ . Plus le nuage de points est conforme au modèle, plus cette énergie diminue.

### Mesures existantes

Il existe diverses approches permettant de quantifier la qualité de l'ajustement d'un modèle 3D sur un nuage de points. On peut décomposer ces approches en deux familles: une première où le modèle 3D est confronté à l'ensemble du nuage de points, sans qu'on ne puisse dire *a priori* si un point correspond au modèle 3D ou non (ce qui est notre cas), et une seconde où l'ensemble des points décrivant le modèle 3D est initialement identifié.

Considérons tout d'abord la première situation. Dans l'algorithme RANSAC, chaque primitive composant le modèle 3D reconstruit se voit affecter un score positif correspondant au nombre de points qui décrivent la forme (de tels points sont appelés *inliers*). Le critère utilisé pour décider si un point décrit la forme peut être plus ou moins élaboré. L'approche originale [FB87] considère simplement qu'un *inlier* est un point dont la distance à une primitive est inférieure à un seuil lié au bruit dans le nuage de points. Cette mesure triviale peut être légèrement améliorée en considérant de plus les normales dans le nuage de points. Dans les travaux de R. Schnabel et al. [SWK07], un *inlier*  $\mathbf{p}$  doit non seulement être suffisamment proche d'une primitive  $\mathcal{S}$ , mais l'angle entre le vecteur normal associé à  $\mathbf{p}$  et la normale théorique à  $\mathcal{S}$  en  $\mathbf{p}$  doit être inférieur à un second seuil.

La nature discrète de cette mesure, puisqu'il s'agit d'un simple décompte de points, peut sembler insuffisante pour permettre l'ajustement fin de modèles 3D sur un nuage de points. Le score proposé pour l'algorithme MSAC par P. Torr et A. Zisserman [TZ00] introduit une nuance parmi les *inliers*, en modulant leur importance en fonction (quadratique) de leur distance à la primitive: les points les plus proches contribuent significativement à l'amélioration du score, tandis que les plus éloignés ne contribuent que faiblement. Au-delà d'un certain seuil, la contribution est constante. Les auteurs insistent sur la supériorité de

cette mesure en précisant qu'elle peut être évaluée exactement au même coût que le score du RANSAC original.

T. Ullrich et al. [USF08] utilisent une mesure similaire à celle de l'algorithme MSAC, basée sur la composition de distance et de fonction gaussienne. Cette mesure correspond à un équivalent continument dérivable de la mesure du MSAC qui, elle, n'est pas dérivable. Cette fonction possède donc de meilleures propriétés pour l'affinement d'ajustement de modèles 3D sur les nuages de points.

Supposons maintenant connu l'ensemble des points  $\mathcal{Q}$  correspondant à une surface donnée  $S$ . L'ajustement de la surface sur ces points est communément quantifiée par la mesure d'erreur moyenne quadratique (MSE pour *Mean Squared Error* en anglais), ou sa racine carrée (RMSE pour *Root Mean Squared Error*). Cette mesure issue des statistiques correspond à l'écart moyen entre les données observées et les données attendues. Concrètement, elle s'exprime comme la somme des carrés des erreurs, où l'erreur dans le cas de l'ajustement de surface est usuellement la distance euclidienne  $d$  séparant un point de la surface (on parle alors d'erreur "géométrique"):

$$MSE(S, \mathcal{Q}) = \frac{1}{|\mathcal{Q}|} \sum_{\mathbf{p}_i \in \mathcal{Q}} d(S, \mathbf{p}_i)^2 \quad (2.16)$$

Dans certaines variantes, les carrés des distances sont pondérés par une incertitude portant généralement sur la position de chaque point. Dans tous les cas, plus l'erreur moyenne quadratique est faible (notons au passage qu'elle est positive), plus le modèle 3D est pertinent vis-à-vis du nuage de points.

Bien qu'elle soit moins utilisée, l'erreur algébrique [AS05] constitue une alternative à l'erreur géométrique. L'erreur algébrique se fonde sur l'expression mathématique des objets 3D considérés. Supposons que l'on dispose d'une expression implicite de la surface de  $S$ , qui correspond donc à l'ensemble des points de  $\mathbb{R}^3$  vérifiant une équation de la forme:

$$f_S(x, y, z) = 0 \quad x, y, z \in \mathbb{R}$$

ce qui est typiquement le cas des quadriques, dont font partie les cylindres, cônes et ellipsoïdes. La distance algébrique d'un point  $\mathbf{p}_i = (x_i, y_i, z_i)^T$  à la surface  $S$  est alors simplement la quantité  $f_S(x_i, y_i, z_i)$ . Cette mesure peut alors être substituée à  $d$  dans l'expression 2.16 pour définir une erreur moyenne quadratique algébrique.

Quoiqu'il en soit, l'inconvénient majeur de l'erreur moyenne quadratique réside dans sa faible robustesse face aux points aberrants. Tout point ne correspondant manifestement pas au processus d'échantillonnage de la primitive possède une influence considérable sur l'erreur moyenne quadratique, et peut donc fausser à lui seul l'évaluation de l'ajustement. La fiabilité de l'ensemble  $\mathcal{Q}$  est donc cruciale pour la pertinence de la mesure MSE.

### Notre mesure

L'utilisation courante d'une fonction quadratique pour permettre l'ajustement fin du modèle sur le nuage de points nous conduit à choisir une fonctionnelle de ce type pour notre application. Dans le voisinage proche d'une primitive géométrique, les points du nuage apportent donc une contribution qui est une fonction quadratique de leur distance à la primitive. De plus, l'intégration des vecteurs normaux dans les calculs permet de moduler l'apport des points qui ne semblent pas décrire convenablement la surface proposée: plus la normale observée (dans le nuage) est proche de la normale théorique (à la surface de la

primitive), meilleur est l'ajustement.

Cependant, il nous semble pertinent de pousser plus loin le raisonnement. Beaucoup d'auteurs considèrent effectivement que, pour être bien décrit par le nuage de points, le modèle 3D doit "simplement" maximiser la quantité d'*inliers*, tout en faisant en sorte que ces *inliers* soient aussi proches que possible de sa surface. En ce qui nous concerne, nous pensons que ce n'est pas suffisant. En effet, on s'attend légitimement à trouver des points répartis sur la surface du modèle. Mais on devrait aussi s'attendre à ne pas trouver de points aux endroits où il ne devrait pas y en avoir. Il nous semble donc pertinent de pénaliser les points se situant dans des "zones de vide" attendues.

Si nous considérons le processus d'acquisition du nuage de points, ainsi que la répartition spatiale des primitives dans les modèles CAO, il apparaît que les points se situent majoritairement dans le voisinage immédiat des surfaces scannées, et que très peu de points ne devraient se situer dans un voisinage plus éloigné (qu'on nommera voisinage moyen) de ces surfaces, au-delà de ce que nous avons qualifié jusqu'alors de voisinage immédiat.

Ce raisonnement peut être étendu à l'étude des normales associées aux points. En effet, au-delà d'une certaine tolérance, les normales ne décrivant pas convenablement la forme ajustée doivent conduire à pénaliser l'ajustement.

Il faut toutefois prendre garde à ne pas confondre les zones où l'on s'attend à ne pas trouver de points, et celles qui ne sont soumises à aucune exigence. En effet, il se peut que le modèle CAO reconstruit  $\mathcal{X}$  ne soit que partiel:  $\mathcal{X}$  n'a pas forcément vocation à décrire la totalité du nuage de points  $\mathcal{P}$ . Il peut donc exister une quantité significative de points qui ne sont ni pertinents, ni impertinents, mais dont on ne peut rien conclure. On considérera donc que tout point qui n'est pas dans le voisinage immédiat ni dans le voisinage moyen d'une primitive appartient à cette dernière catégorie. En effet, les points les plus éloignés du modèle 3D évalué ne sont pas soumis à son influence, mais peuvent tout à fait décrire des surfaces cohérentes que l'on ne cherche pas à reconstruire. Il n'est donc pas approprié d'évaluer de tels points.

Pour résumer, le nuage de points  $\mathcal{P}$  se divise en trois catégories de points vis-à-vis d'une primitive  $\mathcal{S}$ :

- les *inliers*, qui sont pertinents vis-à-vis de  $\mathcal{S}$ . Il s'agit des points appartenant au voisinage immédiat de  $\mathcal{S}$  et dont le vecteur normal associé correspond à  $\mathcal{S}$ . Ces points doivent contribuer à la diminution de l'énergie  $H_D$ .
- les *outliers*, qui sont non pertinents par rapport à  $\mathcal{S}$ . Cette catégorie regroupe les points appartenant au voisinage moyen de  $\mathcal{S}$  ainsi que les points dont le vecteur normal associé ne correspond pas à  $\mathcal{S}$ . Ces points doivent contribuer à l'augmentation de l'énergie  $H_D$ .
- les points neutres, n'apportant aucune information vis-à-vis de  $\mathcal{S}$  puisqu'ils en sont trop éloignés. Ces points ne doivent pas contribuer à l'évolution de l'énergie  $H_D$ .

Finalement, nous choisissons d'exprimer l'énergie d'attache aux données comme la somme de contributions  $h_D$  apportées par chaque point du nuage  $\mathcal{P}$ . Nous ajoutons de plus un terme de contrôle  $h_C$  permettant de s'assurer que chaque primitive du modèle est bien décrite par une quantité minimale de points du nuage:

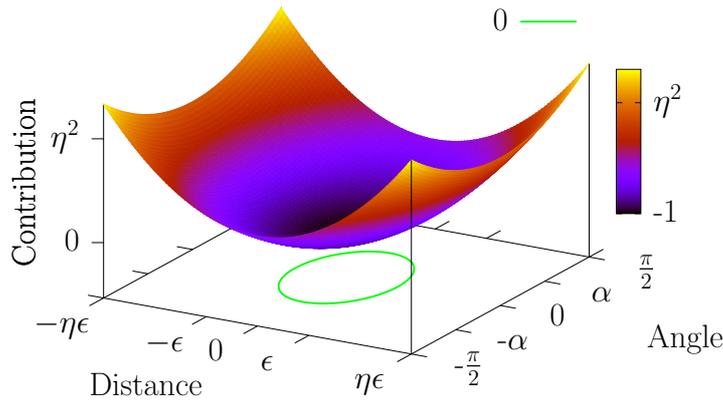
$$H(\mathcal{X}, \mathcal{P}) = \sum_{\mathbf{p}_i \in \mathcal{P}} h_D(\mathcal{X}, \mathbf{p}_i) + \sum_{\mathcal{S} \in \mathcal{X}} h_C(\mathcal{S}, \mathcal{P}) \quad (2.17)$$

**Contribution des points** La contribution  $h_D$  est une parabolôïde tronquée qui valorise les *inliers*, pénalise les *outliers* et est nulle partout ailleurs (cf. figure 2.4):

$$h_D(\mathcal{X}, \mathbf{p}) = \begin{cases} \left( \frac{d(\mathcal{S}_p^{\mathcal{X}}, \mathbf{p})}{\epsilon} \right)^2 + \left( \frac{a(\mathcal{S}_p^{\mathcal{X}}, \mathbf{p})}{\alpha} \right)^2 - 1 & \text{si } d(\mathcal{S}_p^{\mathcal{X}}, \mathbf{p}) \leq \eta\epsilon \\ 0 & \text{sinon} \end{cases} \quad (2.18)$$

où  $\mathcal{S}_p^{\mathcal{X}}$  est la primitive de  $\mathcal{X}$  la plus proche du point  $\mathbf{p}$ :

$$\mathcal{S}_p^{\mathcal{X}} = \operatorname{argmin}_{\mathcal{S} \in \mathcal{X}} d(\mathcal{S}, \mathbf{p})$$



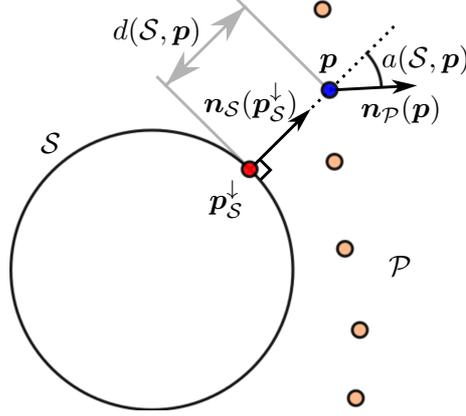
**Figure 2.4:** Graphe de la contribution d'un point  $\mathbf{p}$  en fonction de sa distance à la primitive de  $\mathcal{X}$  dont il est le plus proche et de l'angle entre la normale qui lui est associée et la normale théorique. La tache sombre au centre de la carte de couleur illustre les contributions négatives. La courbe d'isovaleur des contributions nulles est projeté sur le plan horizontal. Bien que ça ne soit pas illustré ici, la contribution est nulle sur le reste de l'espace.

$d(\mathcal{S}, \mathbf{p})$  est la distance séparant la primitive  $\mathcal{S}$  du point  $\mathbf{p}$ , et  $a(\mathcal{S}, \mathbf{p})$  est l'angle entre la normale  $\mathbf{n}_{\mathcal{P}}(\mathbf{p})$  au point  $\mathbf{p}$  dans le nuage de points  $\mathcal{P}$  et la normale  $\mathbf{n}_{\mathcal{S}}(\mathbf{p}_S^\downarrow)$  à la surface de  $\mathcal{S}$  au point  $\mathbf{p}_S^\downarrow$  où  $\mathbf{p}$  se projette sur  $\mathcal{S}$  (cf. figure 2.5):

$$a(\mathcal{S}, \mathbf{p}) = \arccos\left(\left| \mathbf{n}_{\mathcal{P}}(\mathbf{p}) \cdot \mathbf{n}_{\mathcal{S}}(\mathbf{p}_S^\downarrow) \right| \right) \quad (2.19)$$

Les expressions de calcul de  $d(\mathcal{S}, \mathbf{p})$ ,  $\mathbf{p}_S^\downarrow$  et  $\mathbf{n}_{\mathcal{S}}(\mathbf{q})$  pourront être trouvées en annexe A pour les primitives qui nous intéressent ici.

Le paramètre  $\epsilon$  correspond au bruit dans le nuage de points. Cette valeur doit être telle que la grande majorité des points décrivant la primitive  $\mathcal{S}$  soit à une distance d'au plus  $\epsilon$  de  $\mathcal{S}$ . Typiquement, lorsque le bruit du nuage de points est supposé gaussien d'écart type  $\sigma$ , la valeur  $\epsilon = 4 * \sigma$  est appropriée puisqu'elle permet d'englober la quasi totalité des *inliers*. Le paramètre  $\eta$  quant à lui permet de moduler l'amplitude du voisinage moyen, dans lequel les points seront pénalisés. Empiriquement, on constate que  $\eta = 3$  fournit de bons résultats (par la suite, on utilisera cette valeur sauf mention contraire). Le dernier paramètre  $\alpha$  correspond à une tolérance à la déviation angulaire des normales: pour diminuer l'énergie, l'angle formé par les normales observées (dans le nuage de points) et leurs normales théoriques respectives (à la surface de la primitive évaluée) ne doit pas excéder  $\alpha$ .



**Figure 2.5:** Distances et angles pour le calcul de la contribution  $h_D$ . Soit  $\mathbf{p}$  (bleu) le point du nuage  $\mathcal{P}$  dont on veut évaluer la contribution et  $\mathbf{n}_{\mathcal{P}}(\mathbf{p})$  le vecteur normal qui lui est associé.  $\mathbf{p}_S^\downarrow$  (rouge) est le projeté de  $\mathbf{p}$  sur la surface de la primitive  $\mathcal{S}$  (cercle), et  $\mathbf{n}_S(\mathbf{p}_S^\downarrow)$  est le vecteur normal à la surface de  $\mathcal{S}$  en  $\mathbf{p}_S^\downarrow$ .  $a(\mathcal{S}, \mathbf{p})$  est alors l'angle entre les vecteurs  $\mathbf{n}_S(\mathbf{p}_S^\downarrow)$  et  $\mathbf{n}_{\mathcal{P}}(\mathbf{p})$ .  $d(\mathcal{S}, \mathbf{p})$  est par définition la distance entre  $\mathbf{p}$  et  $\mathbf{p}_S^\downarrow$ .

**Terme de contrôle** Le terme de contrôle  $h_C$  permet de pénaliser fortement les primitives ne satisfaisant pas une adéquation minimale avec le nuage de points. En effet, la contribution des points telle que nous l'exprimons ne permet pas à elle seule de discriminer les formes qui ne s'appuient manifestement sur aucun point du nuage. De telles formes apportent une contribution d'attache aux données  $h_D$  nulle puisqu'elles sont éloignées de tout point, et pourraient donc contribuer à l'amélioration de la représentation si elles venaient à se connecter convenablement avec d'autres formes de  $\mathcal{X}$  (induisant une diminution de  $H_T$  - cf. ci-après). Une telle situation n'est évidemment pas souhaitable. Il est donc nécessaire d'empêcher l'existence de telles primitives au sein du modèle reconstruit  $\mathcal{X}$ . Nous proposons pour ce faire d'étudier la couverture des primitives par les points du nuage. Si  $\Gamma(\mathcal{S}, \mathcal{P}) \in [0, 1]$  quantifie la proportion de la surface de  $\mathcal{S}$  couverte par des points de  $\mathcal{P}$ , et étant donné un seuil  $\gamma_0 \in [0, 1]$  de couverture minimale des primitives, on pose alors:

$$h_C(\mathcal{S}, \mathcal{P}) = \begin{cases} 0 & \text{si } \Gamma(\mathcal{S}, \mathcal{P}) \geq \gamma_0 \\ \infty & \text{sinon} \end{cases} \quad (2.20)$$

Nous choisissons d'utiliser une fonction de seuil au lieu d'une fonction continue dans la mesure où l'estimation de  $\Gamma$  est en général très approximative (le calcul est présenté en section 4.4). La mise en place d'une fonction  $h_C$  retournant une très grande valeur lorsque  $\Gamma$  tend vers 0 semble *a priori* pertinente, mais l'incertitude sur la valeur de  $\Gamma$  aurait alors un impact significatif sur l'énergie  $H_D$  et pourrait nuire à l'évaluation de la qualité des primitives. La solution simple que nous choisissons ici permet l'exclusion de toute primitive qui ne serait pas suffisamment recouverte par les points de  $\mathcal{P}$ , et rétablit ainsi la conformité du modèle reconstruit vis-à-vis des données.

L'évaluation d'un terme infini ne peut être intégrée, en pratique, au calcul de l'énergie pour des raisons évidentes de stabilité numérique. Ce terme permet donc uniquement de contrôler la construction du modèle  $\mathcal{X}$  en signifiant que nous interdisons l'insertion de primitives n'étant pas échantillonnées par le nuage.

De manière générale, on prendra d'ailleurs un seuil  $\gamma_0$  bas (de l'ordre de 0.1 au plus), afin de ne pas pénaliser les surfaces qui seraient trop peu échantillonnées. Rappelons que ce terme vise avant tout à s'assurer que toutes les formes passent au moins par un point du nuage, et que l'adéquation des données au modèle est par ailleurs principalement quantifiée

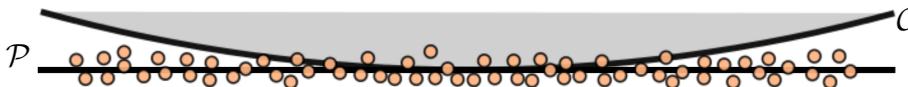
par la contribution des points  $h_D$ .

### Étude du comportement de $H_D$

Considérons ici uniquement la contribution d'attache aux données, et ignorons le terme de contrôle  $h_C$  (par exemple en posant  $\gamma_0 = 0$ ).

Une des principales forces de la démarche proposée réside dans le fait qu'elle permet de pénaliser fortement une forme que l'on tenterait d'ajuster sur une partie du nuage de points qui représenterait une autre forme, mais aussi et surtout de détecter une telle erreur. En effet, les scores existants ne permettent pas de conclure si l'ajustement est pertinent ou non. Leur unique but est de permettre la comparaison de formes, pour décider si une primitive est plus appropriée qu'une autre pour représenter tout ou partie du nuage de points. L'énergie  $H_D$ , lorsqu'elle est positive, permet par construction de déduire que la proportion d'*outliers* vis-à-vis des *inliers* n'est pas conforme aux attentes, et donc qu'il y a trop de points qui décrivent une autre forme que celle ajustée. Dans une telle situation, on peut conclure que la forme évaluée n'est pas appropriée.

		Score		
200 points sur	Forme évaluée	$MSE$	MSAC	Énergie $H_D$
Le plan $\mathcal{P}$	Cylindre $\mathcal{C}$	0.11	21.4	36.66
	Plan $\mathcal{P}$	0.03	7.19	-177.5
Le cylindre $\mathcal{C}$	Cylindre $\mathcal{C}$	0.04	7.4	-175.2
	Plan $\mathcal{P}$	651.8	119.6	176.3

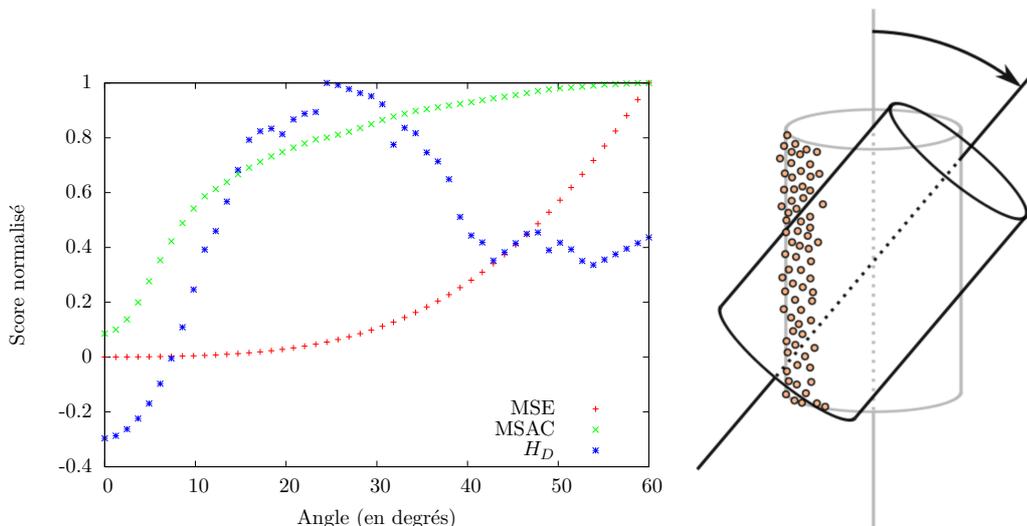


**Table 2.2:** Comparaison de mesures d'ajustement sur un jeu de données synthétique. On considère ici deux primitives: un cylindre  $\mathcal{C}$  de rayon 20 et un plan  $\mathcal{P}$  (patch rectangulaire de  $10 \times 10$ ) tangent à  $\mathcal{C}$ . Pour chaque test, 200 points sont générés aléatoirement sur la surface d'une des deux formes. Les données sont perturbées avec un bruit gaussien d'écart type 0.2 sur la position des points et une erreur angulaire Gaussienne d'écart type  $2^\circ$  sur l'orientation des normales. Le nuage de points est ensuite confronté à chacune des formes. Ce tableau regroupe les mesures d'ajustement obtenues pour chaque cas. On utilise la valeur  $\epsilon = 0.8$  pour définir les inliers (pour MSAC et  $H_D$ ) et la tolérance angulaire  $\alpha = 8^\circ$  pour les normales ( $H_D$  uniquement).

Le tableau 2.2 montre les résultats obtenus sur un exemple synthétique. On y constate clairement que l'ajustement d'un plan sur un cylindre (ou inversement) fournit une énergie positive, tandis que la forme appropriée permet d'obtenir une énergie négative basse.

Finalement, l'énergie  $H_D$  que nous définissons ici présente donc un comportement très intéressant pour notre application, puisque le fait d'introduire dans le modèle CAO reconstruit une primitive qui ne s'ajuste pas convenablement sur le nuage de point conduit à une augmentation de  $H_D$ , permettant ainsi d'en déduire que l'on détériore la qualité du modèle CAO reconstruit. Inversement, l'introduction d'une forme correspondant au nuage diminue l'énergie, et améliore donc la qualité de la reconstruction.

Cependant, l'énergie  $H_D$  présente rapidement un comportement chaotique lorsque la primitive s'éloigne de l'instance optimale (cf. figure 2.6). Cet effet est induit par la non prise en compte des points les plus éloignés, alors que les mesures MSA et MSAC par exemple s'appliquent en tout point de l'espace. Ce comportement peut être gênant pour des applications d'affinement de l'ajustement de primitives sur des nuages de points, lorsque



**Figure 2.6:** Évolution des scores lorsque la primitive ajustée est soumise à une perturbation. En l'occurrence, on considère le cylindre  $\mathcal{C}$  présenté dans le tableau 2.2 ajusté sur un nuage de 200 points bruité. Pour chaque test, on évalue les différents scores (ordonnées) après que le cylindre eût été désaxé d'un certain angle (abscisses).

l'estimation initiale est légèrement trop éloignée de l'optimum. La plupart des algorithmes efficaces en matière d'ajustement requièrent en effet de poser le problème comme une minimisation de fonction convexe. En revanche, au voisinage de l'optimum, l'énergie  $H_D$  semble présenter de bonnes caractéristiques: elle est convexe et décroît très rapidement vers son minimum. Le puits d'énergie faible est donc très resserré autour du minimum.

### 2.3.2 Énergie géométrique

L'énergie géométrique  $H_G(\mathcal{X}, \mathcal{M}_0)$  quantifie la ressemblance entre un modèle CAO reconstruit  $\mathcal{X}$  et le modèle CAO *a priori*  $\mathcal{M}_0$ . Plus ces représentations sont (géométriquement) proches l'une de l'autre, plus cette énergie diminue.

Puisque les modèles CAO s'expriment comme des assemblages d'éléments atomiques, on se propose de mettre en place une mesure de l'éloignement entre modèles CAO qui repose sur une distance entre les éléments qui les composent. Cette démarche, quoique naïve, permet d'établir des critères simples de proximité entre les modèles considérés.

Dans cette optique, l'estimation de l'énergie géométrique passe par la mise en correspondance entre éléments: chaque primitive du modèle CAO reconstruit devra être appariée avec une primitive de  $\mathcal{M}_0$  pour estimer sa pertinence géométrique *a priori*.

#### Méthodes existantes en comparaison de formes

Le problème de la mise en correspondance entre formes est souvent traité à l'aide de descripteurs surfaciques des formes étudiées (cf. section 1.3.4). A. E. Johnson et M. Herbert [JH99] par exemple proposent un descripteur surfacique basé sur le calcul d'images pivotées (*spin images*): chaque image, dont la taille est spécifiée par l'utilisateur, est issue d'un processus de collecte des points à la surface des objets lors de sa rotation autour d'un axe défini par un point et la normale associée à ce point. À l'issue du processus, chaque point à la surface de l'objet (en l'occurrence, la surface est décrite par un maillage surfacique) se voit affecter une image qui, après compression, est exprimée sous forme

de vecteur. La mise en correspondance entre surfaces est ensuite gérée dans l'espace des vecteurs ainsi calculés.

Ces approches sont appropriées pour des surfaces libres relativement vallonnées, présentant donc une grande variété de descripteurs. Pour les primitives géométriques qui nous intéressent ici, ce genre d'approche n'est pas appropriée dans la mesure où les surfaces des primitives qui nous intéressent présentent une géométrie très peu variée. La surface de deux primitives fondamentalement différentes peut ne présenter que de très légères différences au niveau des descripteurs, et il est typiquement difficile de distinguer un plan d'une sphère de très grand rayon à l'aide de ces outils.

M. P. Dubuisson et A. Kain [DJ94] présentent un ensemble de mesures génériques de similarité entre formes basé sur la distance de Hausdorff. La distance de Hausdorff  $HD$  entre deux ensembles  $\mathbb{A}$  et  $\mathbb{B}$  quelconques, étant donnée une distance  $d$  entre les éléments de ces ensembles, se définit comme la plus grande des plus petites distances entre  $\mathbb{A}$  et  $\mathbb{B}$ :

$$HD(\mathbb{A}, \mathbb{B}) = \max \left( \sup_{a \in \mathbb{A}} \inf_{b \in \mathbb{B}} d(a, b), \sup_{b \in \mathbb{B}} \inf_{a \in \mathbb{A}} d(a, b) \right) \quad (2.21)$$

Pour la comparaison de formes en  $n$ -dimensions,  $\mathbb{A}$  et  $\mathbb{B}$  sont des ensembles finis dénombrables de points dans  $\mathbb{R}^n$ , et  $d$  est la distance euclidienne sur  $\mathbb{R}^n$ .

Parmi les distances étudiées par M.P. Dubuisson et A. Kain, il ressort que la distance de Hausdorff modifiée est la plus pertinente pour la comparaison de formes. La distance de Hausdorff modifiée  $MHD$  se définit comme étant le maximum des distances moyennes entre les formes:

$$MHD(\mathbb{A}, \mathbb{B}) = \max \left( \frac{1}{|\mathbb{A}|} \sum_{a \in \mathbb{A}} \min_{b \in \mathbb{B}} d(a, b), \frac{1}{|\mathbb{B}|} \sum_{b \in \mathbb{B}} \min_{a \in \mathbb{A}} d(a, b) \right) \quad (2.22)$$

Dans la situation où les formes à comparer sont des ensembles denses de  $\mathbb{R}^3$  (ce qui est notre cas), on peut aisément imaginer une extension de l'expression 2.22 en remplaçant les sommes par des intégrales et les cardinaux des ensembles par le volume des formes. Le calcul exact de la distance de Hausdorff modifiée devient alors très complexe, mais peut être approximé à l'aide d'un algorithme de Monte Carlo. En tirant un certain nombre de points aléatoirement à l'intérieur de chacune des formes à comparer, l'approximation de la distance de Hausdorff modifiée peut alors être calculée en remaniant légèrement l'expression discrète 2.22 avec  $\mathbb{A}$  et  $\mathbb{B}$  les ensembles de points générés à partir des deux formes comparées, en utilisant des distances "point à primitive" au lieu de distances "point à point".

### Notre mesure

Considérons le modèle probabiliste à la base de notre démarche. La distribution *a priori* se définit couramment à l'aide d'un modèle d'incertitude: on se donne un certain nombre d'attentes qui, lorsqu'elles sont satisfaites, réalisent les plus hautes probabilités. La probabilité décroît ensuite avec l'éloignement à ces attentes, selon une vitesse dépendant de valeurs de tolérances (ou incertitudes).

Dans le cas qui nous intéresse, le modèle CAO *a priori*  $\mathcal{M}_0$  correspond aux attentes. Il s'agit du modèle le plus probable selon des considérations purement *a priori*. Tout naturellement, ce modèle CAO est accompagné d'incertitudes portant sur sa composition géométrique. Par exemple, les diamètres des tuyaux observés (dans le nuage de points) pourraient être légèrement plus grands que ceux attendus, du fait des différences

pouvant survenir entre les scènes. De même, l'emplacement des divers composants dans l'environnement pourrait changer d'une scène à l'autre, selon des amplitudes que l'on suppose connues *a priori*. Les tolérances définissant les différences attendues font partie de l'*a priori*, au même titre que le modèle CAO  $\mathcal{M}_0$  lui-même. Ainsi, la mise à disposition du modèle CAO *a priori* et des incertitudes qui l'accompagnent permettent la définition d'une distribution pour l'*a priori* géométrique.

Ce modèle d'incertitude permet de séparer les différences géométriques entre  $\mathcal{X}$  et  $\mathcal{M}_0$  selon leur nature. Ainsi, on distingue trois types de changements pouvant survenir:

**orientation** d'une scène à l'autre, les composants peuvent présenter de légères différences d'inclinaison. Généralement, ces différences ne sont pas significatives puisque la fonction d'un composant industriel définit en partie son orientation dans la scène: les générateurs de vapeurs ou plus simplement les échelles, par exemple, sont implantés verticalement, tandis que les tuyaux, souvent fixés contre le bâti, sont principalement verticaux ou horizontaux. Les différences observables sont généralement le fait de contraintes physiques (vibrations, changements de température, etc.) liées à l'utilisation du composant occasionnant des changements mineurs, ou à la construction même qui diffère d'un site à l'autre.

**position** un même composant peut apparaître en des endroits différents d'un site à l'autre. Certains éléments sont en effet mobiles par nature, et peuvent être déplacés au cours du temps. D'autres n'ont pas été implantés à l'emplacement prévu à la construction du site, du fait de contraintes rencontrées durant les travaux. Quoiqu'il en soit, les déplacements constatés peuvent être significatifs à l'échelle de la scène.

**géométrie** ce critère regroupe tous les paramètres permettant de moduler la géométrie intrinsèque d'un objet. Il s'agit typiquement du rayon du cylindre, ou des rayons du tore. Au même titre que son orientation, la géométrie d'un composant est déterminée par la fonction qu'il occupe dans la scène. Les barreaux de rambardes, par exemple, possèdent un rayon très faible tandis que les tuyauteries du circuit principal sont très larges. En général, les changements géométriques de grande amplitude sont donc peu probables. Toutefois, la présence ou l'absence d'isolants calorifuges autour de certaines lignes de tuyauterie peuvent fausser ce constat, et l'on peut occasionnellement constater des géométries significativement différentes pour un même composant.

D'une part, les tolérances à ces trois types de changement n'expriment pas les mêmes grandeurs physiques. D'autre part, leurs amplitudes peuvent être considérablement différentes: alors que l'on peut légitimement s'attendre à voir un composant déplacé, il est beaucoup plus improbable de le voir désorienté ou déformé. Il est donc nécessaire de proposer un modèle d'incertitude intégrant ces tolérances distinctes, ce que ne permettent pas les distances de Hausdorff bien qu'elles soient pertinentes pour la comparaison de formes.

Chacune de ces tolérances est décrite par un paramètre spécifié par l'utilisateur, au titre de donnée *a priori*. En effet, ces valeurs ne peuvent être inférées à partir des données (à moins de disposer du modèle CAO reconstruit, que l'on cherche justement à calculer) et doivent être fournies à titre de connaissance *a priori* quant aux données traitées.

Considérons une primitive donnée  $\mathcal{S}_i$  figurant dans le modèle CAO reconstruit  $\mathcal{X}$ . La probabilité *a priori* de voir  $\mathcal{S}$  apparaître doit être élevée si le modèle CAO *a priori*  $\mathcal{M}_0$  contient une primitive  $\mathcal{S}^j$  semblable à  $\mathcal{S}_i$ . Au contraire, cette probabilité doit être faible

s'il n'existe manifestement aucune forme dans  $\mathcal{M}_0$  suffisamment semblable à  $\mathcal{S}$  pour appuyer la légitimité *a priori*.

Ainsi, l'énergie géométrique que nous proposons s'exprime comme la somme des contributions de chaque primitive  $\mathcal{S}_i$  constituant le modèle CAO reconstruit  $\mathcal{X}$ , après qu'elle ait été associée à la primitive  $\mathcal{S}^j$  de  $\mathcal{M}_0$  dont elle est la plus semblable:

$$H_G(\mathcal{X}, \mathcal{M}_0) = \sum_{\mathcal{S}_i \in \mathcal{X}} \min_{\mathcal{S}^j \in \mathcal{M}_0} h_G(\mathcal{S}_i, \mathcal{S}^j) \quad (2.23)$$

où la contribution  $h_G$  constitue une mesure de similarité entre éléments. On considérera qu'elle est infinie si  $\mathcal{S}_i$  et  $\mathcal{S}^j$  sont deux instances de types différents (un cylindre et un plan). Dans le cas contraire, conformément aux remarques précédentes, cette mesure intègre trois paramètres  $\sigma_p$ ,  $\sigma_o$  et  $\sigma_g$  définissant respectivement les tolérances aux changements de position, d'orientation et de géométrie:

$$h_G(\mathcal{S}_i, \mathcal{S}^j) = \left( \frac{\delta(\mathcal{S}_i, \mathcal{S}^j)}{\sigma_p} \right)^2 + \left( \frac{\theta(\mathcal{S}_i, \mathcal{S}^j)}{\sigma_o} \right)^2 + \left( \frac{\gamma(\mathcal{S}_i, \mathcal{S}^j)}{\sigma_g} \right)^2 \quad (2.24)$$

$\delta()$  correspond au déplacement,  $\theta()$  correspond à la rotation et  $\gamma()$  correspond à la déformation entre deux primitives de même type. L'expression de chacune de ces fonctions varie en fonction du type de primitive considérée (rappelons que nous ne considérons dans un premier temps que les cylindres et plans).

**Cylindres tronqués** On cherche à estimer l'amplitude des déformations conduisant du cylindre  $\mathcal{C}^j$  au cylindre  $\mathcal{C}_i$ .

Tout d'abord, pour évaluer le déplacement, nous proposons d'utiliser la distance séparant le centre  $\mathbf{c}_{\mathcal{C}_i}$  de  $\mathcal{C}_i$  du segment axial  $[\mathbf{s}_{\mathcal{C}^j}, \mathbf{t}_{\mathcal{C}^j}]$  de  $\mathcal{C}^j$  ( $\mathbf{s}_{\mathcal{C}^j}$  et  $\mathbf{t}_{\mathcal{C}^j}$  étant les centres des extrémités de  $\mathcal{C}^j$ ):

$$\delta(\mathcal{C}_i, \mathcal{C}^j) = \begin{cases} \|\mathbf{c}_{\mathcal{C}_i} - \mathbf{s}_{\mathcal{C}^j}\| & \text{si } (\mathbf{t}_{\mathcal{C}^j} - \mathbf{s}_{\mathcal{C}^j}) \cdot (\mathbf{c}_{\mathcal{C}_i} - \mathbf{s}_{\mathcal{C}^j}) < 0 \\ \|\mathbf{c}_{\mathcal{C}_i} - \mathbf{t}_{\mathcal{C}^j}\| & \text{si } (\mathbf{s}_{\mathcal{C}^j} - \mathbf{t}_{\mathcal{C}^j}) \cdot (\mathbf{c}_{\mathcal{C}_i} - \mathbf{t}_{\mathcal{C}^j}) < 0 \\ \|(\mathbf{c}_{\mathcal{C}_i} - \mathbf{c}_{\mathcal{C}^j}) \times \mathbf{a}_{\mathcal{C}^j}\| & \text{sinon} \end{cases} \quad (2.25)$$

Le changement d'orientation est évalué en étudiant l'angle formé par les vecteurs supportant les axes des cylindres:

$$\theta(\mathcal{C}_i, \mathcal{C}^j) = \arccos(|\mathbf{a}_{\mathcal{C}_i} \cdot \mathbf{a}_{\mathcal{C}^j}|) \quad (2.26)$$

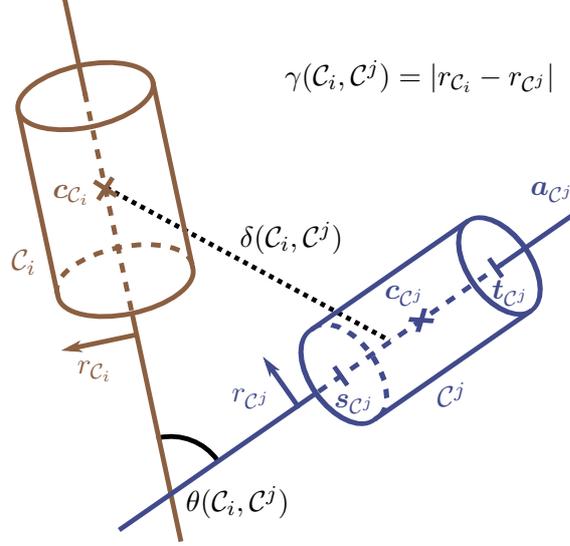
Quant à la déformation (changement de géométrie), nous considérons que la différence de forme n'est liée qu'aux rayons des cylindres. Nous ne prenons pas en compte la longueur des cylindres puisqu'elle ne définit pas intrinsèquement la géométrie des composants. La longueur est en effet dépendante de l'interaction du cylindre avec les autres composants de la scène: le cylindre s'arrête à l'endroit où il rejoint une nouvelle primitive, et ne se termine que très rarement spontanément. La différence de géométrie est donc simplement la différence des rayons :

$$\gamma(\mathcal{C}_i, \mathcal{C}^j) = |r_{\mathcal{C}_i} - r_{\mathcal{C}^j}| \quad (2.27)$$

**Patch rectangulaire plan** On cherche à estimer l'amplitude des déformations conduisant du plan  $\mathcal{P}^j$  au plan  $\mathcal{P}_i$ .

Le déplacement peut être évalué selon le même principe que pour le cas du cylindre, en étudiant la distance du centre  $\mathbf{c}_{\mathcal{P}_i}$  de  $\mathcal{P}_i$  au plan  $\mathcal{P}^j$ :

$$\delta(\mathcal{P}_i, \mathcal{P}^j) = d(\mathbf{c}_{\mathcal{P}_i}, \mathcal{P}^j) \quad (2.28)$$



**Figure 2.7:** Comparaison de cylindres. La similarité entre  $C_i$  et  $C^j$  est estimée en quantifiant leur différence de position  $\delta$  (déplacement), d'orientation  $\theta$  (rotation) et de géométrie  $\gamma$  (déformation).

La distance  $d(\mathbf{p}, \mathcal{P})$  d'un point  $\mathbf{p}$  à un plan  $\mathcal{P}$  étant définie en équation A.2 (annexes).

De même que pour le cylindre, la composante de rotation correspond à l'angle formé par les vecteurs normaux aux plans:

$$\theta(\mathcal{P}_i, \mathcal{P}^j) = \arccos(|\mathbf{n}_{\mathcal{P}_i} \cdot \mathbf{n}_{\mathcal{P}^j}|) \quad (2.29)$$

On ignore en effet la rotation autour du vecteur normal (représentée par  $\mathbf{d}_{\mathcal{P}_i}$ ) puisqu'elle ne sert qu'à définir l'orientation des bords du plan. Or, comme pour le cylindre, les bords du plan ne sont pas une caractéristique intrinsèque de la forme, mais résultent plutôt de son interaction avec son environnement. Il ne nous paraît donc pas pertinent de tenir compte de ce paramètre dans l'évaluation de la similarité entre deux plans.

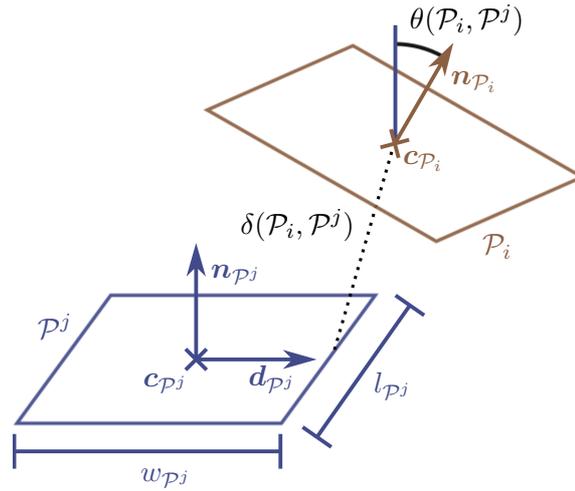
Concernant la différence géométrique, elle n'a pas lieu d'être pour les plans et la fonction  $\gamma$  est donc nulle.

### 2.3.3 Énergie topologique

Le terme  $H_T(\mathcal{X})$  permet d'évaluer la cohérence de tout modèle CAO  $\mathcal{X}$  en se basant sur un certain nombre de critères *a priori*. Ces règles sont établies à partir de connaissances métier portant sur la scène à reconstruire, qui définissent les attentes quant à la composition du modèle reconstruit (cf. section 2.1.3). Ces attentes peuvent être de nature géométrique (paramètres nominaux pour les composants) ou topologique (règles d'assemblage des éléments constitutifs des modèles CAO).

Cependant, les critères géométriques sont "contenus" dans le modèle CAO *a priori*  $\mathcal{M}_0$ . En effet, étant représentatif de l'environnement que l'on cherche à reconstruire,  $\mathcal{M}_0$  est soumis aux mêmes règles, et satisfait donc par hypothèse les connaissances métier de nature géométrique. Or, la pertinence géométrique des modèles CAO reconstruits est évaluée via l'énergie  $H_G$  définie précédemment, qui encode donc (entre autres) les connaissances métier géométriques quant à la scène traitée.

En conséquence, l'énergie  $H_T$  ne doit évaluer que la topologie du modèle CAO reconstruit, c'est à dire la qualité de l'assemblage des primitives en un modèle CAO consistant.



**Figure 2.8:** Comparaison de plans. La similarité entre deux plans se base uniquement sur les différences de position  $\delta$  et d'orientation  $\theta$ .

On s'intéresse donc à la manière dont les primitives composant le modèle CAO reconstruit s'interconnectent. On considère que deux primitives se connectent si elles sont en contact (c'est-à-dire s'il existe un point de  $\mathbb{R}^3$  qui soit à la fois dans les deux primitives - cf. annexe B), ou si elles partagent un point du nuage (c'est-à-dire s'il existe un point de  $\mathcal{P}$  dont la distance aux deux primitives est inférieure à  $\eta\epsilon$  introduit précédemment). On choisit alors d'exprimer l'énergie topologique  $H_T$  comme la somme des contributions apportées par chaque paire de primitives connectées  $(\mathcal{S}_i, \mathcal{S}_j) \in \mathcal{X}^2$ :

$$H_T(\mathcal{X}) = \frac{1}{2} \sum_{\mathcal{S}_i \in \mathcal{X}} \sum_{\mathcal{S}_j \in \mathcal{X} \setminus \{\mathcal{S}_i\}} h_T(\mathcal{S}_i, \mathcal{S}_j) \quad (2.30)$$

L'expression de la contribution topologique  $h_T$  dépend des types des primitives évaluées. Quoiqu'il en soit, chaque contribution modélise les attentes correspondant aux règles métier associées aux primitives étudiées.

### Contribution cylindre - cylindre

Partant des connaissances métiers présentées en section 2.1.3, nous retenons cinq règles concernant les connexions entre cylindres:

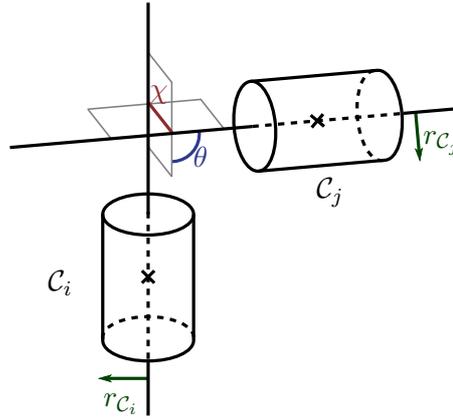
1. les axes de cylindres connectés doivent être concourants.
2. les axes de cylindres connectés ne peuvent former qu'un angle droit ou plat.
3. le cas où le rayon de deux cylindres connectés (non coaxiaux) est identique doit être favorisé, mais on ne doit pas pénaliser la différence des rayons
4. deux cylindres connectés coaxiaux possèdent nécessairement deux rayons différents. En effet, la présence simultanée de deux cylindres alignés, en contact et de même rayon dans un même modèle CAO est incohérente du point de vue du métier de la tuyauterie (il devrait alors s'agir d'un seul et même cylindre).
5. les cylindres ne peuvent se chevaucher. Le volume d'intersection de deux cylindres doit être négligeable face au volume des cylindres eux-mêmes.

Notons d'ores et déjà que, alors que toutes les autres règles définissent des contraintes dures, la troisième est plus lâche puisqu'elle propose de ne pas interdire les situations ne correspondant pas aux cas favorables.

La contribution d'une paire de cylindres connectés  $(\mathcal{C}_i, \mathcal{C}_j)$  s'exprime comme la somme de 5 termes représentant chacun une des contraintes mentionnées ci-avant:

$$h_T(\mathcal{C}_i, \mathcal{C}_j) = \frac{h_T^d(\mathcal{C}_i, \mathcal{C}_j) + h_T^a(\mathcal{C}_i, \mathcal{C}_j) + h_T^r(\mathcal{C}_i, \mathcal{C}_j) + h_T^c(\mathcal{C}_i, \mathcal{C}_j) + h_T^v(\mathcal{C}_i, \mathcal{C}_j)}{3} \quad (2.31)$$

Les termes du membre de droite apparaissent dans l'ordre des règles auxquelles ils sont associés.



**Figure 2.9:** Connexion entre deux cylindres. On suppose les cylindres en contact bien que (pour des raisons de lisibilité) ce ne soit pas le cas sur cette illustration. La contribution apportée par cette connexion dépend notamment de la distance  $\chi(\mathcal{C}_i, \mathcal{C}_j)$  et de l'angle  $\theta(\mathcal{C}_i, \mathcal{C}_j)$  entre les axes des cylindres, ainsi que de leur différence de rayon.

**Distance entre les axes** Le premier terme  $h_T^d$  favorise les cylindres dont la distance  $\chi$  entre les axes est faible, et pénalise au contraire ceux pour lesquelles  $\chi$  excède une tolérance  $\tau_d$  spécifiée, de manière à encourager l'apparition de cylindres connectés possédant des axes concourants. On utilise pour se faire une gaussienne inversée centrée:

$$h_T^d(\mathcal{C}_i, \mathcal{C}_j) = \beta - (\beta + 1) \left( \frac{\beta}{\beta + 1} \right) \left( \frac{\chi(\mathcal{C}_i, \mathcal{C}_j)}{\tau_d} \right)^2 \quad (2.32)$$

Le paramètre  $\beta$  définit l'amplitude de la fonction  $h_T^d$ : la gaussienne définie par cette équation est une fonction croissante de  $\chi$  variant de  $-1$  à  $\beta$ , et s'annulant en  $\tau_d$  (cf. figure 2.10).

La distance  $\chi$  entre les axes des cylindres  $\mathcal{C}_i$  et  $\mathcal{C}_j$  est la distance entre les deux droites définies respectivement par le point  $\mathbf{c}_{\mathcal{C}_i}$  et le vecteur directeur  $\mathbf{a}_{\mathcal{C}_i}$  d'une part, et le point  $\mathbf{c}_{\mathcal{C}_j}$  et le vecteur directeur  $\mathbf{a}_{\mathcal{C}_j}$  d'autre part. Cette distance correspond à la distance euclidienne séparant les deux points les plus proches portés par chacune des deux droites, qui peuvent être calculés en posant le problème de minimisation suivant: on cherche les points

$$\begin{aligned} \mathbf{q}_i(t_i) &= \mathbf{c}_{\mathcal{C}_i} + t_i \mathbf{a}_{\mathcal{C}_i} \\ \mathbf{q}_j(t_j) &= \mathbf{c}_{\mathcal{C}_j} + t_j \mathbf{a}_{\mathcal{C}_j} \end{aligned} \quad (2.33)$$

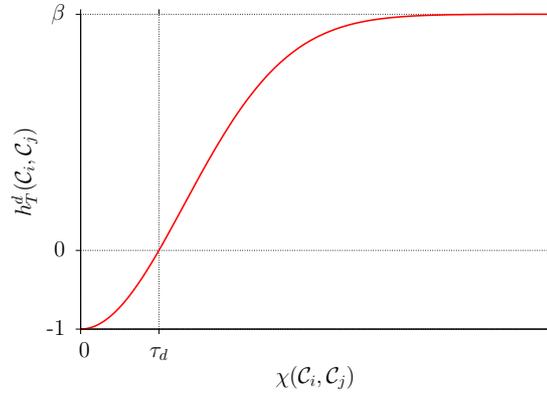
qui minimisent la distance quadratique  $\|\mathbf{q}_i(t_i) - \mathbf{q}_j(t_j)\|^2$ . Le calcul du point critique, c'est-à-dire le point où les dérivées partielles de la fonction cible par rapport à  $t_i$  et  $t_j$  s'annulent, permet ainsi de constituer un système de deux équations à deux inconnues  $t_i$  et  $t_j$ , dont l'inversion permet de trouver:

$$\begin{aligned} t_i &= \frac{(\mathbf{c}_{C_i} - \mathbf{c}_{C_j}) \cdot ((\mathbf{a}_{C_i} \cdot \mathbf{a}_{C_j}) \mathbf{a}_{C_j} - \|\mathbf{a}_{C_j}\|^2 \mathbf{a}_{C_i})}{\|\mathbf{a}_{C_i}\|^2 \|\mathbf{a}_{C_j}\|^2 - (\mathbf{a}_{C_i} \cdot \mathbf{a}_{C_j})} \\ t_j &= \frac{(\mathbf{c}_{C_i} - \mathbf{c}_{C_j}) \cdot (\|\mathbf{a}_{C_i}\|^2 \mathbf{a}_{C_j} - (\mathbf{a}_{C_i} \cdot \mathbf{a}_{C_j}) \mathbf{a}_{C_i})}{\|\mathbf{a}_{C_i}\|^2 \|\mathbf{a}_{C_j}\|^2 - (\mathbf{a}_{C_i} \cdot \mathbf{a}_{C_j})} \end{aligned} \quad (2.34)$$

On a ainsi

$$\chi(\mathcal{C}_i, \mathcal{C}_j) = \|\mathbf{q}_i(t_i) - \mathbf{q}_j(t_j)\| \quad (2.35)$$

où  $\mathbf{q}_i$  et  $\mathbf{q}_j$  d'une part, et  $t_i$  et  $t_j$  d'autre part sont respectivement définis par les équations 2.33 et 2.34.



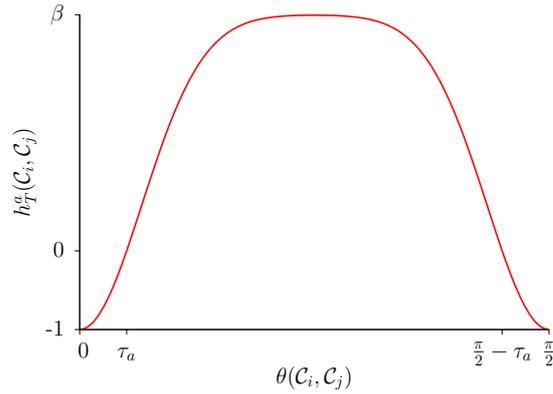
**Figure 2.10:** Évolution de la contribution d'une paire de cylindres  $(\mathcal{C}_i, \mathcal{C}_j)$  connectés en fonction de la distance  $\chi$  entre leurs axes. La fonction est négative lorsque cette distance est faible, et positive lorsqu'elle excède la tolérance  $\tau_d$ .

**Angle entre les axes** Le second terme  $h_T^a$  permet de favoriser les paires de cylindres formant des angles droits ou plats, tout en pénalisant les configurations ne satisfaisant pas cette attente. Il s'exprime comme une mixture de deux fonctions gaussiennes de l'angle  $\theta$  entre les axes des cylindres, centrées sur 0 et  $\frac{\pi}{2}$  et d'écart type commun  $\tau_a$ :

$$h_T^a(\mathcal{C}_i, \mathcal{C}_j) = \beta - (\beta + 1) \left( \left( \frac{\beta}{\beta + 1} \right)^{\left( \frac{\theta(\mathcal{C}_i, \mathcal{C}_j)}{\tau_a} \right)^2} + \left( \frac{\beta}{\beta + 1} \right)^{\left( \frac{\theta(\mathcal{C}_i, \mathcal{C}_j) - \frac{\pi}{2}}{\tau_a} \right)^2} \right) \quad (2.36)$$

L'angle  $\theta(\mathcal{C}_i, \mathcal{C}_j)$  entre les axes des cylindre  $\mathcal{C}_i$  et  $\mathcal{C}_j$  est défini en équation 2.26. Les figures 2.11 et 2.13 illustrent la contribution  $h_T^a$ .

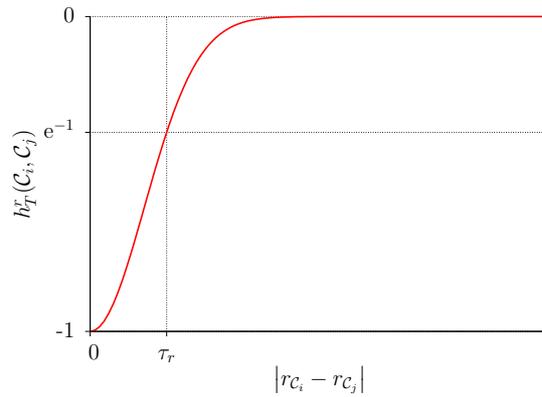
**Différence de rayon** Le troisième terme  $h_T^r$  sert à la valorisation des paires de cylindres connectées possédant un rayon équivalent. Toutefois, il ne s'agit pas ici de pénaliser les cylindres de rayon significativement différents, puisque cette situation est courante. Le but principal du terme  $h_T^r$  est d'encourager l'harmonisation des rayons lorsque cela semble pertinent, sans chercher à contraindre les connections entre cylindres.



**Figure 2.11:** Évolution de la contribution d'une paire de cylindres  $(C_i, C_j)$  connectés en fonction de l'angle  $\theta$  entre leurs axes. La contribution est proche d'être nulle en  $\tau_a$  et  $\frac{\pi}{2} - \tau_a$ , bien qu'elle ne le soit pas exactement. Quoiqu'il en soit, la fonction est positive sur les angles indésirables, et négative au voisinage des angles plats et droits.

De la même manière que ce que nous proposons pour les autres termes, le recours à une fonction gaussienne permet de modéliser cette attente:

$$h_T^r(C_i, C_j) = -e^{\left(\frac{rc_i - rc_j}{\tau_r}\right)^2} \quad (2.37)$$



**Figure 2.12:** Évolution de la contribution d'une paire de cylindres  $(C_i, C_j)$  connectés en fonction de leur différence de rayon. Cette fonction est négative sur son espace de définition, permettant de ne pénaliser aucune configuration.

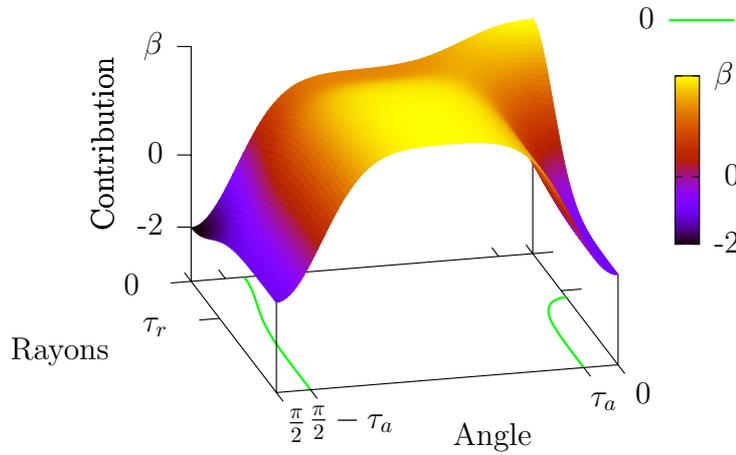
Le paramètre  $\tau_r$  permet de moduler la largeur du bassin d'attraction défini. Il nous semble pertinent d'utiliser  $\tau_r = \epsilon$ . En effet, les cylindres reconstruits à partir du nuage de points sont soumis à l'incertitude du nuage: la précision de la reconstruction dépend, outre le nombre de points utilisés pour l'ajustement des surfaces, du niveau de bruit  $\epsilon$ . Ainsi, pour affiner la précision du rayon des cylindres reconstruits, la valeur  $\epsilon$  paraît appropriée.

**Non alignement de cylindres de même diamètre** Le rôle du quatrième terme  $h_T^c$  est la pénalisation de cylindres connectés possédant à la fois un rayon et une direction similaire. Ce terme doit donc compenser la faible énergie induite par  $h_T^a$  et  $h_T^r$  lorsque l'angle entre les axes des cylindres et leur différence de rayon sont nulles. Comme la somme de ces deux termes vaut  $-2$  dans cette situation, on introduit donc une gaussienne

bidimensionnelle centrée d'amplitude  $\beta + 2$ , permettant de rétablir une pénalité de  $\beta$  pour cette configuration que l'on juge défavorable:

$$h_T^c(\mathcal{C}_i, \mathcal{C}_j) = (\beta + 2) e^{\left(\frac{\theta(\mathcal{C}_i, \mathcal{C}_j)}{2\tau_a}\right)^2 + \left(\frac{r_{\mathcal{C}_i} - r_{\mathcal{C}_j}}{\tau_r}\right)^2} \quad (2.38)$$

Le paramètre  $\tau_r$  (cf. équation 2.37) correspond à une tolérance vis-à-vis de la différence de rayon entre deux cylindres connectés, mais son rôle est ici inversé: au lieu de valoriser les cylindres de diamètres similaires, on cherche dans cette situation à les pénaliser lorsque l'angle qu'ils forment est proche d'être plat. La bosse singulière observable dans la figure 2.13 correspond au terme  $h_T^c$ .



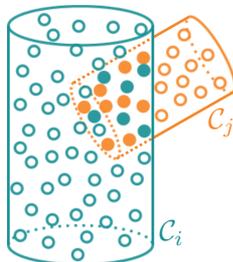
**Figure 2.13:** Contribution d'une paire de cylindres connectés en fonction de la différence de rayons et de l'angle entre les axes des cylindres. La surface représentée correspond à la somme des termes  $h_T^a$ ,  $h_T^r$  et  $h_T^c$  (qui peuvent respectivement être lus dans les plans Rayon= $\infty$ , Angle= $\frac{\pi}{2}$  et Angle=0). La ligne d'isoniveau projetée correspond aux valeurs nulles de la fonction.

**Non chevauchement** Le dernier terme  $h_T^v$  est crucial puisqu'il permet d'empêcher la situation la plus incohérente que l'on puisse rencontrer dans un modèle CAO reconstruit, à savoir le chevauchement de formes s'ajustant sur une même partie du nuage de points. La détection de cylindres se chevauchant se base sur la mesure du volume relatif commun aux deux primitives.

Pour ce faire, on se donne une mesure  $V(\mathcal{C}_i \cap \mathcal{C}_j)$  correspondant à la proportion maximale du volume d'un cylindre qui est inclus dans le second cylindre. L'estimation exacte du volume d'intersection de cylindres quelconques dans  $\mathbb{R}^3$  est un problème complexe. Cependant, l'approximation de ce volume peut être traité de manière relativement simple en adoptant une démarche de Monte Carlo. Le calcul s'effectue en tirant aléatoirement  $N_i$  points et  $N_j$  points respectivement à l'intérieur de chaque cylindre  $\mathcal{C}_i$  et  $\mathcal{C}_j$ . Plus le nombre de points générés est élevé, plus l'approximation est précise. Il suffit ensuite de comptabiliser les points qui se trouvent simultanément dans les deux cylindres. Soient  $n_i$  la quantité de points générés dans  $\mathcal{C}_i$  qui se trouvent dans  $\mathcal{C}_j$ , et  $n_j$  la quantité de points générés dans  $\mathcal{C}_j$  qui se trouvent dans  $\mathcal{C}_i$ . On peut ainsi calculer:

$$V(\mathcal{C}_i \cap \mathcal{C}_j) = \max\left(\frac{n_i}{N_i}, \frac{n_j}{N_j}\right) \quad (2.39)$$

Les paramètres  $N_i$  et  $N_j$  doivent être ajustés de manière à équilibrer l'échantillonnage volumique (le nombre de points par unité de volume) des formes. Il peuvent être calculés à partir d'un paramètre de densité  $\rho$  spécifié par l'utilisateur. Pour estimer  $\rho$ , on pourra par exemple partir sur une base de  $N$  points dans le plus petit cylindre.



**Figure 2.14:** Approximation du volume d'intersection de deux cylindres.  $N_i = 50$  et  $N_j = 20$  points sont générés dans les cylindres de cet exemple. La couleur du contour des points indique le cylindre dont ils proviennent. Les points pleins sont à la fois dans les deux formes, tandis que les vides n'appartiennent qu'au cylindre à partir duquel ils ont été générés. On comptabilise ici  $n_i = 6$  et  $n_j = 9$ , d'où  $V(\mathcal{C}_i \cap \mathcal{C}_j) = 45\%$  approximativement.

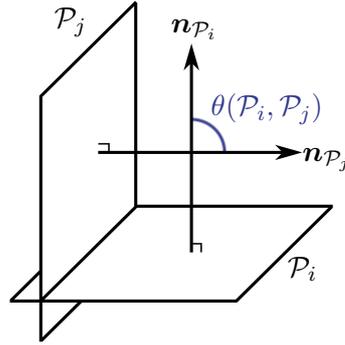
Lorsque le volume commun aux deux formes dépasse un certain seuil, il est alors nécessaire de pénaliser fortement la configuration. Du fait de la nature approximative de  $V(\mathcal{C}_i \cap \mathcal{C}_j)$  et de l'amplitude de la plage de pénalité que l'on souhaite mettre en place pour  $h_T^v$  (de 0 à une très grande valeur positive), la mise en place d'une fonction continue du volume n'est pas la solution la plus opportune, puisque la moindre erreur dans l'évaluation du volume se trouverait multipliée dans la pénalité résultante. Une simple fonction binaire basée sur un seuil est donc plus appropriée: seules les erreurs d'approximation aux alentours du seuil sont impactantes:

$$h_T^v(\mathcal{C}_i, \mathcal{C}_j) = \begin{cases} \omega & \text{si } V(\mathcal{C} \cap \mathcal{C}_j) > \sigma_v \\ 0 & \text{sinon} \end{cases} \quad (2.40)$$

La pénalité  $\omega$  doit être significativement plus grande que  $\beta$  et doit même, pour interdire formellement tout chevauchement, permettre de compenser toute contribution négative (attache au données, géométrie ou topologie) apportée par un cylindre. Typiquement, la valeur  $\omega = 2 \times |\mathcal{P}|$  permet en théorie de proscrire tout chevauchement de cylindres, puisque chaque point du nuage peut apporter une contribution optimale de -1, de même que chaque connexion parfaite avec un élément existant (or il ne peut pas y avoir plus d'éléments dans le modèle reconstruit que de points dans le nuage). Avec une telle valeur, les chevauchements induisent donc fatalement une augmentation d'énergie, et donc une diminution de la probabilité du modèle CAO reconstruit. À des fins de stabilité numérique, des valeurs de l'ordre de quelques dizaines ou centaines de fois supérieures à  $\max(\beta, 1)$  semblent toutefois plus raisonnables, et l'on constate empiriquement qu'elles suffisent à exclure tout chevauchement.

### Contribution plan - plan

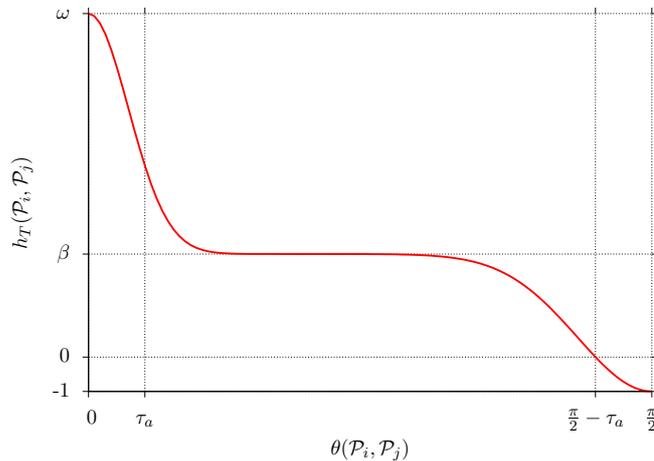
Les règles métiers portant sur les plans (cf. section 2.1.3) se résument en une unique contrainte: les plans connectés doivent être orthogonaux. On peut toutefois nuancer cette contrainte pour, de la même manière que pour les cylindres, pénaliser très fortement le chevauchement de plans.



**Figure 2.15:** Connexion entre deux plans. La contribution apportée par cette connexion dépend de l'angle  $\theta(\mathcal{P}_i, \mathcal{P}_j)$  entre les normales aux plans, dont on attend qu'il soit droit, mais en aucun cas plat.

La contribution topologique  $h_T(\mathcal{P}_i, \mathcal{P}_j)$  pour toute paire de plans connectés  $(\mathcal{P}_i, \mathcal{P}_j)$  est donc uniquement fonction de l'angle  $\theta(\mathcal{P}_i, \mathcal{P}_j)$  entre les vecteurs normaux aux plans (cf. figure 2.15). Elle s'exprime comme une mixture de deux fonctions gaussiennes. La première favorise les angles droits, alors que la seconde pénalise fortement les angles plats puisque deux plans connectés parallèles correspondent à des plans qui se chevauchent:

$$h_T(\mathcal{P}_i, \mathcal{P}_j) = \beta - (\beta + 1) \left( \frac{\beta}{\beta + 1} \right)^{\left( \frac{\theta(\mathcal{P}_i, \mathcal{P}_j) - \frac{\pi}{2}}{\tau_a} \right)^2} + (\omega - \beta) \left( e^{-\left( \frac{\theta(\mathcal{P}_i, \mathcal{P}_j)}{\tau_a} \right)^2} - e^{-\left( \frac{\pi}{2\tau_a} \right)^2} \right) \quad (2.41)$$



**Figure 2.16:** Évolution de la contribution topologique d'une paire de plans  $(\mathcal{P}_i, \mathcal{P}_j)$  connectés en fonction de l'angle  $\theta(\mathcal{P}_i, \mathcal{P}_j)$ . Cette contribution se décompose en deux parties: un bassin d'attraction centré sur  $\frac{\pi}{2}$ , et un pic de pénalisation centré sur 0.

L'angle  $\theta(\mathcal{P}_i, \mathcal{P}_j)$  entre les plans  $\mathcal{P}_i$  et  $\mathcal{P}_j$  est défini en équation 2.29.

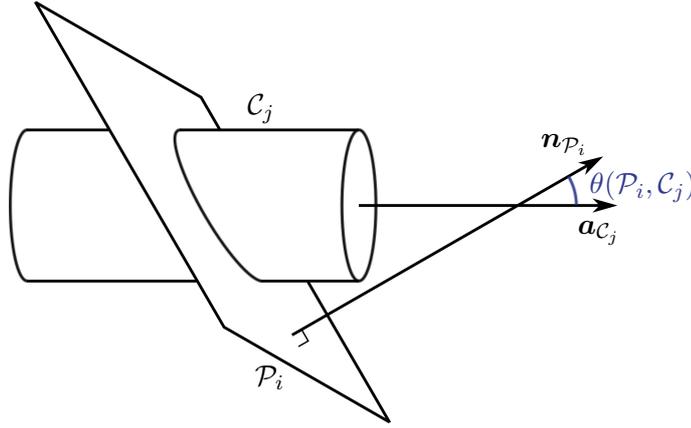
On utilise ici les paramètres  $\beta$  et  $\omega$  introduits précédemment, définissant respectivement les amplitudes de dévalorisation des angles non droits et de pénalité infligée aux angles plats. On a naturellement  $\omega \gg \beta$ .

### Contribution plan - cylindre

Les règles de connexion entre un plan  $\mathcal{P}_i$  et un cylindre  $\mathcal{C}_j$  sont très semblables aux règles de connexion entre plans. On cherche effectivement à favoriser les connexions impliquant un cylindre dont l'axe est orthogonal au plan, tout en évitant à tout prix les connexions où l'axe du cylindre est parallèle au plan.

On définit l'angle  $\theta$  entre le plan  $\mathcal{P}_i$  et le cylindre  $\mathcal{C}_j$ :

$$\theta(\mathcal{P}_i, \mathcal{C}_j) = \arccos(|\mathbf{n}_{\mathcal{P}_i} \cdot \mathbf{a}_{\mathcal{C}_j}|) \quad (2.42)$$



**Figure 2.17:** Connexion entre un plan et un cylindre. La contribution apportée par cette connexion dépend de l'angle  $\theta(\mathcal{P}_i, \mathcal{P}_j)$  entre la direction de l'axe du cylindre et le vecteur normal au plan. Idéalement, cet angle doit être plat.

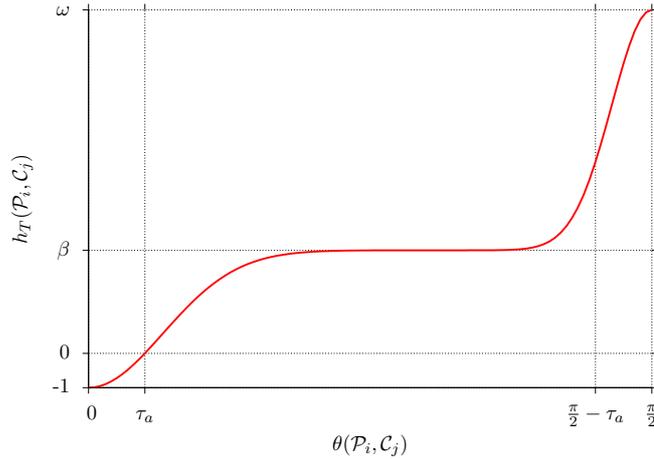
La contribution apportée par la paire  $(\mathcal{P}_i, \mathcal{C}_j)$  s'exprime alors de manière symétrique par rapport à l'équation 2.41:

$$\begin{aligned} h_T(\mathcal{P}_i, \mathcal{C}_j) &= \beta - (\beta + 1) \left( \frac{\beta}{\beta + 1} \right) \left( \frac{\theta(\mathcal{P}_i, \mathcal{C}_j)}{\tau_a} \right)^2 \\ &+ (\omega - \beta) \left( e^{-\left( \frac{\theta(\mathcal{P}_i, \mathcal{C}_j) - \frac{\pi}{2}}{\tau_a} \right)^2} - e^{-\left( \frac{\pi}{2\tau_a} \right)^2} \right) \end{aligned} \quad (2.43)$$

### Paramètres topologiques

L'énergie topologique que nous proposons fait intervenir de nombreux paramètres (cf. tableau 2.3). Contrairement aux paramètres impliqués dans le calcul de l'énergie géométrique, qui ne peuvent pas (ou très difficilement) être inférés, on peut suggérer des valeurs pour les paramètres définissant l'énergie topologique susceptibles de convenir dans la plupart des cas.

Nous avons d'ores et déjà exposé les raisons pour lesquelles il est pertinent de choisir  $\tau_r = \epsilon$ . Nous pouvons en effet anticiper l'erreur attendue sur les rayons de cylindres, du fait du processus de reconstruction. La précision des rayons des cylindres reconstruits est ainsi notamment dépendante du niveau de bruit  $\epsilon$  du nuage de points.



**Figure 2.18:** Évolution de la contribution topologique d'une paire plan - cylindre  $(\mathcal{P}_i, \mathcal{C}_j)$  connectés en fonction de l'angle  $\theta(\mathcal{P}_i, \mathcal{C}_j)$ . La courbe définie symétrique par rapport à celle présentée en figure 2.16

Concernant le choix du paramètre  $\tau_d$ , il paraît judicieux d'adapter cette décision à chaque situation. En effet, l'erreur tolérable quant à l'écart entre les axes des cylindres dépend notamment de la taille des cylindres considérés. Pour paraître concourants, les axes des cylindres connectés devraient être séparés d'une distance inférieure à une fraction du plus petit des rayons des cylindres  $\mathcal{C}_i$  et  $\mathcal{C}_j$  impliqués dans la connexion évaluée. Nous proposons d'utiliser en pratique un dixième du plus petit rayon  $\tau_d = \frac{\min(\mathcal{C}_i, \mathcal{C}_j)}{10}$ .

Quant au paramètre  $\tau_a$ , sa valeur doit être suffisamment faible pour limiter les interférences entre gaussiennes dans les diverses contributions que nous avons définies (il faut que la valeur d'une gaussienne au point central d'une seconde gaussienne soit négligeable par rapport à l'amplitude de cette seconde), tout en étant tout de même relativement permissif vis-à-vis des erreurs pouvant survenir. Une valeur de l'ordre de  $\tau_a = 10^\circ$  permet de satisfaire ce compromis.

Le seuil de recouvrement des cylindres  $\sigma_v$ , pour sa part, ne doit pas être trop faible, puisqu'il n'est pas exclu que les contacts entre cylindres induisent un faible recouvrement. En revanche, nous estimons qu'au delà d'un seuil avoisinant  $\sigma_v = 50\%$  du volume, le recouvrement est trop important pour ne pas représenter un chevauchement.

Enfin, nous avons défini deux types de pénalités. La première ( $\beta$ ) permet de défavoriser légèrement les configurations ne satisfaisant pas une des attentes *a priori*, alors que la seconde ( $\omega$ ) permet de sanctionner lourdement les configurations que l'on souhaite exclure. Ces deux valeurs doivent être positives (strictement), et  $\omega$  doit, par définition, être nettement supérieure à  $\beta$ .

Toutes les contributions topologiques que nous avons définies possèdent un point commun: une attente pleinement satisfaite engendre une contribution de -1. Partant de ce constant, et en observant l'équation 2.31, on remarque qu'une valeur  $\beta$  supérieure à 3 induit une contribution positive pour les connexions entre cylindres si l'une des contraintes "dures" au moins n'est pas satisfaite, quel que soit le statut des 4 autres attentes. Nous proposons donc de choisir  $\beta = 4$ , mais une valeur supérieure permet d'augmenter la "dureté" des diverses attentes, quel que soit le type de connexion considéré.

L'amplitude de pénalité  $\omega$  doit quant à elle permettre de compenser rigoureusement

Paramètre	Description	Connexion			Suggestion
		CC	PP	PC	
$\tau_d$	Tolérance à l'écart entre axes de cylindres. S'exprime en unité de distance.	✓			$\frac{\min(C_i, C_j)}{10}$
$\tau_a$	Tolérance à l'éloignement par rapport aux angles attendus. S'exprime en unité d'angle.	✓	✓	✓	10°
$\tau_r$	Tolérance à la différence de rayons de cylindres. S'exprime en unité de distance	✓			$\epsilon$
$\sigma_v$	Seuil de recouvrement entre cylindres, au delà duquel on considère que les formes se chevauchent. S'exprime en pourcentage.	✓			50%
$\beta$	Amplitude de l'énergie, permettant de pénaliser faiblement certaines configurations peu cohérentes vis-à-vis des attentes <i>a priori</i> .	✓	✓	✓	4
$\omega$	Amplitude des plus lourdes sanctions attribués aux configurations que l'on juge interdites (chevauchements de formes).	✓	✓	✓	100

**Table 2.3:** Récapitulatif des paramètres impliqués dans le calcul de l'énergie topologique. Le type de connexions auxquelles s'appliquent les divers paramètres sont indiqués sous forme de sigles: "CC" correspond aux connexions entre cylindres, "PP" aux connexions entre plans et "CP" aux connexions entre plans et cylindres.

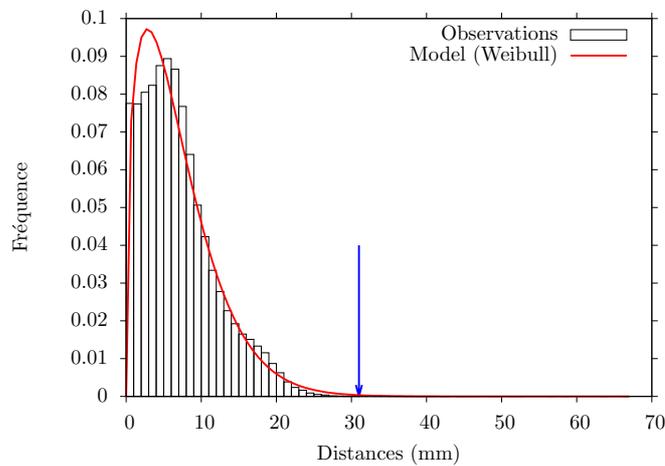
toutes les contributions, de manière à rendre positive la somme des contributions topologique, géométrique et d'attache aux données. Nous avons expliqué qu'en théorie, toute valeur supérieure à deux fois la taille du nuage de point induit fatalement une augmentation de l'énergie si un chevauchement de formes survient, mais qu'il n'est pas souhaitable de recourir à de si grandes valeurs car elles pourraient perturber les calculs. En pratique, nous utilisons une valeur de l'ordre de  $\omega = 100$  pour nos tests, ce qui semble convenir.

### 2.3.4 Évaluation des paramètres de pondération

La plupart des paramètres introduits précédemment peuvent être appréhendés facilement. Les paramètres géométriques  $\sigma_p$ ,  $\sigma_o$  et  $\sigma_g$  (cf. équation 2.24), par exemple, peuvent être rapidement estimés par l'utilisateur, après un bref examen comparatif du nuage de points et du modèle CAO *a priori*. L'utilisateur peut ainsi évaluer visuellement l'amplitude des déplacements de composants, des rotations ainsi que de la différence géométrique. Concernant les paramètres topologiques, nous avons exposé quelques suggestions permettant de leur affecter des valeurs convenables. À l'exception de  $\beta$  et  $\omega$ , l'avantage majeur des paramètres évoqués (tant géométriques que topologiques) réside dans le fait qu'ils ont un sens physique: il expriment des grandeurs liées à la nature même du problème. C'est d'ailleurs ce qui les rend relativement simples à régler.

La perception du niveau de bruit  $\epsilon$  et de l'erreur angulaire  $\alpha$  (cf. équation 2.18) est légèrement plus complexe. L'étude visuelle du nuage n'est pas simple, ou insuffisante pour cerner une valeur convenable. Le paramètre  $\epsilon$  peut être déduit des caractéristiques techniques de l'appareil ayant servi à l'acquisition, lorsque les spécifications sont disponibles. En l'absence de cette information, ou à titre de vérification, il est toutefois possible d'évaluer ce paramètre par le calcul. Il est pour cela nécessaire de se donner une surface correspondant à une partie du nuage de points. Typiquement, on isole un sous-ensemble de points échantillonnant une surface que l'on sait plane, puis on ajuste un plan sur ce

sous-nuage. Une analyse statistique des distances séparant les points de la surface permet d'établir un modèle de bruit pour l'échantillon. Les travaux de thèse de D. Girardeau-Montaut [GM06] englobent cette problématique. Il peut toutefois exister un léger biais introduit par l'ajustement de la surface sur le nuage (idéalement, il faudrait disposer d'une surface issue de vérité terrain), mais ce biais n'est pas critique pour l'estimation des seuils. Une fois le modèle calculé, on peut alors estimer le seuil  $\epsilon$  comme étant la distance au delà de laquelle ne peuvent théoriquement figurer qu'une très faible quantité de points (moins de 0.5% par exemple). Cette étude s'applique de la même manière pour le calcul du seuil  $\alpha$ , en étudiant la répartition statistique des normales autour de la normale théorique, en calculant un modèle de cette répartition, puis en décidant du seuil à adopter de manière à n'exclure qu'une quantité marginale de données. En pratique, il est préférable de sur-estimer les seuils  $\epsilon$  et  $\alpha$ , de manière à permettre de légères erreurs d'ajustement durant le processus de reconstruction, puisque les formes reconstruites sont rarement parfaites et présentent souvent des erreurs minimales qui pourraient induire des pénalités considérables si les seuils n'étaient pas suffisamment permissifs.



**Figure 2.19:** Évaluation du seuil de bruit  $\epsilon$  par étude statistique des distances à une surface (un plan) ajustée sur une partie sélectionnée dans le nuage, dont on sait qu'elle représente la surface modélisée. On peut alors déterminer le seuil  $\epsilon \approx 31$  en éliminant la queue de la distribution déterminée à partir l'histogramme des distances (en l'occurrence une distribution de Weibull, avec  $P(d > \epsilon) = 0.1\%$ ).

La tâche la plus complexe pour l'utilisateur réside dans l'évaluation des paramètres de pondération  $\lambda_D$ ,  $\lambda_G$  et  $\lambda_T$  des différentes énergies (cf. section 2.2.2). Tout d'abord, ces quantités sont adimensionnelles et n'ont aucun sens physique. En outre, elles doivent permettre de gérer la normalisation des énergies pour les ramener à des plages de valeurs comparables, permettant en second lieu d'établir les importances relatives des différents termes. La normalisation ne peut décemment pas être laissée à la charge de l'utilisateur, qui n'a quasiment aucun moyen d'estimer *a priori* l'amplitude des énergies une fois le modèle reconstruit.

Il est donc nécessaire d'apporter une aide à l'utilisateur pour le paramétrage de la pondération des énergies. Divers types d'approches permettent d'aborder ce problème.

**Estimation des valeurs attendues** La méthode la plus simple consiste en l'estimation *a priori* des énergies qui seront probablement atteintes une fois le modèle CAO reconstruit,

Énergie	Min	Max	Commentaire
Attache aux données $H_D$	$- \mathcal{P} $	$(\eta^2 - 1)  \mathcal{P} $	L'énergie d'un modèle CAO reconstruit convenable est négative, mais est en pratique relativement éloignée du minimum théorique puisque les points n'échantillonnent que de manière imparfaite les surfaces à reconstruire.
Attache au modèle CAO <i>a priori</i> $H_G$	0	$+\infty$	En pratique, un modèle CAO reconstruit convenable ne possède qu'une faible énergie, dont la valeur n'excède pas le nombre d'éléments dans le modèle reconstruit, puisque les primitives présentant une contribution trop élevée ne sont pas acceptées dans le processus de reconstruction.
Qualité topologique $H_T$	$-\frac{ \mathcal{X} ( \mathcal{X}  - 1)}{2}$	$\omega \frac{ \mathcal{X} ( \mathcal{X}  - 1)}{2}$	La valeur qu'aura cette énergie une fois le modèle CAO reconstruit ne peut être appréhendée. Elle dépend en effet du nombre de connections, qui ne peut être estimé <i>a priori</i> , ainsi que de la qualité de ces connections qui ne seront que très rarement parfaites. On peut toutefois raisonnablement estimer que cette énergie sera négative pour un modèle CAO reconstruit cohérent.

**Table 2.4:** Plages de valeurs théoriques pour les différentes énergies définies. Ce tableau illustre la diversité des plages de valeurs qui peuvent être prises par les différentes énergies, et donc la complexité d'une estimation *a priori* des paramètres de pondération.

puis en la recherche des coefficients permettant d'équilibrer ces énergies pour leur donner une amplitude comparable. Comme nous le disions, cette estimation est très complexe et demande une expertise poussée des énergies en jeu. Il est possible de se faire une idée très grossière de ce que pourront être ces énergies, en utilisant le tableau 2.4, et en remplaçant le modèle reconstruit par le modèle *a priori*. Cependant cette approche n'est pas fiable, puisqu'elle repose sur une approximation trop grossière et ne permet pas d'estimer précisément l'énergie d'attache aux données, ce qui s'avère critique en pratique.

En supposant que l'on sait estimer les énergies  $H_D^*$  (attache aux données),  $H_G^*$  (géométrique) et  $H_T^*$  (topologique) à l'issue de la reconstruction, la normalisation se fait alors en intégrant l'inverse de ces quantités aux facteurs  $\lambda_D$ ,  $\lambda_G$  et  $\lambda_T$ , de manière à équilibrer la valeur absolue (amplitude) des énergies pondérées:

$$\lambda_D = \left| \frac{\widehat{\lambda}_D}{H_D^*} \right| \quad \lambda_G = \left| \frac{\widehat{\lambda}_G}{H_G^*} \right| \quad \lambda_T = \left| \frac{\widehat{\lambda}_T}{H_T^*} \right| \quad (2.44)$$

De cette manière, les trois termes pondérés sont comparables en amplitude, et il devient alors possible à l'utilisateur de paramétrer intuitivement l'importance que revêt réellement chaque terme dans le processus de reconstruction via les paramètres  $\widehat{\lambda}_D$ ,  $\widehat{\lambda}_G$  et  $\widehat{\lambda}_T$ . Ces trois nouveaux termes correspondent à la proportion dans laquelle l'énergie associée est impliquée dans l'énergie totale  $H$ . Elle peuvent s'exprimer simplement comme des pourcentages, et l'on peut retirer simplement un degré de liberté au problème du paramétrage en se donnant la contrainte suivante :

$$\widehat{\lambda}_D + \widehat{\lambda}_G + \widehat{\lambda}_T = 1$$

En fixant par exemple  $\widehat{\lambda}_D = 0.5$  et  $\widehat{\lambda}_T = 0.4$ , l'utilisateur spécifie ainsi que l'énergie d'attache aux données intervient à 50% dans la résolution du problème, l'énergie topologique à 40% et par conséquent l'énergie géométrique à 10%. La normalisation est alors entièrement transparente, et le paramétrage très intuitif et à la portée de tous, à condition que l'on sache estimer de manière automatique et relativement fiable les valeurs  $H_D^*$ ,  $H_G^*$  et  $H_T^*$ , ce qui est manifestement la tâche la plus complexe. Nous aborderons cette question par la suite.

**Apprentissage brute force** Une méthode fiable, mais très peu subtile, consisterait en la parallélisation des exécutions de diverses instances du programme de reconstruction avec divers jeux de paramètres  $(\lambda_D, \lambda_G, \lambda_T)$ . L'utilisateur devrait alors éliminer les instances qui ne donneraient manifestement pas de bons résultats, de manière à ne retenir que celle permettant d'obtenir la meilleure reconstruction selon lui. Évidemment, il serait souhaitable de ne pas attendre que toutes les reconstructions soit complétées pour ne garder que la meilleure, puisque cela engendrerait une consommation inutilement excessive des ressources de calcul. On suppose donc que le processus de reconstruction est en mesure de fournir à la demande tout résultat intermédiaire pour que l'utilisateur puisse l'évaluer (l'algorithme de reconstruction que nous mettons en place satisfait effectivement cette contrainte).

Pour limiter les ressources nécessaires à ce processus, il est toutefois important de n'explorer que les jeux de paramètres qui paraissent pertinents *a priori*. On se trouve alors face au fameux problème de la poule et de l'œuf, puisqu'il nous faut, pour trouver un bon jeu de paramètres, fournir de bons jeux de paramètres. Cette solution semble donc difficilement applicable en pratique, bien que l'idée de l'intervention de l'utilisateur soit attrayante.

**Apprentissage supervisé** La solution que nous retenons combine les deux pistes mentionnées précédemment. Elle utilise en effet l'expertise de l'utilisateur tout en sollicitant peu d'efforts de sa part, et approfondit l'approche de normalisation à partir d'estimations d'énergies pour le modèle CAO reconstruit. Nous proposons en effet de mettre en place une procédure simple d'apprentissage de valeurs assimilables à  $H_D^*$ ,  $H_G^*$  et  $H_T^*$  à partir de données fournies par l'utilisateur.

Pour ce faire, nous attendons de l'utilisateur qu'il fournisse en entrée du programme une segmentation partielle du nuage (une dizaine de segments au plus), où chacun des segments représente un cylindre ou un plan. Le programme calcule ensuite automatiquement les primitives s'ajustant sur chaque segment, par minimisation de l'erreur quadratique moyenne (cf. thèse de T. Chaperon [Cha02] par exemple). Les primitives ajustées permettent au passage de calculer  $\epsilon$  et  $\alpha$ .

Les primitives calculées forment ainsi un modèle CAO reconstruit partiel  $\tilde{\mathcal{X}}$ , dont l'énergie d'attache au nuage de points  $H_D(\tilde{\mathcal{X}}, \mathcal{P})$  peut être calculée.

Une estimation de l'énergie topologique  $H_T(\tilde{\mathcal{X}})$  est ensuite calculée à partir de ce modèle partiel, selon deux situations possibles:

**Les données fournies concernent une zone concentrée** dans laquelle les éléments segmentés sont interconnectés. Dans ce cas,  $\tilde{\mathcal{X}}$  fournit une représentation pertinente de ce que seront les connections dans le modèle CAO reconstruit complet, et  $H_T(\tilde{\mathcal{X}})$  peut alors être calculé directement à partir de  $\tilde{\mathcal{X}}$ .

**Les primitives segmentées sont indépendantes** et  $\tilde{\mathcal{X}}$  ne contient pas suffisamment de connections pour être représentatif de ce que sera le modèle CAO complet. Dans

ce cas, nous calculons  $H_T(\tilde{\mathcal{X}}) = -2|\tilde{\mathcal{X}}|$  à partir de l'hypothèse selon laquelle chaque primitive est parfaitement connectée avec 2 autres primitives, ce qui correspond à une approximation grossière d'un cas moyen.

Nous supposons que la dizaine de segments fournie ne peut être représentative de la diversité des modifications pouvant survenir entre le modèle CAO *a priori* et le modèle CAO reconstruit complet. On estime  $H_G(\tilde{\mathcal{X}}, \mathcal{M}_0) = (\frac{1}{2})^2 \times |\tilde{\mathcal{X}}|$  sur la base d'une hypothèse grossière moyenne selon laquelle chaque primitive présente des modifications dont l'amplitude vaut la moitié des tolérances aux changements.

Pour finir, les énergies calculées à partir du modèle CAO partiel  $\tilde{\mathcal{X}}$  permettent d'obtenir une approximation pour les coefficients en utilisant  $H_D^* = H_D(\tilde{\mathcal{X}}, \mathcal{P})$ ,  $H_G^* = H_G(\tilde{\mathcal{X}}, \mathcal{M}_0)$  et  $H_T^* = H_T(\tilde{\mathcal{X}})$  dans l'équation 2.44. Plus le nombre de segments fournis en entrée est élevé, plus cette approximation est précise, puisque fidèle au modèle CAO final. Cependant, il est évidemment hors de question de demander une segmentation complète de la scène.

Cette démarche ne se veut donc en aucun cas exacte. Elle permet simplement d'aider grandement l'utilisateur en le déchargeant de la tâche de normalisation des énergies. Il se peut que les diverses hypothèses formulées ne soient pas vérifiées, induisant une erreur significative dans les coefficients calculés. L'utilisateur devra alors compenser ces erreurs en agissant sur les coefficients  $\widehat{\lambda}_D$ ,  $\widehat{\lambda}_G$  et  $\widehat{\lambda}_T$ . En pratique, cette méthode s'est avérée suffisante sur les cas testés.

De manière générale, les énergies d'attache aux données et topologique ont une importance équivalente. En revanche, l'énergie géométrique agit comme un terme de contrôle, permettant d'éviter les configurations manifestement incohérentes vis-à-vis du modèle CAO *a priori*. L'objectif n'est pas nécessairement de faire en sorte que le modèle CAO reconstruit soit le plus fidèle possible au modèle CAO *a priori*, mais plutôt d'éviter qu'il n'en soit trop différent. De sorte que le calcul d'un modèle CAO optimal consiste en la recherche d'un modèle CAO qui minimise simultanément les énergies d'attache aux données et topologique, tout en limitant la croissance de l'énergie géométrique. L'importance  $\widehat{\lambda}_G$  accordée à l'énergie géométrique doit donc, la plupart du temps, être très inférieure à celle accordée aux énergies topologique ( $\widehat{\lambda}_T$ ) et d'attache aux données ( $\widehat{\lambda}_D$ ).



# Chapter 3

## Optimisation pour la reconstruction

### Sommaire

---

<b>3.1</b>	<b>Considérations générales d'optimisation</b>	<b>103</b>
<b>3.2</b>	<b>Méthodes de Monte Carlo sur les Chaînes de Markov</b>	<b>104</b>
3.2.1	Cadre théorique	104
3.2.2	Applications	111
3.2.3	Difficultés	114
<b>3.3</b>	<b>Approche gloutonne</b>	<b>116</b>
3.3.1	Éléments incompatibles	116
3.3.2	Transition optimale localement	117
3.3.3	Calcul efficace des conflits	119
<b>3.4</b>	<b>Traitement des coudes</b>	<b>121</b>
3.4.1	Déterminer les points à la jonction des cylindres	122
3.4.2	Calcul du tore approximant les points	123
<b>3.5</b>	<b>Résultats</b>	<b>125</b>
3.5.1	Expérimentation pour la détection de formes dans des images	125
3.5.2	Expérimentations sur des données 3D réelles	127

---

---

### Préambule

---

La recherche du modèle CAO le plus probable a posteriori nécessite la mise en place de méthodes spécifiques d'optimisation. En effet, la complexité de l'ensemble des modèles CAO acceptables et la fonction de probabilité que nous proposons au chapitre 2 ne permettent pas le recours aux techniques les plus courantes en matière d'optimisation. D'une part, la fonction cible ne présente aucune garantie de convexité, et possède selon toute vraisemblance une multitude d'optima locaux. D'autre part, l'espace des solutions au sein duquel on souhaite identifier le modèle CAO le plus probable est relativement complexe, puisqu'il mêle discret (nombre d'objets du modèle) et continu (définition des objets).

La littérature fournit quelques solutions abordant ce type de problème atypique. Les approches stochastiques basées sur l'utilisation de chaînes de Markov réversibles semblent notamment particulièrement appropriées pour la résolution du calcul du modèle CAO le plus probable (section 3.2). Toutefois, les approches existantes ne sont pas sans poser quelques difficultés, et nous proposons dans ce chapitre une méthode alternative s'affranchissant de ces écueils tout en s'inspirant de quelques fondamentaux des algorithmes reposant sur les chaînes de Markov (section 3.3.2). Nous retenons en particulier le mécanisme de saut entre les sous-espaces de modèles CAO, par insertion de nouvelles primitives créées aléatoirement ou par suppression de primitives figurant d'ores et déjà dans la solution en cours de reconstruction.

L'approche proposée est relativement simple dans son principe. A chaque étape, une nouvelle primitive générée aléatoirement est soumise à l'insertion dans le modèle CAO courant. Lors de cette tentative d'insertion, l'étude de l'évolution de la probabilité permet d'identifier un certain nombre de primitives du modèle CAO qui semblent empêcher l'ajout du candidat proposé, du fait des contraintes d'assemblage qui peuvent se voir enfreintes lors de cette opération. Si l'insertion du candidat après que ces primitives conflictuelles aient été supprimées permet l'amélioration de la solution, alors la transition est acceptée. Ainsi, la probabilité du modèle CAO reconstruit grandit au fil des tentatives d'insertion, jusqu'à l'obtention d'une solution optimale conduisant à la stabilisation du processus.

Par la suite, la spécificité du traitement des scènes industrielles nous permet de ne traiter les coudes des lignes de tuyauterie qu'en guise de post-traitement, grâce à des tores s'appuyant sur les cylindres reconnus à l'aide de l'algorithme d'optimisation pour la reconstruction (section 3.4). Ce dernier peut ainsi ne se consacrer qu'au traitement des plans et cylindres, simplifiant ainsi la recherche de la solution optimale.

Bien que moins robuste, en théorie, que les approches existantes, la méthode d'optimisation que nous proposons semble malgré tout fournir en pratique des résultats satisfaisants (section 3.5).

---

### 3.1 Considérations générales d'optimisation

La résolution d'un problème d'optimisation non trivial repose en général sur la mise en œuvre d'une technique d'exploration de l'espace des solutions en vue d'y repérer les extrema recherchés. Introduisons  $f$  une fonction à optimiser, définie sur un ensemble quelconque  $\mathbb{E}$ :

$$\begin{aligned} f : \quad \mathbb{E} &\rightarrow \mathbb{F} \subset \mathbb{R} \\ x \in \mathbb{E} &\mapsto y \in \mathbb{F} \end{aligned}$$

Admettons que l'on cherche à identifier un minimum de  $f$  (bien que le raisonnement soit aussi valable lorsque l'on cherche à maximiser  $f$ ). La fonction  $f$ , si elle admet effectivement une borne inférieure, peut posséder deux types de minima:

- les minima globaux:  $x^*$  est un minimum global s'il n'existe aucun élément de  $\mathbb{E}$  dont l'image par  $f$  soit strictement inférieure à  $f(x^*)$
- les minima locaux:  $x^*$  est un minimum local s'il existe dans  $\mathbb{E}$  un voisinage de  $x^*$  qui ne contienne aucun élément dont l'image par  $f$  soit strictement inférieure à  $f(x^*)$ .

Dans la mesure du possible, on cherche évidemment à identifier un minimum global de la fonction cible  $f$ . Notons que nous ne faisons aucune supposition quant à l'unicité de l'optimum global. La plupart du temps, la présence de minima locaux est une gêne dans la mesure où elle ralentit ou empêche (selon la méthode utilisée) l'identification des minima globaux.

Nous considérons ici le cas où  $\mathbb{E}$  n'est pas un ensemble fini dénombrable, et  $f$  est une fonction non linéaire. Lorsque  $f$  est suffisamment complexe, et que son optimisation ne peut être résolue par le calcul de points critiques (encore faut-il que  $f$  soit dérivable sur son domaine de définition), les méthodes d'optimisation existantes font appel à des schémas de résolution itératifs, reposant sur deux éléments essentiels:

- une stratégie d'exploration de  $\mathbb{E}$ , basée sur une procédure de proposition  $q$ : étant donné un état courant  $x^i \in \mathbb{E}$ , il s'agit de suggérer plus ou moins judicieusement un nouvel état  $x^{i+1} = q(x^i) \in \mathbb{E}$
- une stratégie d'acceptation des solutions, permettant de décider si le nouvel état  $x^{i+1}$  doit être accepté au détriment de l'ancien  $x^i$ .

En alternant successivement les propositions et acceptations, ces méthodes fournissent des garanties de convergence vers un minimum de la fonction objectif  $f$ . Toutefois, peu d'algorithmes sont en mesure de garantir l'obtention d'un minimum global.

Typiquement, lorsque  $f$  est dérivable (donc continue) sur son domaine de définition, il existe toute une famille d'approches permettant de minimiser à chaque itération la fonction en s'appuyant sur ses propriétés différentielles. C'est par exemple le cas de la méthode de descente de gradient, dont la fonction de proposition est décrite par:

$$q(x^i) = x^i - \gamma(x^i)\nabla f(x^i)$$

$\nabla f(x)$  décrivant le gradient de la fonction  $f$  au point  $x$ .

Si la fonction  $\gamma$  est appropriée, la méthode garantit que  $f(q(x)) \leq f(x)$  en tout  $x \in \mathbb{E}$ , de sorte que toutes les propositions peuvent être acceptées. La stratégie d'acceptation est donc ici inexistante. Tout l'intérêt de la méthode repose sur sa stratégie d'exploration basée sur un socle théorique fort.

Toutefois, de par sa simplicité, cette méthode ne permet d'atteindre un minimum global

que si  $x^0$  est convenablement initialisé. La remarque est également valable pour les autres méthodes fonctionnant selon le même principe (méthode de Newton, Levenberg-Marquardt, etc.) et pour lesquelles seule la procédure de proposition change. Finalement, ces méthodes sont parfaitement adaptées pour l'optimisation de fonctions dérivables convexes puisqu'elles possèdent au plus un minimum (*a fortiori* global).

Pour permettre de s'extraire des optima locaux, il est possible d'intervenir sur chacun des deux aspects de l'optimisation: stratégie d'exploration ou celle d'acceptation.

La méthode du recuit simulé [BT93], par exemple, propose un processus d'acceptation élaboré permettant d'autoriser des états pour lesquels la valeur de  $f$  augmente. En tolérant l'augmentation de  $f$  lors de la résolution, il est ainsi possible de quitter un minimum local pour éventuellement atteindre un minimum global.

L'algorithme du recuit simulé introduit l'utilisation d'un paramètre de température comme une fonction décroissante du temps. Ce paramètre régit la dynamique du système: plus la température est élevée, plus le processus de résolution est permissif vis-à-vis des augmentations de  $f$  et donc plus il est instable en apparence. Lors de cette phase hautement dynamique, la capacité de l'algorithme à s'extraire des minima locaux est donc élevée. Au contraire, lorsque la température est faible (en fin de processus), le processus tend à se stabiliser de manière à n'accepter quasiment que les états qui diminuent effectivement la valeur de  $f$ . Lors de cette phase, l'algorithme ne sera donc en mesure de converger vers un optimum global qui si l'état courant en est suffisamment proche, ou si le processus de proposition utilisé permet d'atteindre un tel optimum.

Concrètement, l'intérêt principal du recuit simulé réside dans sa méthode d'acceptation probabiliste. Considérons  $x^i$  l'état courant,  $x^{i+1}$  un état candidat et  $T_i$  la température du système à l'étape  $i$ . La probabilité pour  $x^{i+1}$  d'être accepté est définie par:

$$P(x^{i+1}|x^i) = \begin{cases} 1 & \text{si } f(x^{i+1}) < f(x^i) \\ e^{-\frac{f(x^i)-f(x^{i+1})}{T_i}} & \text{sinon} \end{cases} \quad (3.1)$$

On constate effectivement que plus  $T_i$  diminue, moins il est probable d'accepter une augmentation de  $f$ . Les états pour lesquels  $f$  diminue sont quant à eux systématiquement acceptés, indépendamment de  $T_i$ . Notons de plus que cette approche ne repose sur aucune hypothèse quant à la continuité ni la dérivabilité de  $f$ , ce qui constitue un atout non négligeable.

Concernant la procédure de proposition, il s'agit d'un processus générant aléatoirement un voisin de l'état courant. De sorte que le recuit simulé repose sur une exploration de l'espace de solutions par une succession de déplacements aléatoires relativement localisés.

## 3.2 Méthodes de Monte Carlo sur les Chaînes de Markov

### 3.2.1 Cadre théorique

Le recuit simulé fait partie d'une famille de méthodes reposant sur une exploration stochastique de l'espace des états. Une partie de ces méthodes prennent leurs fondements dans la théorie des chaînes de Markov, définissant des mécanismes de transition probabilistes entre les éléments de l'espace de solutions. Certaines de ces méthodes se prêtent particulièrement bien, en théorie, au traitement du problème que nous avons posé (chapitre 2).

### Chaînes de Markov

Considérons  $X_t$  une variable aléatoire au temps  $t$ , à valeurs dans  $\mathbb{E}$  (l'ensemble d'états). Nous considérons ici des temps discrets: le temps s'écoule par incréments de 1. La variable aléatoire  $X_t$  suit un processus de Markov si la probabilité pour la variable aléatoire de prendre une valeur donnée <sup>1</sup> au temps  $t + 1$  ne dépend que de la valeur de la variable aléatoire au temps  $t$ :

$$\forall x^i \in \mathbb{E} \quad P(X_{t+1} = x^{t+1} | X_t = x^t, \dots, X_1 = x^1, X_0 = x^0) = P(X_{t+1} = x^{t+1} | X_t = x^t)$$

On appelle chaîne de Markov toute séquence  $(X_0, \dots, X_n)$  générée par un tel processus.  $P$  est appelé noyau de transition, et définit le comportement de la chaîne à chaque étape.

Nous ne considérerons dans cette section que les chaînes de Markov homogènes, c'est à dire indépendantes du paramètre temporel  $t$ :

$$\forall x, y \in \mathbb{E} \quad \forall t_0, t_1 \quad P(X_{t_0+1} = y | X_{t_0} = x) = P(X_{t_1+1} = y | X_{t_1} = x)$$

Une chaîne de Markov est dite irréductible s'il existe une probabilité non nulle de relier toute paire d'états  $(x, y)$ :

$$\forall x, y \in \mathbb{E} \quad \exists n \text{ tel que } P(X_{t+n} = x | X_t = y) > 0$$

Une chaîne de Markov est dite périodique s'il existe un état  $x \in \mathbb{E}$  qu'elle ne peut atteindre, après l'avoir quitté, qu'en un nombre de transitions qui soit un multiple d'un nombre entier (autre que 1):

$$\exists x \in \mathbb{E}, n_0 \in \mathbb{N} \setminus \{0, 1\} \text{ tels que } P(X_{t+n} = x | X_t = x) > 0 \implies n = k \times n_0, \quad k \in \mathbb{N} \setminus \{0, 1\}$$

Dans le cas contraire, la chaîne est dite apériodique.

Lorsque l'ensemble des états  $\mathbb{E}$  est fini, toute chaîne de Markov irréductible et apériodique admet une distribution stationnaire  $\pi$ , définie comme vérifiant la relation suivante:

$$\forall x \in \mathbb{E} \quad \pi(x) = \sum_{y \in \mathbb{E}} \pi(y) \cdot P(x|y) \tag{3.2}$$

Cette distribution stationnaire joue un rôle central dans la mesure où elle implique que l'application répétée du processus de Markov défini par  $P$  converge vers la distribution  $\pi$ . Ce qui signifie notamment qu'à partir d'un certain rang  $t \geq t_0$  la variable aléatoire  $X_t$  suit une loi dont la densité de probabilité est  $\pi$ .

Dans le cas où la variable aléatoire  $X_t$  est continue, (l'ensemble des états  $\mathbb{E}$  constitue un espace dense) la distribution stationnaire  $\pi$ , si elle existe, est définie comme une généralisation de l'expression 3.2:

$$\forall x \in \mathbb{E} \quad \pi(x) = \int_{\mathbb{E}} \pi(y) \cdot P(x|y) dy$$

Les conditions assurant l'existence d'une telle distribution dans ce dernier cas sont relativement complexes à vérifier. Elles font appel aux notions de récurrence et d'ergodicité que

---

<sup>1</sup>Par abus de notation, on écrira  $P(X_t = x)$  avec  $x \in \mathbb{E}$  pour dénoter la probabilité que la variable aléatoire  $X_t$  prenne la valeur  $x$ , bien que  $X_t$  puisse prendre ses valeurs dans un ensemble dense. En toute rigueur, il aurait fallu utiliser la notation  $P(X_t \in dx)$  avec  $dx \subset \mathbb{E}$

nous avons choisi de ne pas détailler ici (se reporter à la thèse de Mathias Ortner [ODJ03] par exemple pour plus de détails). Cependant, il existe une classe de chaînes de Markov dont les propriétés constituent une condition suffisante à l'existence ainsi qu'à l'unicité de distribution stationnaire.

Une chaîne de Markov est réversible si elle vérifie la relation suivante (nommée *detailed balance condition*):

$$\forall x, y \in \mathbb{E} \quad \exists \pi' \text{ telle que } \pi'(x) \cdot P(y|x) = \pi'(y) \cdot P(x|y) \quad (3.3)$$

Toute chaîne de Markov réversible admet une probabilité  $\pi''$  comme unique distribution stationnaire.

### Markov Chain Monte Carlo

Les propriétés intéressantes des chaînes de Markov admettant une loi stationnaire peuvent être utilisées pour simuler des distributions quelconques de manière relativement efficace. Il s'agit d'échantillonner une distribution cible  $\pi$  en générant une séquence d'éléments dépendants  $(x^0, \dots, x^n)$ . On parle alors de "Markov Chain Monte Carlo" (MCMC).

L'algorithme de Metropolis-Hastings (algorithme 3, décrit notamment dans [CG95, Wal04, Bro98]) constitue une méthode phare dans ce domaine. Cette approche repose sur la construction d'un noyau de transition caractérisant un processus de Markov réversible. On se donne un noyau de proposition  $q(x^t) \rightarrow x^{t+1}$  permettant de générer un état candidat  $x^{t+1}$  à partir de  $x^t$  avec une probabilité  $Q(X_{t+1} = x^{t+1} | X_t = x^t)$ .

Ce noyau sert exclusivement à l'exploration de l'ensemble des états. Afin de définir le mécanisme de transition, il est couplé à un noyau d'acceptation défini par la probabilité  $A$ :

$$A(X_{t+1} = y | X_t = x) = \min(1, \alpha(y|x))$$

avec

$$\alpha(y|x) = \frac{\pi(y) \cdot Q(x|y)}{\pi(x) \cdot Q(y|x)} \quad (3.4)$$

Par construction, la probabilité de transition  $P_{MH}(X_{t+1} = x^{t+1} | X_t = x^t)$  d'un état  $x^t$  vers un état  $x^{t+1}$  se définit comme le produit de la probabilité de générer  $x^{t+1}$  à partir de  $x^t$  et de la probabilité d'accepter ce nouvel état:

$$P_{MH}(X_{t+1} = x^{t+1} | X_t = x^t) = Q(X_{t+1} = x^{t+1} | X_t = x^t) \cdot A(X_{t+1} = x^{t+1} | X_t = x^t)$$

Dans le cas particulier où l'algorithme refuse une transition, il faut intégrer au noyau  $P_{MH}$  le fait que le processus puisse stagner en  $x^t$ :

$$P_{MH}(X_{t+1} = x^t | X_t = x^t) = 1 - \int_{\mathbb{E}} Q(y|x^t) \cdot A(y|x^t) dy$$

En constatant que  $\alpha(y|x) = \alpha(x|y)^{-1}$  (lorsque l'inversion est possible), on remarque que si  $\alpha(y|x) \leq 1$ , alors  $A(x|y) = 1$ . C'est justement ce mécanisme qui permet d'équilibrer la probabilité de quitter un état et la probabilité d'y revenir, critère permettant d'établir la réversibilité du processus. Partant de ce constat, il est alors possible de vérifier que le noyau de transition  $P_{MH}$  satisfait la condition de réversibilité:

$$\pi(x) \cdot P_{MH}(y|x) = \pi(y) \cdot P_{MH}(x|y)$$

Il en découle que la chaîne de Markov définie par  $P_{MH}$  admet  $\pi$  pour distribution stationnaire. Le processus permet donc de simuler  $\pi$  en se donnant comme seul outil une procédure stochastique d'exploration  $q$  dont on connaît la distribution de probabilité  $Q$ .

---

**Algorithme 3** : Algorithme de Metropolis-Hastings.

---

**Données** :  $q$  : un noyau de proposition aléatoire suivant une loi  $Q$   
 $\pi$ : distribution définissant (à un facteur près) la loi à simuler  
 $T$ : valeur du burn-in (nombre de valeurs à ignorer)  
 $N$ : nombre d'éléments souhaités  
 $x^0 \in \mathbb{E}$ : état initial  
**Résultat** :  $(x_0, \dots, x_{N-1})$  une séquence de  $N$  valeurs distribuées selon  $\pi$ .

```

1  $t \leftarrow 0$  ;
2 Tant que  $t < T + N$  faire
3    $y \leftarrow q(x^t)$  ;                               /* Proposition */
4   Choisir aléatoirement  $u$  dans  $]0; 1]$  suivant une loi uniforme;
5   Si  $u \leq A(X_{t+1} = y | X_t = x^t)$  alors          /* Acceptation */
6      $x^{t+1} \leftarrow y$ ;
7   Sinon
8      $x^{t+1} \leftarrow x^t$ ;
9    $t \leftarrow t + 1$  ;
10 Retourner  $(x^T, \dots, x^{T+N-1})$ 

```

---

Une force souvent mise en avant de l'algorithme de Metropolis-Hastings réside dans le fait que  $\pi$  peut n'être connu qu'à un facteur près: il ne doit pas nécessairement s'agir d'une densité de probabilité. Ceci tient au fait que le noyau d'acceptation dépend du rapport  $\frac{\pi(y)}{\pi(x)}$  (cf. Expression 3.4), dont la valeur est identique quelle que soit la constante multiplicative accolée à  $\pi$ . La seule contrainte quant à  $\pi$  est qu'elle doit être non négative sur son domaine de définition. Notons aussi que l'algorithme requiert un certain temps avant de converger vers la distribution souhaitée, en fonction de la valeur initiale  $x^0$  choisie. Typiquement, lorsque les valeurs de  $\pi$  sont très faibles dans le voisinage de  $x^0$ , il est nécessaire d'ignorer les premières valeurs qui servent essentiellement à positionner la chaîne dans un état  $x^t$  où les valeurs de  $\pi$  sont plus élevées, après quoi la chaîne suivra effectivement la distribution  $\pi$ . On parle alors de *burn-in* pour mentionner ces premières valeurs rejetées.

L'algorithme du recuit simulé s'inspire de l'algorithme de Metropolis-Hastings. L'analogie est évidente si l'on suppose que le noyau de proposition est symétrique:  $Q(x|y) = Q(y|x)$ . La fonction  $f$  à minimiser peut être rendue positive par composition avec la fonction exponentielle, de sorte que, à température  $T$  fixée, la distribution simulée soit:

$$\pi(x) \propto e^{-\frac{f(x)}{T}}$$

Par substitution de  $\pi$  dans l'expression du noyau d'acceptation (équation 3.4), et en tenant compte de la symétrie de  $Q$ , on aboutit bien au noyau d'acceptation du recuit simulé présenté en équation 3.1.

En simulant la densité  $\pi$ , on favorise l'apparition d'états dont l'image par  $f$  est de faible valeur. Lorsque  $T$  diminue, ce phénomène s'accroît puisque les états pour lesquels la valeur de  $f$  est élevée voient leur probabilité  $\pi$  diminuer rapidement, permettant ainsi de focaliser l'exploration sur les minima de  $f$ .

Il s'avère en fait que la contrainte de symétrie de  $Q$  n'est pas nécessaire au bon fonctionnement de l'algorithme (cf. [BT93]). L'algorithme du recuit simulé est en pratique utilisé avec tout type de noyau de proposition, parfois même des noyaux déterministes.

La figure 3.1 illustre sur un exemple pratique simple le comportement de cette méthode, en particulier sa remarquable capacité à passer outre les minima locaux.

### Reversible Jump Markov Chain Monte Carlo

Les méthodes mentionnées jusqu'alors sont applicables lorsque l'espace des solutions  $\mathbb{E}$  est relativement "classique", typiquement  $\mathbb{E} \subset \mathbb{R}^n$ ,  $n \in \mathbb{N}^*$ . Or, dans le cadre de nos travaux, nous nous intéressons à des espaces de la forme:

$$\mathbb{E} = \bigcup_{m_i \in \mathbb{I}} \mathbb{E}^{m_i}$$

où  $\mathbb{I}$  est un ensemble dénombrable (potentiellement infini) d'indicateurs, dont chaque élément  $m_i$  permet de référencer le sous-ensemble  $\mathbb{E}^{m_i}$ . Chaque  $\mathbb{E}^{m_i}$  est de dimension  $d_{\mathbb{E}}(m_i)$ .

Dans ses travaux [Gre95, GH09], Peter J. Green propose une extension des méthodes "Markov Chain Monte Carlo" s'adaptant à de tels espaces. Le principe de base est le même que celui des algorithmes MCMC: on se donne un noyau de proposition, à partir duquel on se propose de construire une chaîne de Markov réversible dont la distribution stationnaire est la distribution cible  $\pi$ . Toute la difficulté réside dans la prise en compte de l'altération du processus de proposition induite par la nécessité de naviguer entre les différents  $\mathbb{E}^{m_i}$ . Il faut en permanence s'assurer que la probabilité de quitter un état et la probabilité d'y revenir s'équilibrent. On parle alors de méthode à base de "sauts réversibles" (RJMCMC: Reversible Jump Markov Chain Monte Carlo).

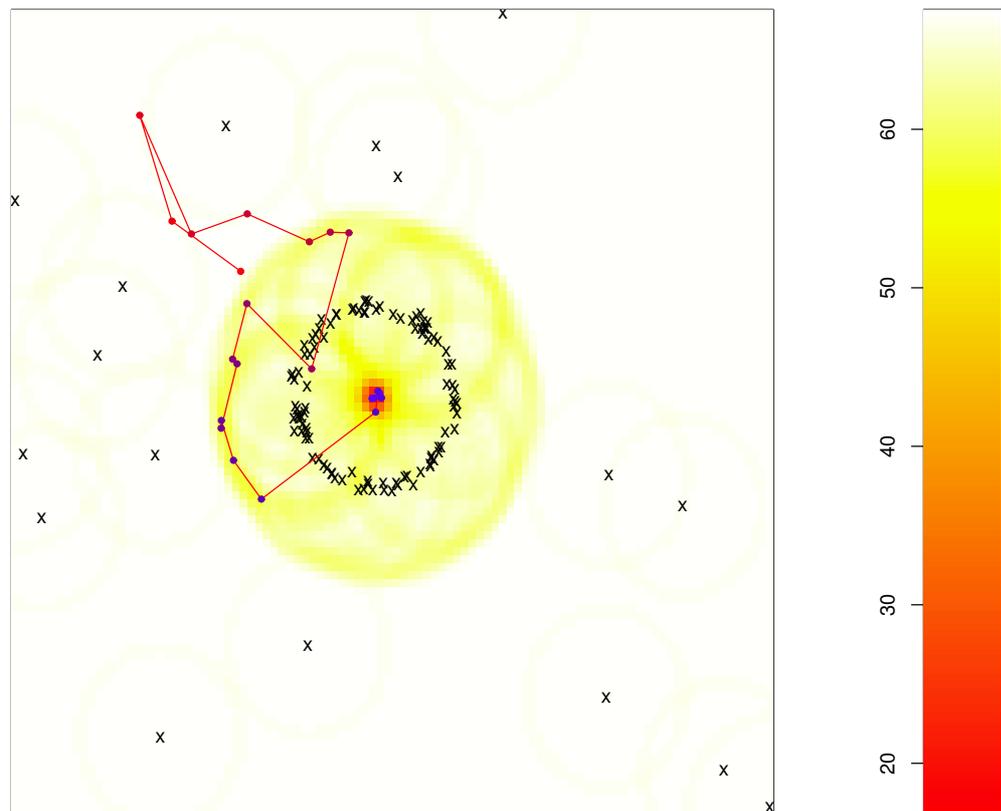
On se place dans un cas simple, où le seul type de proposition disponible est un mécanisme de sauts entre espaces distincts. On suppose de plus que le noyau de saut de l'espace  $m_i$  vers l'espace  $m_j$  est choisi avec une probabilité  $p(m_i, m_j)$ . On a donc en particulier  $p(m, m) = 0$  pour tout  $m \in \mathbb{I}$ .

Pour définir le processus de transition, on introduit des vecteurs aléatoires  $u$  permettant de "compléter" aléatoirement l'état courant lors du changement d'espace. La transition de l'état courant  $x^i \in \mathbb{E}^{m_i}$  vers un état cible  $x^j \in \mathbb{E}^{m_j}$  s'effectue selon l'algorithme suivant:

1. générer aléatoirement un vecteur  $u_i$  de dimension  $n_i$ , selon une densité de probabilité  $Q_{m_i}$
2. calculer  $(x^j, u_j) = g_{m_i \rightarrow m_j}(x^i, u_i)$ . On note  $n_j$  la dimension du vecteur  $u_j$  utilisé ici.

La fonction  $g_{m_i \rightarrow m_j}$  permet de générer de manière déterministe un état de  $\mathbb{E}^{m_j}$  à partir d'un état de  $\mathbb{E}^{m_i}$  et d'une seconde entrée aléatoire  $u_i$  de dimension appropriée. Cette fonction doit être différentiable et bijective. En particulier, la fonction  $g_{m_i \rightarrow m_j}$  admet une fonction réciproque  $g_{m_j \rightarrow m_i}$ , et les dimensions des espaces de départ et d'arrivée doivent se correspondre:

$$n_i + d_{\mathbb{E}}(m_i) = n_j + d_{\mathbb{E}}(m_j)$$



**Figure 3.1:** Exemple de minimisation par recuit simulé. On se donne un ensemble de points  $p_i$  en entrée (croix noires), dont on sait qu'ils échantillonnent un cercle de rayon 1 (on note la présence de bruit et d'outliers). On cherche à trouver le centre du cercle à partir duquel ces données ont été produites. Pour ce faire, on utilise une mesure d'attachement aux données, dont les valeurs sont affichées en fond (le rouge correspond aux valeurs faibles). On peut clairement distinguer que cette fonction n'est pas convexe. La marche de l'algorithme est représentée par la ligne brisée: les étapes sont colorées du rouge (état initial) au bleu (solution calculée). On constate que l'algorithme trouve une solution satisfaisante puisqu'il se stabilise bien en centre du cercle recherché. La température est initialisée à une valeur de 1, et multipliée par un facteur 0.99 à chaque étape. Le noyau de proposition utilisé est le générateur aléatoire de Box-Müller, simulant la loi normale centrée réduite sur  $\mathbb{R}^2$ .

Pour assurer la réversibilité du processus de Markov associé au mécanisme de saut que nous venons de décrire, le noyau d'acceptation se définit par construction comme étant:

$$A(x^j|x^i) = \min \left( 1, \frac{\pi(x^j) \cdot p(m_i, m_j) \cdot Q_{m_j}(u_j)}{\pi(x^i) \cdot p(m_j, m_i) \cdot Q_{m_i}(u_i)} \cdot \left| \frac{\partial g_{m_i \rightarrow m_j}(x^j, u_j)}{(x^i, u_i)} \right| \right) \quad (3.5)$$

Le terme le plus à droite désigne le Jacobien (déterminant de la matrice Jacobienne) associé à l'application  $g_{m_i \rightarrow m_j}$ .

Le processus ainsi défini admet  $\pi$  comme distribution stationnaire. On note que  $\pi$  peut n'être connue qu'à un facteur près puisque, de la même manière que pour l'algorithme de Metropolis-Hastings, les noyaux de transition se définissent en fonction de rapports de  $\pi$ .

### Birth And Death Markov Chain Monte Carlo

M. Stephens [Ste00] présente une alternative aux RJMCMC basée elle aussi sur la navigation entre les différents sous-espaces. L'auteur introduit cette méthode en la qualifiant de "version continue des RJMCMC ne permettant qu'une quantité limitée de transitions possibles pour en simplifier l'implémentation". Concrètement, on se propose deux types de sauts: les naissances et les morts, d'où le nom de *Birth and Death Markov Chain Monte Carlo* (BDMCMC) donné à cette approche.

Avant tout, les états  $x = \{\mathcal{S}_0, \dots, \mathcal{S}_r\} \in \mathbb{E}$  ont une nature particulière, puisqu'ils se définissent désormais comme des ensembles d'éléments atomiques  $\mathcal{S}_i \in \mathbb{F}$ . Ici,  $\mathbb{F}$  désigne l'espace des éléments s'assemblant en une solution, et on a donc:

$$\mathbb{E} = \bigcup_{r \in \mathbb{N}} \mathbb{F}^r \quad (3.6)$$

Par définition, l'ordre des éléments  $\mathcal{S}_i$  qui composent  $x$  n'a aucun impact sur les calculs.  $\mathbb{F}^r$  correspond à l'ensemble des configurations contenant exactement  $r$  éléments atomiques.

Le mécanisme de naissance  $b$  correspond au passage de l'espace  $\mathbb{F}^r \subset \mathbb{E}$  à l'espace  $\mathbb{F}^{r+1} \subset \mathbb{E}$ , par ajout d'un élément atomique à l'état courant:

$$y = x \cup \{\mathcal{S} \in \mathbb{F}\} \quad x \in \mathbb{F}^r \quad y \in \mathbb{F}^{r+1}$$

Le mécanisme de mort  $d$ , quant à lui, correspond au passage de l'espace  $\mathbb{F}^{r+1}$  à l'espace  $\mathbb{F}^r$  lorsque c'est possible (ce qui est le cas si  $r > 0$ ), par retrait d'un élément atomique de l'état courant:

$$y = x \setminus \{\mathcal{S}_i \in x\} \quad x \in \mathbb{F}^{r+1} \quad y \in \mathbb{F}^r$$

Contrairement aux méthodes mentionnées jusqu'alors, les BDMCMC reposent sur des processus de Markov continus, et s'appliquent donc en tout temps  $t \in \mathbb{R}$ . Considérons  $x^t$  l'état courant. Les réalisations de chacun des deux types de transition au cours du temps suivent des processus de Poisson<sup>2</sup> indépendants:

<sup>2</sup>Un processus de Poisson homogène correspond à l'apparition aléatoire d'événements indépendants, et se caractérise par son intensité  $\lambda \in \mathbb{R}$  définie par rapport à une mesure de volume de l'espace dans lequel surviennent ces événements (temps, espace 2D, ...). En particulier, la probabilité d'apparition d'une quantité  $N = k \in \mathbb{N}$  d'événements par unité de volume est définie par:

$$P_\lambda(N = k) = \frac{\lambda^k e^{-\lambda}}{k!}$$

On parle de processus de Poisson non-homogène lorsque le paramètre d'intensité  $\lambda(t)$  est une fonction définie dans l'espace où surviennent les événements.

- les naissances surviennent selon une intensité  $\beta(x^t)$ . De plus, la naissance est un processus aléatoire, lors duquel la probabilité d'apparition de tout élément atomique se caractérise par une densité  $B$  définie sur l'espace  $\mathbb{F}$ .
- les morts surviennent indépendamment pour chaque  $\mathcal{S}_i \in x^t$  selon une intensité  $\delta_i(x^t)$ . L'intensité globale du processus de mort est  $\delta(x^t) = \sum_i \delta_i(x^t)$ .

Dans cette méthode, toute transition est acceptée. Cependant, pour garantir la convergence du processus vers la distribution souhaitée, il est nécessaire de s'assurer que les noyaux de naissance et de mort s'équilibrent convenablement, en vérifiant que la condition 3.3 est bien satisfaite vis-à-vis des transitions ainsi définies ainsi que de la distribution  $\pi$  que l'on cherche à simuler.

Typiquement, on peut construire un mécanisme de naissance et mort équilibré en utilisant un processus de naissance uniforme ( $\beta(x) = \lambda \in \mathbb{R}$ ) et en faisant en sorte que la probabilité de tuer un élément atomique soit forte lorsque cet élément ne semble pas pertinent.  $\delta_i(x)$  doit donc être d'autant plus élevée que  $\pi(x)$  est faible par rapport à  $\pi(x \setminus \mathcal{S}_i)$ .

### De la simulation vers la minimisation

Comme l'algorithme de Metropolis-Hastings, les méthodes RJMCMC et BDMCMC permettent de simuler une distribution  $\pi$  donnée.

Ces méthodes d'échantillonnage, lorsqu'elles sont couplées avec un recuit simulé, permettent l'optimisation de fonctions définies sur des espaces complexes tel que celui qui concerne nos travaux. Du fait de l'utilisation d'une température décroissante, les chaînes de Markov non homogènes ainsi construites convergent vers une distribution de Dirac localisée sur les maxima globaux de  $\pi$  (pour reprendre la formulation de Mathias Ortner et al. [ODJ03]). Comme  $\pi$  se définit elle-même comme une fonction strictement décroissante de la fonction  $f$  à optimiser, on obtient bien en théorie les minima globaux de  $f$ .

### 3.2.2 Applications

Les extensions de la théorie des MCMC offrent un cadre théorique idéal pour la reconnaissance d'un nombre inconnu d'objets parmi des données observées.

Xavier Descombes et al. proposent une grande quantité de travaux basés sur la mise en pratique de ces méthodes, plus spécialement de BDMCMC, dans le cadre de traitement d'images. Nous ne détaillerons que deux d'entre eux, afin d'illustrer l'utilisation de ces approches dans des cas concrets relativement proches de notre thématique de travail. Le lecteur curieux pourra trouver une multitude de travaux sur le sujet, notamment dans les articles [BDZ09, KJD<sup>+</sup>09, ODJ03, DZ02], dans les thèses de Mathias Ortner [Ort04] ou Barna Keresztes [Ker09], ou encore dans le manuscrit d'HDR de Xavier Descombes [Des04].

#### Détection de disques dans des images par BDMCMC

Dans [DMZ08] par exemple, les auteurs s'intéressent à la détection de disques de diamètre fixe dans les images. Ces travaux trouvent typiquement leur application dans le dénombrement d'arbres ou d'oiseaux à partir de photographies aériennes. Pour ce faire, on se

donne tout d'abord une mesure  $H$  de pertinence des configurations, composée de deux énergies: une mesure  $H_1$  d'attache aux données, permettant d'évaluer la présence effective des disques dans la photographie, et une mesure  $H_2$  de chevauchement, permettant de pénaliser les disques s'intersectant. La détection des membres de la population (forêt, oiseaux, ...) dans la photographie se pose alors comme un problème de minimisation de cette mesure, que les auteurs choisissent de résoudre à l'aide de BDMCMC. Chaque état représente un ensemble de disques détectés jusqu'alors, et les opérations de naissance et de mort sont respectivement définies comme l'ajout et le retrait de disques dans l'état courant. L'algorithme repose sur deux paramètres fondamentaux qui varient au cours du temps: la température  $T$  du recuit simulé caractérisant la dynamique du système, et le pas de discrétisation temporel  $dt$ . Les auteurs montrent que cet algorithme (détaillé en algorithme 4), par nature discret, converge vers un processus continu lorsque le pas de discrétisation  $dt$  tend vers 0, et que ce processus correspond effectivement à une BDMCMC permettant de minimiser l'énergie  $H$  lorsque la température diminue.

En pratique, les auteurs suggèrent de recourir à un noyau de naissance non uniforme, de manière à privilégier la naissance d'objets pertinents. Ce principe est mis en œuvre au moyen d'une carte de naissances précalculée, qui associe à tout pixel de l'image sa probabilité d'être le centre d'un disque au sens de l'énergie  $H_1$  d'attache aux données. Quant au noyau de mort, il permet de supprimer préférentiellement les disques non pertinents, permettant de faire diminuer l'énergie. Globalement, la convergence du processus vers un minimum est avant tout définie par le noyau de mort. Quant au noyau de naissance, il permet essentiellement d'influer sur la vitesse de convergence: plus les objets pertinents apparaissent souvent, plus vite on peut atteindre le minimum global.

### Détection de primitives géométriques diverses

Florent Lafarge et al. [LGD10], quant à eux, se placent dans le contexte de la détection de primitives géométriques dans des images (typiquement, le détection de motif répétitifs dans une texture donnée). De la même manière que les travaux de Xavier Descombes et al. présentés ci-avant, les auteurs définissent une énergie quantifiant la pertinence des configurations, se décomposant en deux termes: l'attache aux données et la pénalité de recouvrement entre objets. Il s'agit ici encore de minimiser cette énergie, en utilisant un mécanisme stochastique de sauts entre les différentes configurations, basé sur les processus *Birth and Death*. Toutefois, les auteurs utilisent un mécanisme supplémentaire de "diffusion stochastique", permettant d'assurer l'exploration de l'espace des configurations contenant un nombre constant d'éléments atomiques. Cette approche mêlant sauts entre sous-espaces et exploration aléatoire de chaque sous espace est nommée *Jump-diffusion*.

Supposons que l'énergie  $H$  que l'on cherche à minimiser soit dérivable sur  $\mathbb{F}^r$ , le sous-espace des configurations contenant exactement  $r$  éléments atomiques. La diffusion combine les mouvements de descente de gradients, se déplaçant dans le sens de la plus forte pente de  $H$ , et les mouvements parfaitement aléatoires dont l'amplitude diminue au cours du temps. Ainsi, au temps  $t$ , si  $x^t$  désigne la configuration courante, et  $T(t)$  la température courante, le déplacement effectué est donné par l'équation stochastique de Langevin:

$$dx^t = \nabla H(x^t).dt + \sqrt{2T(t)}d\omega^t$$

où  $d\omega^t \in \mathbb{F}^r$  représente un déplacement aléatoire suivant une loi normale centrée d'écart type  $dt^2$ . On constate effectivement dans la formulation précédente que, lorsque la température diminue, le terme de droite tend à disparaître permettant à la diffusion de se

---

**Algorithme 4** : BDMCMC pour la détection de disques dans une image [DMZ08]

---

**Données** :  $I$ : image de support  $\mathbb{I} \subset \mathbb{N} \times \mathbb{N}$

$H(x) = \alpha.H_1(x) + H_2(x)$ : fonction d'évaluation des configurations  $x$

$T_0$ : température initiale

$f_T(T)$ : fonction positive décroissante de mise à jour de la température

$dt_0$ : pas de discrétisation initial

$f_{dt}(dt)$ : fonction positive décroissante de mise à jour du pas de discrétisation

**Résultat** :  $x^* = \{\mathcal{S}_0, \dots, \mathcal{S}_r\}$ : ensemble des disques détectés

```

1  $T \leftarrow T_0$  ; /* Initialisation */
2  $dt \leftarrow dt_0$ 
3  $x^* \leftarrow \emptyset$ 
  /* Par la suite, on appellera  $\mathcal{S}(i, j)$  le disque centré en  $(i, j)$  */
4 Calculer la carte des naissances  $C_B$  en tout pixel  $(i, j) \in \mathbb{I}$ :
```

$$C_B(i, j) = \frac{\max_{(a,b) \in \mathbb{I}} H_1(\{\mathcal{S}(a, b)\}) - H_1(\{\mathcal{S}(i, j)\})}{\max_{(a,b) \in \mathbb{I}} H_1(\{\mathcal{S}(a, b)\}) - \min_{(a,b) \in \mathbb{I}} H_1(\{\mathcal{S}(a, b)\})}$$

5 **Répéter**

```

6   Pour tout pixel  $(i, j) \in \mathbb{I}$  faire /* Naissances */
7     Si  $\mathcal{S}(i, j) \notin x^*$  alors
8        $x^* \leftarrow x^* \cup \{\mathcal{S}(i, j)\}$  avec une probabilité  $dt.C_B(i, j)$ 
9   Pour tout  $\mathcal{S} \in x^*$  par ordre d'énergie  $H_1(\mathcal{S})$  décroissante faire /* Morts */
10  Calculer la probabilité  $p$  de tuer  $\mathcal{S}$ :
      
$$p(x^*, \mathcal{S}) = \frac{dt.a(x^*, \mathcal{S})}{1 + dt.a(x^*, \mathcal{S})}$$

      avec
      
$$a(x^*, \mathcal{S}) = e^{\frac{H(x^* \setminus \{\mathcal{S}\}) - H(x^*)}{T}}$$

11   $x^* \leftarrow x^* \setminus \{\mathcal{S}\}$  avec une probabilité  $p(x^*, \mathcal{S})$ 
12   $T \leftarrow f_T(T)$  ; /* Mise à jour des paramètres du processus */
13   $dt \leftarrow f_{dt}(dt)$ 
14 jusqu'à tous les éléments nés à cette itération, et seulement ceux-là, ont été tués
15 Retourner  $x^*$ 
```

---

comporter comme une descente de gradient. Au contraire, lorsque la température est suffisamment élevée, le terme de droite induit une composante aléatoire forte, permettant de s'extraire d'éventuels extrema locaux.

Concernant les sauts entre sous espaces, les auteurs choisissent de recourir aux mécanismes de naissance et morts, utilisés conjointement avec un noyau de "transtypage" permettant de transformer un objet de type  $A$  en un objet de type  $B$ . La fonction de transformation associée doit satisfaire les contraintes que nous avons présentées dans la section 3.2.1 présentant les RJMCMC: elle doit être déterministe, bijective et dérivable sur son domaine de définition. En se donnant les outils nécessaires, cette fonction permet alors de construire un noyau de transition conformément à l'équation 3.5.

Afin de permettre la minimisation, ce processus doit être associé à un recuit simulé, dont la température décroissante permet d'assurer la convergence vers les minima recherchés.

Cette approche de résolution se retrouve dans l'article [LKBH10], traitant de la reconstruction hybride de scènes 3D (composition de surfaces maillées et de primitives géométriques).

### 3.2.3 Difficultés

L'élément commun à toutes les approches que nous avons mentionnées ci-avant est le recours au recuit simulé afin de permettre la convergence vers les minima recherchés. Comme nous l'avons déjà mentionné, l'avantage majeur de cette approche réside dans le fait qu'elle permet en théorie de s'extraire des minima locaux.

Si l'évolution de la température est suffisamment lente, l'algorithme doit en théorie se maintenir dans un état permettant d'assurer l'optimisation de la fonction cible. Dans cette optique, deux approches peuvent être utilisées:

- la température est maintenue fixe durant un temps suffisant pour permettre à la chaîne de Markov homogène associée de parvenir à son état d'équilibre, dans lequel l'échantillonnage des configurations obéit effectivement à la distribution stationnaire de la chaîne. Lorsqu'on estime que c'est le cas, on peut alors diminuer la température, et continuer la simulation en tenant compte de la nouvelle température.
- la température diminue après chaque transition. Dans ce schéma, il est nécessaire que l'évolution soit assez lente pour permettre au processus de Markov de "rester collé" à sa distribution stationnaire.

Dans tous les cas, l'obtention d'un minimum global ne peut être garantie qu'au prix de temps de calculs prohibitifs. En théorie [BT93], la convergence du recuit simulé peut être obtenue si et seulement si le schéma de refroidissement satisfait:

$$T_t \geq \frac{d}{\log(t)} \quad (3.7)$$

où  $d$  quantifie la difficulté pour s'extraire d'un extremum local de la fonction cible.

Dans le cas d'un refroidissement trop brutal, on s'expose donc au risque de rester enfermé dans un minimum local. Malgré tout, la plupart des auteurs préfèrent un refroidissement géométrique ( $T_t = T_0 * \gamma^t$  où  $\gamma \in ]0, 1[$ ) au refroidissement logarithmique

(équation 3.7), puisque toutes les applications ne requièrent pas nécessairement la détection du minimum global. Il est toutefois préférable d'être conscient du sacrifice que ce choix implique.

Quel que soit le choix effectué concernant le schéma de refroidissement, il est nécessaire de fournir à l'algorithme une température initiale  $T_0$ . L'appréhension de ce paramètre par l'utilisateur n'est pas simple: cette valeur ne s'exprime pas dans les grandeurs manipulées par le problème, et est propre au recuit simulé. Elle doit en théorie être aussi grande que possible, mais reste difficilement calculable *a priori*.

Nous avons étudié la possibilité d'adapter effectivement l'algorithme 4 pour permettre la détection des primitives géométriques dans des données 3D. En pratique, l'ajout simultané de plusieurs primitives à chaque étape de naissance (primitives générées selon un processus probabiliste non uniforme - cf. chapitre 4), combiné à la méthode proposée pour les morts (tri des objets par pertinence décroissante vis-à-vis des données) semblent empêcher la convergence du processus qui ne cesse d'osciller.

Finalement, l'algorithme du recuit simulé combiné à un outil d'exploration stochastique de l'espace des solutions constitue un bon outil pour la résolution du problème que nous avons posé. Sa mise en pratique ne permet pas forcément d'aboutir à une solution optimale, selon le schéma de refroidissement adopté. En revanche, cette méthode offre des garanties (statistiques) d'obtention de bonnes solutions, ce qui nous convient. Cependant, le paramétrage de l'algorithme le rend très difficile à appréhender: il n'est pas envisageable de laisser l'utilisateur non expert tâtonner pour trouver une bonne température initiale  $T_0$  ou un bon coefficient de refroidissement  $\gamma$ . De plus, même les meilleurs schémas de refroidissement nécessitent de grandes quantités de calculs avant d'aboutir à une solution qui soit à la fois stable et satisfaisante: cet algorithme n'est certainement pas reconnu pour ces bonnes performances en matière de complexité temporelle, et n'est de toute manière pas conçu dans ce but.

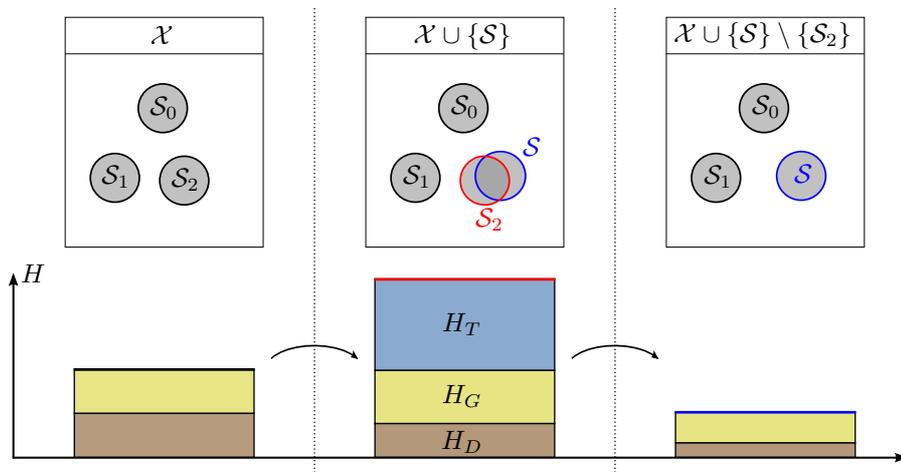
On peut donc légitimement se poser la question suivante: existe-t-il un algorithme plus simple à paramétrer et qui permette d'aboutir à une solution satisfaisante en un temps raisonnable ? Il ne s'agit pas forcément de trouver une solution optimale, puisque les implémentations des méthodes à base de recuit simulé ne permettent pas non plus d'atteindre, en pratique, une telle qualité. En revanche, une méthode alternative devrait, pour être valable, permettre d'obtenir des résultats de qualité équivalente. Typiquement, dans le cadre de cette thèse, une estimation incomplète de la solution, si elle est fiable (bien que sous-optimale), est une solution acceptable puisqu'elle permet d'automatiser une partie du travail de reconstruction.

Les méthodes d'optimisation basées sur la mise en œuvre d'un recuit simulé combinent l'exploration aléatoire de l'espace des solutions et l'acceptation aléatoire des solutions proposées. Ces méthodes supposent que l'on connaisse parfaitement les distributions associées au processus d'exploration, de manière à pouvoir adapter certaines autres propriétés et garantir la réversibilité des MCMC sous-jacentes. Mais que se passe-t-il si nous ignorons les probabilités régissant l'exploration des solutions ? Supposons que l'on dispose d'un oracle permettant de proposer des configurations pertinentes, sans pour autant être en mesure de renseigner les probabilités d'apparition. Il devient alors impossible d'assurer la réversibilité des MCMC.

### 3.3 Approche gloutonne

Au vu des problèmes soulevés par les recuits simulés, nous avons choisi d'étudier une piste alternative, basée elle aussi sur l'exploration aléatoire de l'espace des solutions. En revanche, l'acceptation des solutions est effectuée de manière déterministe, ce qui distingue clairement cette approche des méthodes mentionnées précédemment. Globalement, cette nouvelle méthode fait donc partie de la famille des approches stochastiques: l'état du processus à un instant donné ne peut être déterminé par avance, et seule la probabilité d'y aboutir peut éventuellement être estimée. Nous proposons "uniquement" de restreindre la part d'aléatoire impliquée dans le processus, pour en accélérer la convergence.

#### 3.3.1 Éléments incompatibles



**Figure 3.2:** Exemple de conflit lors d'une tentative d'ajout d'un nouvel élément  $S$ . Ici, les éléments sont des disques. On constate que l'ajout du disque  $S$  provoque une augmentation significative de l'énergie (diagramme inférieur dans lequel chaque énergie est représentée par une couleur, l'énergie totale  $H$  étant égale au cumul de ces énergies). Dans cet exemple, l'augmentation peut manifestement s'expliquer du fait de l'interaction entre  $S_2$  et  $S$ , puisque la suppression de  $S_2$  permet d'aboutir à une meilleure énergie malgré la présence de  $S$  dans la configuration.

On se positionne dans le cadre suivant:

- $\mathcal{X} = \{S_0, \dots, S_{r-1}\}$  désigne une configuration (modèle CAO) appartenant à l'espace  $\Omega$  présenté en section 2.1.2. Le nombre  $r$  d'éléments dans la configuration est fini. Les éléments  $S_i$  appartiennent à l'ensemble  $\Psi$  des primitives géométriques pouvant s'assembler en un modèle CAO de  $\Omega$ .
- on dispose d'une énergie  $H$  définie sur  $\Omega$  permettant de quantifier la pertinence de toute configuration (cf. Section 2.3). Cette énergie comporte trois termes: un terme  $H_D$  d'attache aux données observées, un terme *a priori*  $H_G$ , et un terme  $H_T$  d'assemblage des éléments atomiques. Plus chacun de ces termes est faible, plus la configuration  $\mathcal{X}$  est conforme aux attentes vis-à-vis du critère concerné.
- les seules opérations pouvant être effectuées sur  $\mathcal{X}$  sont l'ajout ou le retrait d'éléments atomiques  $S$ .

Dans ce contexte, on peut aisément concevoir qu'il existe des situations dans lesquelles il est souhaitable ne pas conserver deux éléments au sein d'une même configuration. Imaginons la situation suivante, illustrée par la figure 3.2: on se propose d'ajouter l'élément

$\mathcal{S} \in \Psi$  à la configuration courante  $\mathcal{X} = \{\mathcal{S}_0, \dots, \mathcal{S}_{r-1}\}$ . Après cette modification, on constate une augmentation de l'énergie:  $H(\mathcal{X} \cup \{\mathcal{S}\}) > H(\mathcal{X})$ . Pourtant, on ne peut absolument pas conclure que  $\mathcal{S}$  est un élément médiocre au sens de  $H$ .

En effet, si  $\mathcal{S}$  n'adhère pas aux données, induisant une augmentation de  $H_D$ , ou se trouve être géométriquement très différent de ce qui est attendu, induisant une augmentation de  $H_G$ , on peut effectivement penser qu'il n'est pas souhaitable de conserver  $\mathcal{S}$  puisque cette situation ne saurait évoluer au cours du processus. En effet, dans la mesure où les observations et l'*a priori* géométrique sont immuables, l'augmentation d'énergie provoquée par l'ajout de  $\mathcal{S}$  ne pourrait être compensée que par la suppression de  $\mathcal{S}$ . Finalement, dans une telle situation, il aurait donc été préférable de refuser l'ajout de  $\mathcal{S}$ .

En revanche, il est tout à fait envisageable que l'augmentation d'énergie soit imputable à la présence d'un conflit entre éléments, induisant une augmentation de l'énergie d'assemblage  $H_T$ . Cette hypothèse ne peut être confirmée que par une étude de l'évolution de l'énergie après suppression des éléments conflictuels de  $\mathcal{X}$ . L'ajout du nouvel élément  $\mathcal{S}$  peut tout à fait fournir un meilleur modèle au sens des énergies  $H_D$  et  $H_G$ . Mais l'incompatibilité entre  $\mathcal{S}$  et ses conflits est telle que la présence simultanée de ces éléments dans la configuration pénalise lourdement  $H_T$ , ayant pour conséquence l'augmentation de l'énergie  $H$ . Dans une telle situation, il peut être judicieux d'accepter l'ajout de  $\mathcal{S}$  dans  $\mathcal{X}$ , si cette insertion s'accompagne de la suppression des éléments conflictuels impliqués dans l'augmentation de  $H$ .

### 3.3.2 Transition optimale localement

Partant de ces constats, il peut être intéressant d'envisager une approche simple permettant de gérer efficacement les différentes situations que l'on peut rencontrer lors d'une tentative d'insertion d'un nouvel élément dans la configuration.

Adoptons une démarche gloutonne, reposant par définition sur le choix systématique de la solution optimale à chaque étape de la résolution. L'inconvénient majeur de ce type d'approche réside dans le fait qu'un choix localement optimal ne permet pas nécessairement de tendre vers l'optimum global. Puisque le noyau d'acceptation ainsi défini ne tolère aucune augmentation de la fonction que l'on cherche à optimiser, seul le noyau de proposition est théoriquement en mesure d'extraire l'algorithme des éventuels optima locaux que l'on pourrait être amené à explorer. Il apparaît donc d'ores et déjà que le noyau de proposition doit fournir une méthode d'exploration pertinente de l'espace des solutions lorsqu'il est couplé à un noyau d'acceptation glouton.

Comme nous l'avons mentionné précédemment, l'évolution de l'énergie globale  $H$  est influencée par trois facteurs indépendants lors de l'ajout d'un nouvel élément  $\mathcal{S}$  dans la configuration  $\mathcal{X}$ :

- le degré d'ajustement de  $\mathcal{S}$  dans le nuage de points: l'énergie d'attache aux données  $H_D$  augmente lorsque l'ajustement est médiocre, et diminue lorsqu'il est bon.
- la pertinence géométrique *a priori* de  $\mathcal{S}$ : si le nouvel élément ne correspond pas aux attentes vis-à-vis de l'*a priori*, l'énergie  $H_G$  augmente d'autant plus que l'erreur constatée est significative.
- la qualité d'assemblage des éléments de  $\mathcal{X}$ : l'énergie topologique  $H_T$  augmente lorsque  $\mathcal{S}$  n'est pas cohérent vis-à-vis des éléments de  $\mathcal{X}$ , et diminue lorsque l'assemblage se fait conformément aux attentes.

Parmi ces trois critères, le dernier occupe une place particulière: en effet, l'augmentation de l'énergie d'assemblage constatée lors de l'ajout de  $\mathcal{S}$  peut être suivie d'une diminution conséquente si l'on supprime certains éléments de  $\mathcal{X}$  impliqués dans l'augmentation constatée. Dans certains cas, l'énergie globale  $H$  peut atteindre un niveau inférieur à celui qu'elle avait avant ces modifications.

---

**Algorithme 5** : Algorithme glouton de construction d'une configuration optimale.

---

**Données** :  $\mathcal{P}$ : nuage de points  
 $\mathcal{M}_0$ : modèle 3D *a priori*  
 $H$ : énergie définissant la pertinence de toute configuration  
 $q$ : fonction de génération aléatoire d'éléments atomiques  
 $N$ : critère de stabilité pour l'arrêt de l'algorithme  
**Résultat** :  $\mathcal{X}^*$ : configuration optimale au sens de  $H$

```

1  $\mathcal{X} \leftarrow \emptyset$  ;                               /* Initialisation */
2  $n \leftarrow 0$ ;
3 Tant que  $n < N$  faire
4   Pour tout  $\mathcal{S}^0 \in \mathcal{M}_0$  faire
5      $\mathcal{S} \leftarrow q(\mathcal{S}^0, \mathcal{P})$  ;                /* Proposition (cf. chapitre 4) */
6     Estimer l'ensemble  $K^*$  pour  $\mathcal{S}$  (équation 3.8 - algorithme 6);
7     Si  $H(\mathcal{X} \setminus K^* \cup \{\mathcal{S}\}) < H(\mathcal{X})$  alors /* Test d'acceptation glouton */
8        $\mathcal{X} \leftarrow \mathcal{X} \setminus K^* \cup \{\mathcal{S}\}$  ; /* Mise à jour */
9        $n \leftarrow 0$ ;
10    Sinon
11       $n \leftarrow n + 1$ ;
12 Retourner  $\mathcal{X}^* \leftarrow \mathcal{X}$ 

```

---

Dès lors, il est possible d'optimiser localement l'énergie  $H$  en identifiant le sous ensemble  $K^* \subset \mathcal{X}$  qui, lorsqu'il est supprimé de la configuration  $\mathcal{X}$ , permet d'atteindre l'énergie la plus basse après que l'élément  $\mathcal{S}$  ait été inséré dans  $\mathcal{X}$ :

$$K^* = \underset{K \subset \mathcal{X}}{\operatorname{argmin}} H(\mathcal{X} \setminus K \cup \{\mathcal{S}\}) \quad (3.8)$$

L'ensemble  $K^*$  désigne alors les conflits vis-à-vis de  $\mathcal{S}$ . En effet, ces éléments gênent d'une certaine manière l'insertion de  $\mathcal{S}$ , dans la mesure où cette insertion est optimale lorsque le sous ensemble  $K^*$  est retiré de la configuration. Si l'énergie obtenue après modification (insertion de  $\mathcal{S}$  et suppression du sous ensemble  $K^*$ ) est inférieure à l'énergie de la configuration d'origine  $\mathcal{X}$ , on peut alors accepter cette modification. Dans le cas contraire, l'augmentation d'énergie induite par l'ajout de  $\mathcal{S}$  ne peut pas être compensée par une modification de  $\mathcal{X}$ , et on peut en déduire que  $\mathcal{S}$  est un élément médiocre au sens de l'énergie  $H$ .

L'idée générale de cette démarche est donc la modification itérative de la configuration par ajout d'éléments:

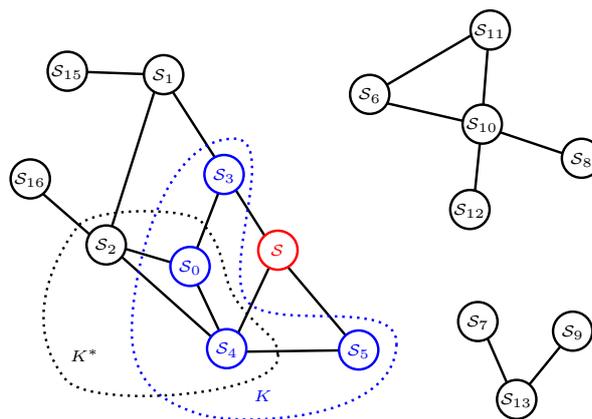
- qui apportent une nouvelle information pertinente vis-à-vis des trois critères définis (si  $K^* = \emptyset$ )
- qui remplacent une information existante, en améliorant la précision de l'estimation vis-à-vis des trois critères définis (si  $K^* \neq \emptyset$ )

L'algorithme 5 décrit en détail la méthode gloutonne que nous proposons pour l'estimation de la configuration optimale au sens de  $H$ , étant donnée une procédure  $q$  de proposition de nouveaux éléments.

### 3.3.3 Calcul efficace des conflits

L'algorithme glouton proposé nécessite de calculer, à chaque ajout d'un nouvel élément  $\mathcal{S}$ , le sous-ensemble optimal  $K^* \subset \mathcal{X}$  que l'on souhaite retirer à  $\mathcal{X}$ . Formulé tel quel, il s'agit d'un problème d'optimisation combinatoire, dont la résolution par une méthode naïve (exploration de l'ensemble des parties de  $\mathcal{X}$ ) nécessiterait des temps de calcul prohibitifs. En effet, ce type de méthode serait de complexité  $\Theta(2^r)$ ,  $r$  étant le nombre d'éléments dans la configuration  $\mathcal{X}$ . Cette complexité est d'autant plus élevée du fait du coût important de l'évaluation de l'énergie.

Afin de pouvoir traiter ce problème efficacement (complexité polynômiale), nous faisons appel à une première heuristique permettant de restreindre l'espace de recherche. Nous choisissons en effet de ne considérer tout d'abord que l'ensemble  $K$  des éléments de  $\mathcal{X}$  qui sont en contact direct avec le nouvel élément  $\mathcal{S}$ , puisque ces éléments sont potentiellement une source de diminution de l'énergie du fait de leur contribution topologique vis-à-vis de  $\mathcal{S}$ . Cette restriction ne permet de calculer qu'une approximation du sous ensemble optimal  $K^*$  exact.



**Figure 3.3:** *Graphes de connexions entre les éléments de  $\mathcal{X}$ , après ajout d'un nouvel élément candidat  $\mathcal{S}$ . Chaque sommet (cercle) représente une primitive de  $\mathcal{X}$ , et une arête relie deux sommets si les primitives correspondantes sont en contact. Notons que le graphe en question peut contenir différentes composantes connexes. Nous proposons, pour approximer efficacement le sous ensemble optimal  $K^*$ , de ne rechercher que parmi l'ensemble  $K$  des éléments directement connectés  $\mathcal{S}$ .*

Pour analyser la qualité de cette approximation, référons-nous à la figure 3.3. Si l'on considère le graphe des connexions entre les primitives des  $\mathcal{X}$ , on s'aperçoit intuitivement que seuls les éléments appartenant à la même composante connexe que  $\mathcal{S}$  sont réellement susceptibles de figurer dans  $K^*$ . En effet, la suppression d'un élément de  $\mathcal{X}$  (prenons  $\mathcal{S}_{10}$  dans l'exemple de la figure 3.3) n'interagissant absolument pas avec  $\mathcal{S}$  ne peut faire diminuer l'énergie que si cet élément est intrinsèquement "mauvais", indépendamment de  $\mathcal{S}$ , c'est à dire si  $H(\mathcal{X} \setminus \{\mathcal{S}_{10}\}) > H(\mathcal{X})$ . Un tel élément n'aurait alors jamais pu être accepté dans la configuration. Il apparaît donc judicieux d'exclure d'ores et déjà les éléments qui ne sont pas connectés (directement ou indirectement) à  $\mathcal{S}$ , puisque par construction de  $\mathcal{X}$ , une modification impliquant ces éléments ne peut pas conduire à une diminution de

l'énergie. Inversement, tout élément appartenant à la même composante connexe que  $\mathcal{S}$  est une source potentielle de diminution de l'énergie par effet de chaîne (en supprimant un élément  $\mathcal{S}'$  connecté à  $\mathcal{S}$ , l'énergie peut encore diminuer en supprimant d'autres éléments connectés à  $\mathcal{S}'$ , etc.), et peut à ce titre apparaître dans  $K^*$ . Cependant, au cours de la reconstruction, il est très probable que le graphe de connexions tende à se compléter, et le problème de la recherche exacte de  $K^*$  en temps polynômial parmi les parties de  $\mathcal{X}$  reste ouvert, justifiant ainsi la restriction à  $K$  au détriment de l'exactitude.

Il est ainsi possible qu'en construisant  $K$ , on exclut prématurément des éléments permettant d'atteindre le minimum théorique. Toutefois, en reprenant le raisonnement précédent, les éléments n'étant pas directement connectés à  $\mathcal{S}$  (prenons  $\mathcal{S}_2$  dans l'exemple de la figure 3.3) et dont la seule suppression diminue l'énergie de la configuration ne devraient pas figurer dans  $\mathcal{X}$  puisqu'ils ne diminuent pas l'énergie cible (au contraire, ils l'augmentent). Mais il se peut que la suppression conjointe de  $\mathcal{S}_2$  et de certains de ses voisins diminue l'énergie de la configuration, ce qui est typiquement le cas lorsque la contribution énergétique de  $\mathcal{S}_2$  n'est imputable qu'à sa bonne connexion avec les voisins en question. La qualité de l'approximation faite en considérant  $K$  est donc dépendante de la fréquence d'apparition de ce type de situations: plus elles apparaissent, plus il est probable que  $K$  soit erroné.

L'erreur que peut introduire cette approximation sur le processus d'optimisation devrait être peu importante: elle peut éventuellement empêcher l'insertion de nouveaux éléments  $\mathcal{S}$  de moindre impact sur l'amélioration de la configuration, mais ne devrait pas empêcher l'ajout de nouveaux éléments contribuant significativement. En effet, un tel échec tiendrait à la seule contribution topologique entre quelques groupes d'éléments de qualité médiocre si l'on se réfère à l'analyse précédente, ce qui semble peu plausible. Quoiqu'il en soit, cette première approximation est une source potentielle d'enfermement du processus dans un minimum local de l'énergie cible.

Partant de  $K$ , il reste à estimer le sous ensemble  $K'$  permettant d'atteindre l'énergie la plus basse. Lorsque la taille de  $K$  est raisonnable ( $|K| \leq 5$ ), il est envisageable d'adopter une méthode "force brute", explorant la totalité des  $2^{|K|}$  possibilités. Dans le cas contraire, nous proposons d'adopter une heuristique gloutonne introduisant un second niveau d'approximation de  $K^*$ , mais permettant d'aboutir à une solution en temps quadratique de  $|K|$ . L'idée est relativement intuitive, et consiste en la construction itérative du sous ensemble  $K'$  en considérant les éléments de  $K$  par ordre inverse de contribution énergétique. On cherche donc à enlever les éléments les moins pertinents d'abord, en gardant la trace du sous ensemble  $K'$  permettant d'atteindre l'énergie la plus basse parmi toutes les solutions explorées. L'algorithme 6 décrit en détail le processus conduisant au calcul de  $K'$  approximant  $K^*$ .

**Remarque** d'après l'expression de l'énergie  $H$ , la suppression d'un élément  $\mathcal{S}_i \in \mathcal{X}$  n'est susceptible de modifier la contribution énergétique d'un second élément  $\mathcal{S}_j \in \mathcal{X}$  que si  $\mathcal{S}_i$  et  $\mathcal{S}_j$  sont connectés. Autrement dit:

$$">\Delta H(\mathcal{X}) - \Delta H(\mathcal{X} \setminus \mathcal{S}_j) \neq \Delta H(\mathcal{X} \setminus \mathcal{S}_i) - \Delta H((\mathcal{X} \setminus \mathcal{S}_i) \setminus \mathcal{S}_j)" \Rightarrow " \mathcal{S}_i \text{ et } \mathcal{S}_j \text{ sont connectés}"$$

Ceci tient aux faits que:

- chaque primitive de  $\mathcal{X}$  apporte une énergie géométrique  $H_G$  indépendamment des autres primitives. La suppression d'un objet ne modifie donc en rien la contribution géométrique des autres primitives de  $\mathcal{X}$ ,

---

**Algorithme 6** : Algorithme heuristique approximant l'ensemble optimal des conflits.
 

---

**Données** :  $\mathcal{X}$ : configuration actuelle  
 $\mathcal{S}$ : élément candidat à l'insertion dans  $\mathcal{X}$   
 $H$ : énergie définissant la pertinence de toute configuration  
**Résultat** :  $K'$ : approximation de l'ensemble de conflits optimal  $K^*$  vis-à-vis de  $\mathcal{S}$

```

1  $K \leftarrow$  ensemble des éléments de  $\mathcal{X}$  en contact avec  $\mathcal{S}$ ;
2  $\mathcal{X}' \leftarrow \mathcal{X} \cup \{\mathcal{S}\}$ ;
3  $K' \leftarrow \emptyset$ ;  $Q \leftarrow \emptyset$ ;
4  $H_0 \leftarrow H(\mathcal{X}')$ ;
5 Pour tout  $\mathcal{S}_i \in K$  faire                                /* Boucle d'initialisation */
6   | Précalculer la contribution de  $\mathcal{S}_i$ :  $\Delta H(\mathcal{S}_i) \leftarrow H(\mathcal{X}') - H(\mathcal{X}' \setminus \{\mathcal{S}_i\})$ 
7 Tant que  $K \neq \emptyset$  faire
8   |  $\mathcal{S}_i \leftarrow \operatorname{argmax}_{\mathcal{S}_k \in K} \Delta H(\mathcal{S}_k)$ ;          /* Sélection du plus "mauvais" élément */
9   |  $Q \leftarrow Q \cup \{\mathcal{S}_i\}$ ;
10  |  $\mathcal{X}' \leftarrow \mathcal{X}' \setminus \{\mathcal{S}_i\}$ ;
11  | Si  $H(\mathcal{X}') < H_0$  alors                                /* Sauvegarde des conflits optimaux */
12  |   |  $K' \leftarrow Q$ ;
13  |   |  $H_0 \leftarrow H(\mathcal{X}')$ 
14  | Pour tout  $\mathcal{S}_j \in K$  connecté à  $\mathcal{S}_i$  faire          /* Boucle de mise à jour */
15  |   | Mettre à jour la contribution de  $\mathcal{S}_j$ :  $\Delta H(\mathcal{S}_j) \leftarrow H(\mathcal{X}') - H(\mathcal{X}' \setminus \{\mathcal{S}_j\})$ 
16  |   |  $K \leftarrow K \setminus \mathcal{S}_i$ ;
17 Retourner  $K'$ 

```

---

- l'énergie topologique  $H_T$  n'est définie que sur les paires d'éléments connectés. La suppression d'une primitive ne modifie donc la contribution d'une autre que si elles sont connectées,
- l'énergie d'attache aux données  $H_D$  est telle que la suppression d'un élément n'interfère avec l'apport énergétique d'un autre élément que si les deux objets partagent un point du nuage (s'ils sont donc connectés).

Ce constat permet de simplifier la boucle de mise à jour des contributions énergétiques (ligne 14). Lorsque l'on supprime un élément de  $\mathcal{S}_i \in K$ , seules les contributions énergétiques des primitives de  $K$  connectées à  $\mathcal{S}_i$  doivent être révisées.

### 3.4 Traitement des coudes

Dans le cas des installations industrielles qui nous intéresse ici, le tores jouent pour la plupart le rôle de coudes reliant les éléments du tuyauterie. À ce titre, il paraît judicieux de ne les traiter qu'une fois les cylindres convenablement détectés, à l'aide d'une approche déterministe cette fois-ci, de manière à s'appuyer sur le modèle CAO reconstruit pour le compléter aux jonctions des cylindres. Ce post-traitement ne s'inscrit pas vraiment dans la démarche de la minimisation de l'énergie, mais permet de compléter la reconstruction en s'appuyant sur un *a priori* fort quant au rôle des tores dans les scènes étudiées.

L'algorithme de reconstruction des coudes est relativement simple. Il s'agit, pour chaque paire de cylindres connectés dont le rayon est sensiblement équivalent, de construire

le coude décrivant au mieux les points du nuage au niveau de la jonction des cylindres. On estimera qu'un couple de cylindres connectés est susceptible de présenter une jonction torique si la différence de leurs rayons n'excède pas le double du seuil de bruit  $\epsilon$ , servant *a posteriori* d'estimation pour la précision des formes reconstruites. La section de tore ainsi reconstruite n'est acceptée (et intégrée au modèle CAO reconstruit  $\mathcal{X}$ ) que si l'énergie d'attache aux données résultante diminue.

### 3.4.1 Déterminer les points à la jonction des cylindres

Pour reconnaître avec précision un coude décrit par le nuage de points, il nous faut avant tout identifier le sous-ensemble du nuage qui décrive effectivement un tore à la jonction de la paire de cylindres  $(\mathcal{S}_i, \mathcal{S}_j)$  testée. Notons que l'on utilise ici les portions effectives de primitives, telles que décrites en section 4.3.3.

Le point de jonction  $\mathbf{j}$  de cylindres se définit comme l'intersection de leurs axes définis par les couples  $(\mathbf{c}_{\mathcal{S}_i}, \mathbf{a}_{\mathcal{S}_i})$  et  $(\mathbf{c}_{\mathcal{S}_j}, \mathbf{a}_{\mathcal{S}_j})$ , où  $\mathbf{c}_{\mathcal{S}}$  et  $\mathbf{a}_{\mathcal{S}}$  désignent respectivement le centre et le vecteur directeur de l'axe du cylindre  $\mathcal{S}$ . L'intersection exacte n'existe pas forcément dans  $\mathbb{R}^3$ , mais on peut calculer le point minimisant la distance aux deux droites ainsi définies, en prenant le barycentre des points  $\mathbf{q}_i$  et  $\mathbf{q}_j$  défini en équation 2.33.

Une fois la jonction  $\mathbf{j}$  calculée, le traitement peut être immédiatement stoppé si l'on constate que le point de jonction se situe à l'intérieur de l'un des deux cylindres. En effet, cette situation correspond à une connexion en "T" des cylindres, et ne nécessite donc pas de jonction torique. Seules les connexions en "L" sont visées par ce traitement, et se caractérisent par la présence du point de jonction au delà des extrémités des deux cylindres, puisque la construction des portions effectives de primitives permet de s'assurer que les sections cylindriques ne débordent pas sur les points définissant le coude.

Partant de  $\mathbf{j}$ , on identifie les deux disques correspondant aux extrémités des cylindres lui faisant face. Soient  $(\mathbf{s}_{\mathcal{S}_i}, \widetilde{\mathbf{a}}_{\mathcal{S}_i})$  et  $(\mathbf{s}_{\mathcal{S}_j}, \widetilde{\mathbf{a}}_{\mathcal{S}_j})$  les centres et vecteur normaux de ces disques. L'intersection des plans portant ces disques définit une droite, sur laquelle on projette les points  $\mathbf{s}_{\mathcal{S}_i}$  et  $\mathbf{s}_{\mathcal{S}_j}$ , puis on définit le point  $\mathbf{b}$  comme étant le barycentre de ces points projetés. Ce point  $\mathbf{b}$  satisfait donc le système d'équations:

$$\begin{cases} (\mathbf{b} - \mathbf{s}_{\mathcal{S}_i}) \cdot \widetilde{\mathbf{a}}_{\mathcal{S}_j} = 0 \\ (\mathbf{b} - \mathbf{s}_{\mathcal{S}_j}) \cdot \widetilde{\mathbf{a}}_{\mathcal{S}_i} = 0 \\ (\mathbf{s}_{\mathcal{S}_i} - \mathbf{b}) \cdot \mathbf{x} + (\mathbf{s}_{\mathcal{S}_j} - \mathbf{b}) \cdot \mathbf{x} = 0 \end{cases}$$

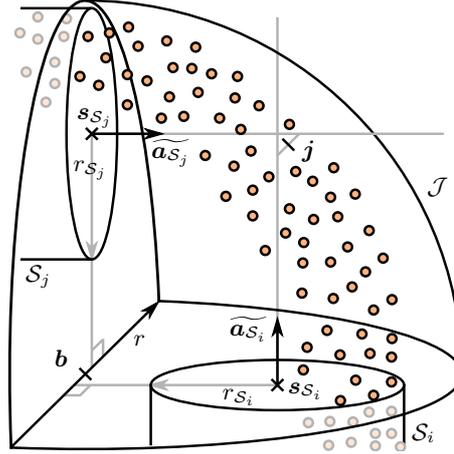
avec  $\mathbf{x} = \widetilde{\mathbf{a}}_{\mathcal{S}_i} \times \widetilde{\mathbf{a}}_{\mathcal{S}_j}$ . Ce système linéaire de trois équations à trois inconnues (les coordonnées de  $\mathbf{b}$ ) peut être résolu simplement à l'aide des approches usuelles d'algèbre linéaire.

Les points échantillonnant l'éventuel coude joignant les cylindres  $\mathcal{S}_i$  et  $\mathcal{S}_j$  peuvent alors être isolés en utilisant la portion de sphère centrée en  $\mathbf{b}$  délimitée par les deux plans portant les disques mentionnés ci-avant. On obtient alors l'ensemble :

$$\mathcal{J} = \{\mathbf{p} \in \mathcal{P} \mid \|\mathbf{p} - \mathbf{b}\| \leq r \text{ et } (\mathbf{p} - \mathbf{s}_{\mathcal{S}_k}) \cdot (\mathbf{j} - \mathbf{s}_{\mathcal{S}_k}) \geq 0 \quad \forall \mathcal{S}_k \in \{\mathcal{S}_i, \mathcal{S}_j\}\}$$

avec le rayon  $r$  défini comme suit:

$$r = \max(\|\mathbf{s}_{\mathcal{S}_i} - \mathbf{b}\| + r_{\mathcal{S}_i}, \|\mathbf{s}_{\mathcal{S}_j} - \mathbf{b}\| + r_{\mathcal{S}_j}) + \eta\epsilon$$



**Figure 3.4:** Sélection des points pour la construction de la jonction torique de deux cylindres  $S_i$  et  $S_j$  connectés. Les points appartenant à l'ensemble  $\mathcal{J}$  retenu figurent en orange, tandis que les autres sont grisés.

### 3.4.2 Calcul du tore approximant les points

Le coude reliant les deux cylindres  $S_i$  et  $S_j$  est construit comme étant la primitive  $\mathcal{S}$  approximant au mieux les points de  $\mathcal{J}$  au sens des moindres carrés. Cette tâche se pose comme un problème de minimisation (cf. section 4.2.3 - équation 4.9) pouvant être résolu à l'aide de l'algorithme de Levenberg-Marquardt (cf. thèse de T. Chaperon [Cha02] pour la distance d'un point à un tore et ses dérivées). Or, cet algorithme doit être initialisé à l'aide d'une estimation proche de la solution recherchée. Il nous faut donc, avant de calculer ce raffinement, trouver un tore approximant grossièrement les données.

Pour ce faire, on peut utiliser les cylindres disponibles. Le tore  $\mathcal{S}$  reliant les cylindres possède en effet un axe  $\mathbf{a}_{\mathcal{S}}$  orthogonal aux vecteurs directeurs  $\mathbf{a}_{S_i}$  et  $\mathbf{a}_{S_j}$  des axes de  $S_i$  et  $S_j$ . On a donc, en première estimation :

$$\mathbf{a}_{\mathcal{S}} = \mathbf{a}_{S_i} \times \mathbf{a}_{S_j}$$

De même, le rayon mineur  $r_{\mathcal{S}}$  du tore le plus approprié est la moyenne des rayons des cylindres:

$$r_{\mathcal{S}} = \frac{r_{S_i} + r_{S_j}}{2}$$

Il reste finalement à déterminer le grand rayon  $R_{\mathcal{S}}$  ainsi que le centre  $\mathbf{c}_{\mathcal{S}}$  du tore. Dans le plan orthogonal à l'axe  $\mathbf{a}_{\mathcal{S}}$ , ce problème possède 3 degrés de liberté (2 pour le centre, un pour le rayon). Il nous faut donc 3 contraintes pour déterminer ces paramètres. Tout d'abord, le cercle défini par le centre  $\mathbf{c}_{\mathcal{S}}$  du tore et son grand rayon  $R_{\mathcal{S}}$  projetés dans le plan orthogonal à  $\mathbf{a}_{\mathcal{S}}$  doit être tangent aux axes des cylindres  $S_i$  et  $S_j$  dans ce même plan, formant ainsi les deux premières contraintes. Le dernier degré de liberté peut être contraint en identifiant une donnée supplémentaire. Pour ce faire, on se propose d'identifier parmi les points de  $\mathcal{J}$  une section cylindrique approximant une partie du tore. Cette section cylindrique, de par son axe, définit une troisième contrainte de tangence pour le cercle recherché. Le centre  $R_{\mathcal{S}}$  et le grand rayon  $r_{\mathcal{S}}$  se calculent alors respectivement comme le point équidistant des trois droites contraignantes, et la distance à ces trois droites.

Ainsi, si les trois droites contraignantes plongées dans le plan orthogonal à  $\mathbf{a}_{\mathcal{S}}$  se

définissent comme l'ensemble des  $\mathbf{x} \in \mathbb{R}^2$  satisfaisant respectivement les équations

$$\begin{cases} (\mathbf{x} - \mathbf{c}_0) \cdot \mathbf{d}_0 = 0 \\ (\mathbf{x} - \mathbf{c}_1) \cdot \mathbf{d}_1 = 0 \\ (\mathbf{x} - \mathbf{c}_2) \cdot \mathbf{d}_2 = 0 \end{cases} \quad (3.9)$$

sous la contrainte d'orientation des vecteurs  $\mathbf{d}_i \in \mathbb{R}^2$

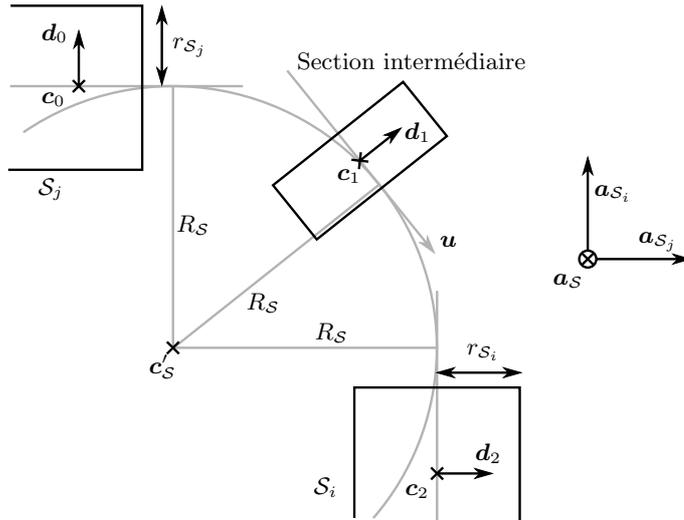
$$\forall i, j \in \{0, 1, 2\} \text{ avec } i \neq j : (\mathbf{c}_j - \mathbf{c}_i) \cdot \mathbf{d}_i < 0$$

alors le centre  $\mathbf{c}'_{\mathcal{S}} \in \mathbb{R}^2$  et le rayon  $R_{\mathcal{S}}$  sont calculés en résolvant le système linéaire suivant:

$$\begin{cases} (\mathbf{c}'_{\mathcal{S}} - \mathbf{c}_0) \cdot \mathbf{d}_0 = -R_{\mathcal{S}} \\ (\mathbf{c}'_{\mathcal{S}} - \mathbf{c}_1) \cdot \mathbf{d}_1 = -R_{\mathcal{S}} \\ (\mathbf{c}'_{\mathcal{S}} - \mathbf{c}_2) \cdot \mathbf{d}_2 = -R_{\mathcal{S}} \end{cases}$$

via les outils habituels d'algèbre linéaire. Les équations 3.9 peuvent être trouvées relativement facilement, après quelques opérations géométriques élémentaires de projection dans le plan des points et vecteurs de  $\mathbb{R}^3$  définissant les axes contraignants (cf. figure 3.5).

La position du centre  $\mathbf{c}_{\mathcal{S}}$  dans  $\mathbb{R}^3$  peut être déduite de  $\mathbf{c}'_{\mathcal{S}} \in \mathbb{R}^2$ , en estimant que le plan considéré contient le point  $\mathbf{j}$  de jonction des cylindres.



**Figure 3.5:** Initialisation des paramètres du tore en vue de calculer le coude s'ajustant au mieux sur les points du nuage. Une fois la section cylindrique intermédiaire calculée, on dispose de 3 droites tangentes permettant de retrouver simplement une estimation du centre  $\mathbf{c}'_{\mathcal{S}}$  et du rayon  $R_{\mathcal{S}}$  dans le plan. On note que tout les vecteurs  $\mathbf{d}_i$  définissant en partie les contraintes sont orientés vers "l'extérieur" du cercle recherché.

Pour identifier la section cylindrique intermédiaire, on se donne un vecteur directeur  $\mathbf{u} \in \mathbb{R}^3$  initial, calculé à partir de la bissectrice des axes de  $\mathcal{S}_i$  et  $\mathcal{S}_j$ . En supposant que les vecteurs axiaux  $\mathbf{a}_{\mathcal{S}_i}$  et  $\mathbf{a}_{\mathcal{S}_j}$  des cylindres  $\mathcal{S}_i$  et  $\mathcal{S}_j$  pointent vers la jonction  $\mathbf{j}$ , on a alors

$$\mathbf{u} = \frac{\mathbf{a}_{\mathcal{S}_i} - \mathbf{a}_{\mathcal{S}_j}}{\|\mathbf{a}_{\mathcal{S}_i} - \mathbf{a}_{\mathcal{S}_j}\|}$$

En identifiant dans la sphère gaussienne de  $\mathcal{J}$  (le nuage de normales associée aux points de  $\mathcal{J}$ ) l'ensemble des normales orthogonales à  $\mathbf{u}$  à une tolérance près, on isole les points

décrivant un cylindre dont l'axe est  $\mathbf{u}$ . Une détection de cercle parmi les points retenus après qu'ils aient été projetés dans le plan orthogonal à  $\mathbf{u}$  permet de calculer un point de passage  $\mathbf{c}_1$  de l'axe du cylindre intermédiaire. La droite engendrée par  $\mathbf{u}$  et  $\mathbf{c}_1$  définit ainsi la troisième contrainte nécessaire au calcul des paramètres initiaux du tore.

Une fois le tore convenablement initialisé, on peut alors calculer le coude en résolvant le problème de minimisation au sens des moindres carrés mentionné précédemment, puis en calculant l'angle d'ouverture du tore permettant aux extrémités du coude de coïncider avec les extrémités des cylindres  $\mathcal{S}_i$  et  $\mathcal{S}_j$  (il est bien entendu nécessaire de modifier les extrémités des cylindres en conséquence). L'acceptation du tore s'accompagne de plus de la suppression de tous les cylindres du modèle CAO reconstruit qui le chevauchent. Pour détecter ces chevauchements, on utilise l'approximation du volume d'intersection décrite précédemment (section 2.3.3) avec un seuil égal à 50% du volume du cylindre testé.

## 3.5 Résultats

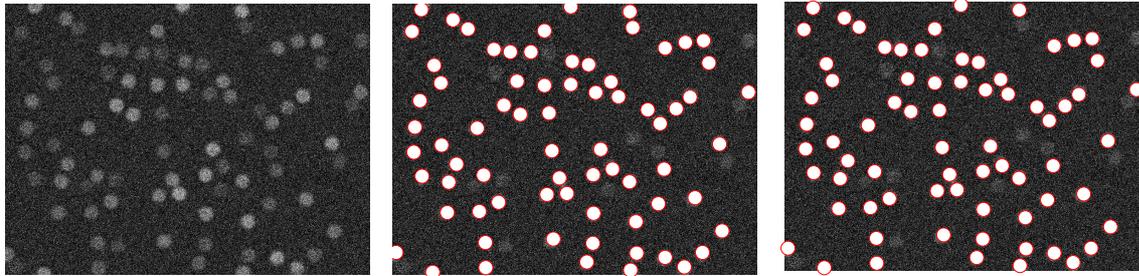
### 3.5.1 Expérimentation pour la détection de formes dans des images

Afin d'étudier la capacité de l'approche gloutonne à traiter convenablement les problèmes de détection de forme, il peut être intéressant d'évaluer le comportement de cette méthode dans le cadre de la détection de disques dans des images. Ceci permet, dans un premier temps, de comparer les résultats obtenus avec ceux fournis par l'algorithme 4. La figure 3.6 présente les résultats obtenus par les deux algorithmes sur un exemple synthétique. L'énergie d'attache aux données  $H_D$  utilisée est celle proposée par Florent Lafarge et al. [LGD10]. Elle repose sur la mesure de la différence entre les distributions des valeurs de pixels à l'intérieur et l'extérieur de la forme, dans une couronne d'épaisseur fixe (2 pixels). Cette mesure s'inspire de la distance de Mahalanobis, et donne d'excellents résultats. Quant à l'énergie de cohérence topologique  $H_T$ , nous utilisons celle proposée par Xavier Descombes et al. [DMZ08], pénalisant fortement les disques se recouvrant.

Le noyau de proposition qu'utilise l'algorithme glouton pour cette application est un échantillonneur de Metropolis-Hastings (cf. section 3.2.1) simulant une densité de probabilité proportionnelle à  $e^{-H_D}$ . La probabilité pour un disque d'être généré est donc directement dépendante de sa pertinence vis-à-vis de l'image, et les disques étant effectivement dans l'image apparaissent donc plus fréquemment. Notons que dans l'exemple de la figure 3.6, les rayons des disques sont supposés connus (conformément à l'application décrite dans [DMZ08]), et seules les positions des disques doivent être réellement générées. De plus, l'algorithme 5 tel qu'il est présenté doit être légèrement adapté pour cette application : en effet, nous ne disposons pas ici de modèle *a priori*. Nous supprimons donc simplement la ligne 4 de l'algorithme.

La figure 3.7 présente l'évolution des différentes instances de minimisation testées, tandis que le tableau 3.1 dresse un bilan des résultats obtenus. Les temps de calculs (tableau 3.1) ainsi que la vitesse de convergence (figure 3.7) ne sont mentionnés qu'à titre indicatif. En effet, chacun des algorithmes testés repose sur un paramètre d'arrêt déterminant le "délai" (en nombre d'itérations) au delà duquel l'algorithme juge être parvenu à un état stable. Or, les deux algorithmes reposent sur des notions d'itérations très différentes. Donc ce paramètre de stabilité n'a pas exactement la même signification dans les deux algorithmes. En outre, la différence dans la définition de ce qu'est une itération rend complexe la comparaison des vitesses d'évolution de l'énergie. Finalement, bien qu'il semble

évident que l'algorithme glouton n'est pas plus rapide que l'algorithme BDMCMC, il n'est pas pertinent de conclure quand à l'efficacité relative de ces deux méthodes. Ces résultats montrent uniquement que la méthode gloutonne est, sur cet exemple, au moins aussi fiable que la méthode BDMCMC pour traiter le problème de détection de disques dans une image, puisque les deux approches atteignent une énergie minimale équivalente, et les résultats obtenus sont visuellement très semblables.



(a) Image traitée: 84 disques de rayon 20 dans  $1024 \times 768$  pixels. (b) Détection de disques par méthode gloutonne: 72 disques trouvés (c) Détection de disques par BDMCMC (algorithme 4): 71 disques trouvés

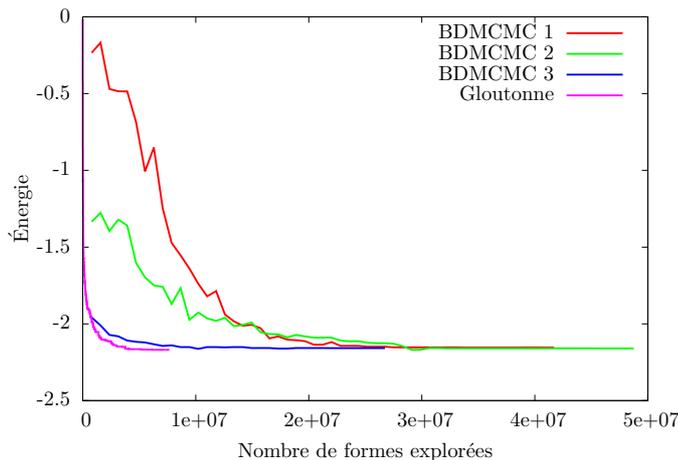
**Figure 3.6:** Détection de disques dans une image synthétique fortement bruitée. L'image est générée par impression de disques uniformes de diverses intensités, de rayon constant et positionnés aléatoirement sur une image de fond noire. L'image est ensuite fortement bruitée (bruit gaussien d'écart type 70), floutée par un filtre moyenneur, puis bruitée à nouveau. L'algorithme glouton proposé semble en mesure de détecter des disques ne se recouvrant pas, et les résultats sont comparables, qualitativement, avec ceux obtenus par l'algorithme 4.

BDMCMC				Temps de calcul (s)	Énergie finale	Disques
$T_0$	$F_T(T)$	$dt_0$	$f_{dt}(dt)$			
50	$T/1.1$	50	$0.99 \times dt$	173	-2.15773	71
50		5		32	-2.15725	71
500		5		28	-2.15631	70
Méthode gloutonne				141	-2.16921	72

**Table 3.1:** Tableau récapitulatif des résultats de détection de disques disjoints obtenus avec les différentes méthodes testées sur l'image synthétique (cf. figure 3.6). Pour la méthode BDMCMC, seuls les résultats correspondant à la première ligne sont illustrés en figure 3.6, puisqu'ils sont vraisemblablement les meilleurs. Ce tableau confirme que la méthode gloutonne est au moins aussi fiable (pour cette application) que l'approche BDMCMC étudiée.

La figure 3.8 et le tableau 3.2 présentent les résultats obtenus sur des images réelles. Ceux-ci confirment la capacité de l'approche gloutonne à détecter les formes dans les images. On peut même constater une légère supériorité de l'approche gloutonne sur les méthodes BDMCMC, dans la mesure où elle permet d'atteindre des énergies plus basses. Toutefois, cette remarque doit être prise avec beaucoup de précautions : rien ne garantit que les paramétrages testés pour la méthode BDMCMC soient les plus adaptés, et il est probable que cette approche soit en mesure de fournir de meilleurs résultats que ceux présentés, bien que nous montrions ici les meilleurs résultats obtenus (les paramétrages utilisés ont été déterminés par essais/erreurs).

La force majeure de la méthode gloutonne vis-à-vis des l'algorithme BDMCMC ré-



**Figure 3.7:** Évolution de l'énergie au cours du processus de minimisation, pour les différents cas présentés dans le tableau 3.1. "BDMCMC  $x$ " fait référence au  $x^{\text{ème}}$  paramétrage de l'algorithme 4. La notion d'itération dans ces algorithmes ne permet pas de comparer leur vitesse de convergence: une itération de l'algorithme BDMCMC est une tentative d'ajout d'un candidat pour chaque pixel de l'image, alors qu'une itération de l'algorithme glouton est une tentative d'ajout d'un unique candidat. Pour permettre la comparaison, nous étudions donc l'évolution de l'énergie en fonction du nombre de formes testées pour l'insertion dans la configuration.

side dans la simplicité de son paramétrage : aucun paramètre n'est requis, alors que l'algorithme d'optimisation BDMCMC nécessite de spécifier 4 entrées. Pour ajuster convenablement ces 4 paramètres en vue d'obtenir des résultats satisfaisants, de nombreux tests sont nécessaires, rendant la tâche très pénible.

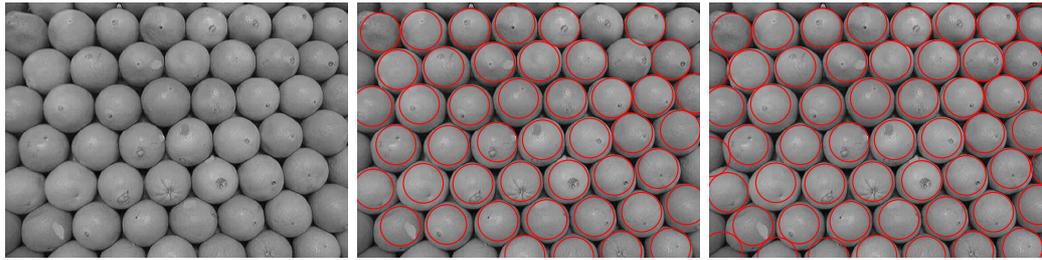
Jeu de test		Temps de calcul (s)	Énergie finale	Disques
Oranges	BDMCMC	131	-3.93429	45
	Gloutonne	1462	-4.55158	54
Partie de go	BDMCMC	18	-0.733139	62
	Gloutonne	64	-0.783895	72

**Table 3.2:** Tableau récapitulatif des résultats de détection de disques disjoints obtenus par les méthodes testées sur des images réelles (cf. figure 3.8). La quatrième colonne les énergies atteintes par la méthode gloutonne et les BDMCMC. Encore une fois, précisons que de par la difficulté de son paramétrage, il se peut que la méthode BDMCMC soit en mesure de produire de meilleurs résultats, mais ce tableau ne montre que les meilleurs résultats obtenus pour cette méthode sur les paramétrages que nous avons testés ("à la main").

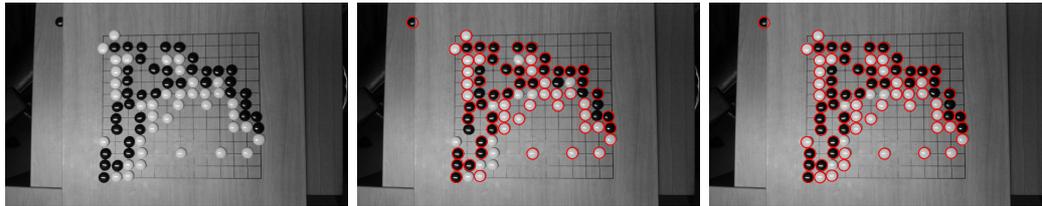
### 3.5.2 Expérimentations sur des données 3D réelles

L'algorithme d'optimisation 5 doit de plus être testé sur des données tridimensionnelles réelles. On utilise pour cela le générateur de primitives décrit par la suite (chapitre 4), ainsi que l'énergie  $H$  introduite précédemment (chapitre 2).

La figure 3.9 présente les données utilisées pour la première expérience illustrée ici, ainsi que les résultats obtenus. Le nuage de points et le modèle *a priori* sont initialement recalés grossièrement, conformément à l'illustration 2.2, et quelques différences significatives dans la position et l'orientation des tuyaux sont visibles. De plus, bon nombre de composants du modèle CAO *a priori* ne sont pas représentés dans le nuage de points. Inversement, la



(a) Oranges (66 oranges)



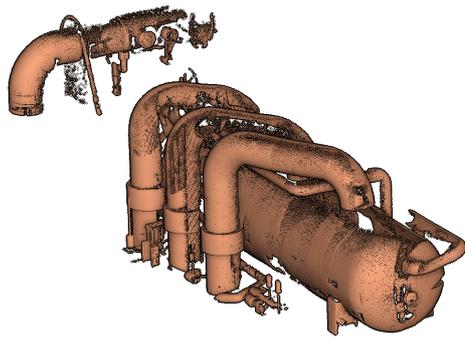
(b) Partie de go (79 pierres sur le plateau, plus une en dehors)

**Figure 3.8:** Tests sur quelques images réelles. Les résultats confirment l'aptitude de l'algorithme glouton à détecter des formes sous contrainte topologique (non intersection des formes). A gauche figure l'image testée, au milieu les résultats de l'algorithme 4, à droite les résultats de l'algorithme glouton. Pour l'algorithme BDMCMC, nous ne conservons que le meilleur résultat, c'est à dire celui dont le paramétrage (parmi ceux testés) permet d'aboutir à l'énergie la plus basse.

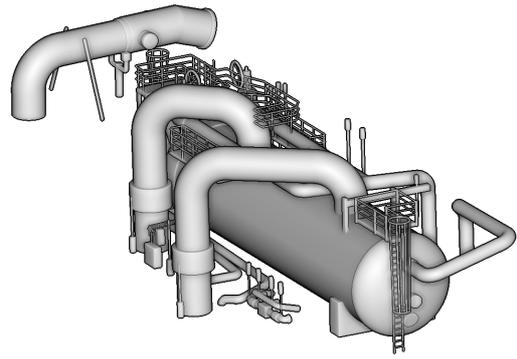
conduite verticale centrale de la scène échantillonnée est absente du modèle CAO *a priori*. Lors de l'optimisation, on estime ici que l'algorithme est stable lorsque la solution en construction reste inchangée durant  $N = 20$  boucles successives de l'algorithme de résolution 5. On utilise ici une tolérance au déplacement de  $\sigma_p = 3$  mètres, une tolérance au désaxage de  $\sigma_o 20^\circ$  et une tolérance au changement de diamètre des cylindres égale à  $\sigma_g = 20\%$  de leur diamètre *a priori*. Le seuil de bruit est ici fixé à  $\epsilon = 3$  centimètres. Les coefficients de pondération  $\lambda_D = 2 \times 10^{-4}$ ,  $\lambda_T = 2$  et  $\lambda_G = 1$  ont été estimées à partir de valeurs suggérées automatiquement par la méthode que nous avons présentée en section 2.3.4.

On constate que les principales primitives ont été détectées convenablement, et que la topologie du résultat satisfait effectivement les contraintes métier spécifiées par les règles d'assemblage. La méthode ne parvient toutefois pas à traiter les petits éléments de cette scène.

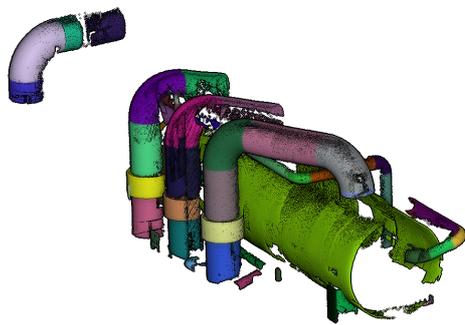
L'obtention de ces résultats nécessite deux heures de calcul environ sur processeur cadencé à 2.3 Ghz. Toutefois, des résultats sensiblement équivalents peuvent être obtenus en un temps beaucoup plus court, en diminuant le paramètre  $N$  servant à évaluer la stabilité du processus. En effet, la minimisation de l'énergie présente un profil exponentiel, et les premières étapes du processus permettent d'atteindre rapidement une énergie très basse, tandis que les dernières étapes ne constituent qu'un affinement léger de la solution au sens de l'énergie  $H$ . On constate par exemple sur les données de la figure 3.10 que les premières 20% d'itérations du processus permettent d'atteindre 57% de l'énergie finale. Par ailleurs, l'étude visuelle de l'évolution de la solution reconstruite au cours du processus illustré en figure 3.12 confirme la rentabilité de la méthode dans sa première phase, puisqu'un résultat visuellement proche du résultat final est obtenu après 30000 itérations sur les 329482 itérations que compte la reconstruction totale. En posant  $N = 1$  au lieu de  $N = 20$ , la méthode estime avoir convergé après 7 minutes de calcul seulement, et fournit cependant des résultats visuellement cohérents (cf. figure 3.11). Le résultat est malgré



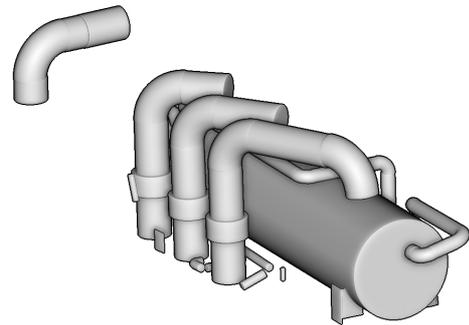
(a) Nuage de points (990323 points)



(b) Modèle CAO a priori (1006 primitives dont 391 cylindres, 456 plans et 141 tores)



(c) Segmentation (821446 points)



(d) Modèle CAO reconstruit (74 primitives dont 52 cylindres, 6 plans et 16 tores)

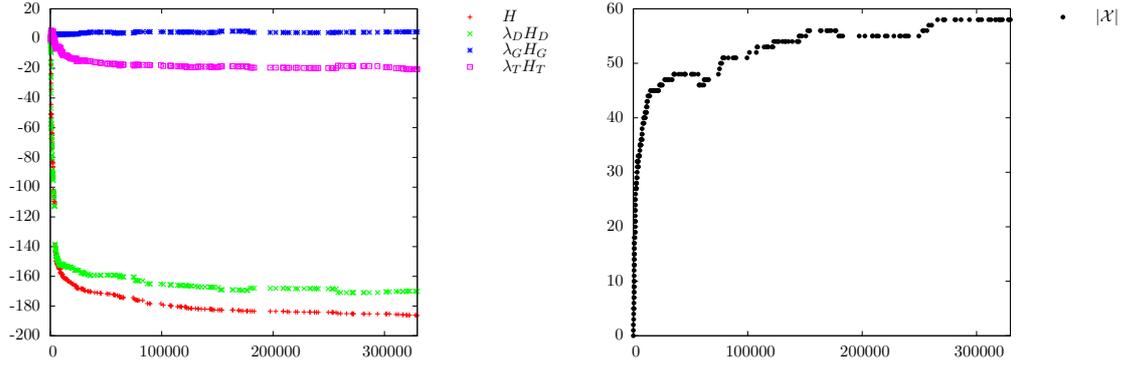
**Figure 3.9:** Reconstruction d'un composant issu d'une scène centrale électrique. Les images supérieures représentent les données en entrée du problème, tandis que les images inférieures illustrent les résultats obtenus. La segmentation illustrée ici correspond à la coloration de chaque point du nuage en fonction de la primitive à laquelle il est associé (une couleur aléatoire par primitive). Un point est associé à la primitive dont il est le plus proche, si la distance entre le point et la primitive est inférieure au seuil  $\epsilon$ .

tout relativement éloigné quantitativement de la solution optimale, comme le montre le tableau 3.3. On note en particulier que la majeure partie des efforts restants concerne l'énergie topologique  $H_T$ .

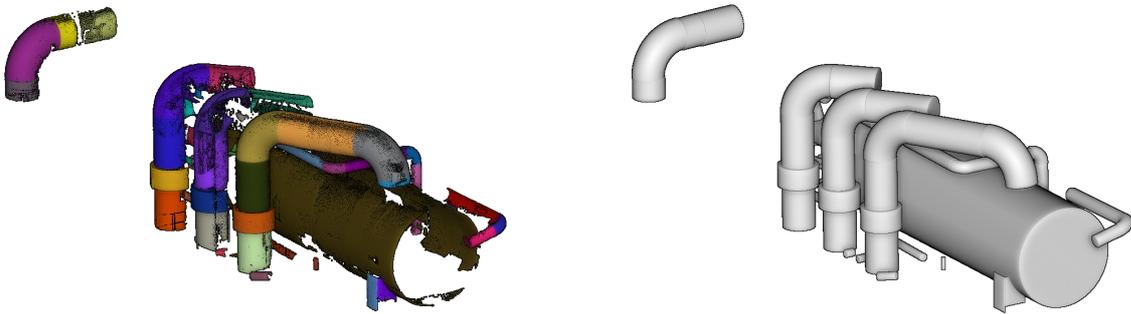
	Itération	$ \mathcal{X} $	$H$	$\lambda_D H_D$	$\lambda_G H_G$	$\lambda_T H_T$
$N = 1$	16925	45	-163.311	-153.927	2.97396	-12.3584
$N = 20$	329482	58	-186.389	-170.212	4.46394	-20.6413
proportion	5.1%	77.6%	87.6%	90.4%	66.6%	59.9%

**Table 3.3:** Comparaison des résultats obtenus en fonction du paramètre de stabilité  $N$ . La dernière ligne illustre l'avancement de la méthode avec  $N = 1$  par rapport aux résultats obtenus avec  $N = 20$ .

Le tableau 3.4 présente la variabilité des résultats obtenus sur différentes exécutions avec un même paramétrage. Du fait de sa nature stochastique, l'algorithme ne permet pas nécessairement d'aboutir systématiquement au même résultat. La probabilité d'aboutir à un même résultat dépend notamment du comportement du générateur de primitives: pour obtenir rigoureusement le même résultat, il est nécessaire que le générateur de prim-



**Figure 3.10:** Évolution de l'énergie (gauche) et du nombre de primitives dans la solution (droite) au cours de l'optimisation.



**Figure 3.11:** Résultats (segmentation à gauche et modèle CAO reconstruit à droite) obtenus après l'arrêt prématuré du processus d'optimisation sur les données présentées en figure 3.9, en posant  $N = 1$ .

itives ait exploré les mêmes objets pour chaque solution, ce qui peut nécessiter une grande quantité d'itérations. En outre, l'aspect glouton de l'algorithme de minimisation peut le conduire à retenir une partie de la solution qui le bloque dans un minimum local de la fonction cible, et empêche donc l'obtention d'un résultat commun sur les différentes exécutions de l'algorithme. Toutefois, les résultats sont visuellement très semblables, et ne se distinguent sur cet exemple que dans la quantité des objets reconnus. En particulier, les composants majeurs sont tous présents et quasiment identiques. Finalement, les différences entre les énergies extrémales observées sur ces 8 tests ne constituent que 4.3% de l'énergie moyenne obtenue, ce qui constitue une marge d'erreur raisonnable.

Le deuxième exemple que nous présentons (figure 3.13) concerne une salle de centrale électrique. Nous considérons cette fois-ci un modèle CAO *a priori* incomplet, et spécifions des tolérances au changement  $\sigma_p$ ,  $\sigma_o$  et  $\sigma_g$  importantes pour étudier la capacité de l'algorithme à reconstruire un modèle CAO le plus complet possible malgré la relative pauvreté des informations *a priori*. Ainsi,  $\sigma_p = 3\text{m}$  correspond approximativement à la largeur de la scène,  $\sigma_o = 30^\circ$  doit *a priori* permettre de retrouver la plupart des cylindres, et  $\sigma_g = 100\%$  du rayon des cylindres ne contraint que très peu la taille des tuyauteries recherchées.

Le paramètre  $N$  de stabilité doit, dans cet exemple, se voir affecter une grande valeur (nous utilisons  $N = 1000$  ici). En effet, étant donné le faible nombre de primitives du modèle CAO *a priori*, il est très probable qu'une boucle de l'algorithme ne permette pas d'améliorer la solution, malgré la disponibilité de composants restant à découvrir dans le nuage de points. Pour tenir compte des imperfections de la méthode d'exploration de

l'espace des solutions (le générateur de primitives), on augmente donc le nombre d'échecs consécutifs tolérés, de manière à permettre le traitement de la plus grande partie possible de la scène.

Les résultats montrent que la majeure partie des composants de cette scène sont bien reconstruits par l'algorithme. Il existe toutefois une très petite quantité de primitives générées et acceptées qui ne semblent pas pertinentes vis-à-vis de la scène. Cet écueil tient notamment à la formulation des énergies, et principalement de l'énergie d'attache aux données  $H_D$  qui, malgré son fort pouvoir discriminant, ne permet pas de refuser une forme qui s'ajuste bien sur une petite portion isolée de surface échantillonnée. On peut considérer ici que le paramètre de couverture minimale  $\gamma_0$  utilisé est légèrement trop faible

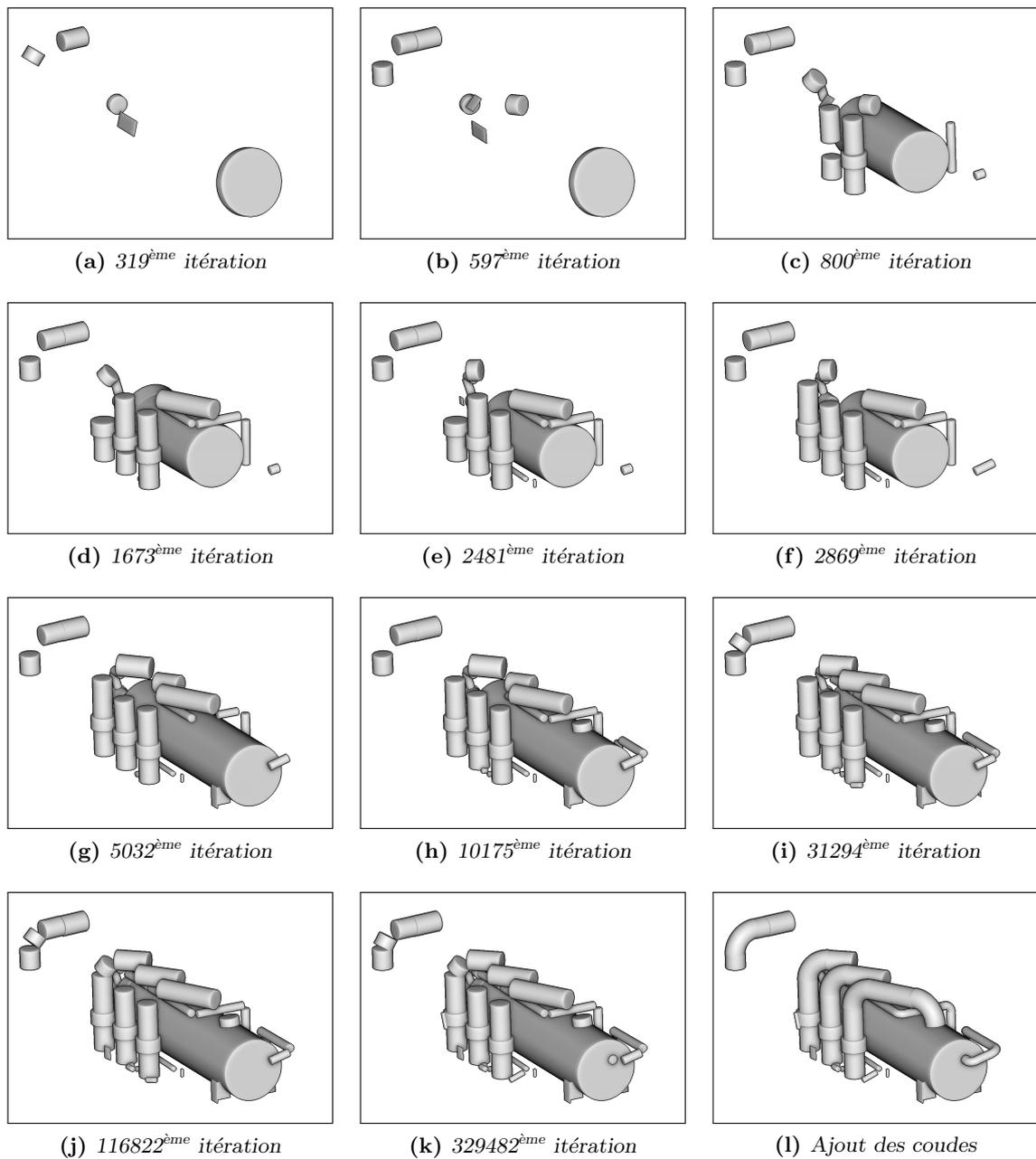
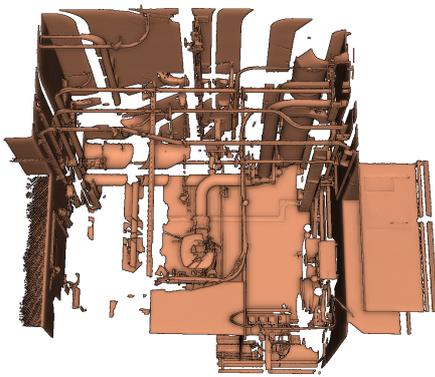


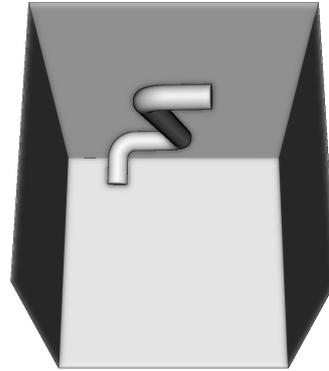
Figure 3.12: Évolution de la solution au cours du processus de reconstruction glouton, avec  $N = 20$ .

	Itération	$ \mathcal{X} $	$H$	$\lambda_D H_D$	$\lambda_G H_G$	$\lambda_T H_T$
Tests	208304	60	-189.953	-173.417	4.49489	-21.0307
	234600	56	-182.033	-167.193	4.0994	-18.9393
	335341	56	-184.484	-166.404	4.64991	-22.7302
	342688	59	-189.057	-171.96	5.02141	-22.1185
	422804	57	-186.985	-169.094	3.92609	-21.8166
	438787	58	-189.227	-173.723	4.95591	-20.4605
	370121	58	-188.387	-172.169	5.98994	-22.2071
	291365	60	-190.018	-169.555	4.80577	-25.2684
Variabilité	69.7%	6.9%	4.3%	4.3%	43.5%	29%

**Table 3.4:** Résultats de différentes exécutions avec le même paramétrage. La dernière ligne présente la variabilité des grandeurs étudiées ici en terme de proportion de la plage des valeurs par rapport à la valeur moyenne.



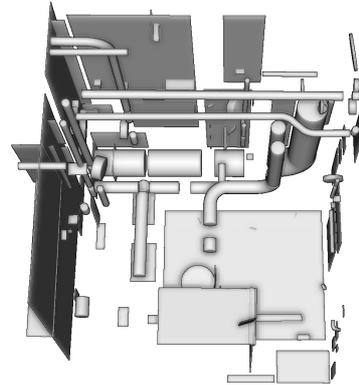
(a) Nuage de points (17170195 points)



(b) Modèle CAO a priori (4 cylindres, 4 plans et 3 tores)



(c) Segmentation (4378507 points)

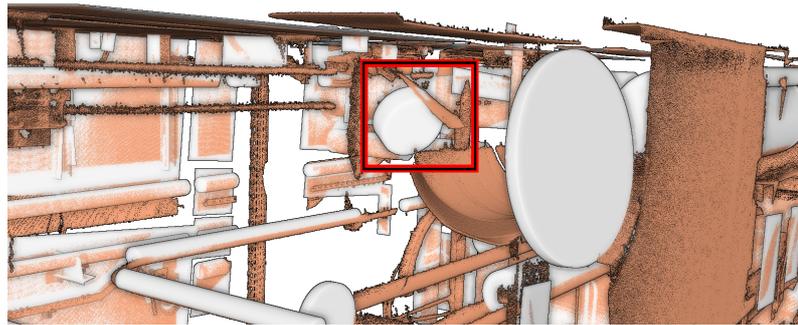


(d) Modèle CAO reconstruit (63 cylindres, 99 plans et 8 tores)

**Figure 3.13:** Reconstruction d'une salle de centrale à partir d'un modèle a priori partiel et de tolérance au changement permissives.

et permet ce genre d'erreur.

L'autre source de problème rencontrée sur cet exemple tient au calcul des limites de primitives, qui sera abordé par la suite (section 4.3). Le paramétrage utilisé ici, en l'occurrence la taille des cellules  $\zeta$ , ne permet pas de construire des primitives qui recou-



**Figure 3.14:** Erreur de reconstruction. Bien que le plupart des formes acceptées par la méthode soient pertinentes vis-à-vis du problème de reconstruction que l'on se pose, il existe parfois des primitives qui n'ont pas leur place dans la solution. Dans la situation présentée ici, le cylindre mis en évidence est par exemple accepté puisque l'énergie d'attache aux données n'est pas en mesure de pénaliser l'objet, et l'accepte de justesse.

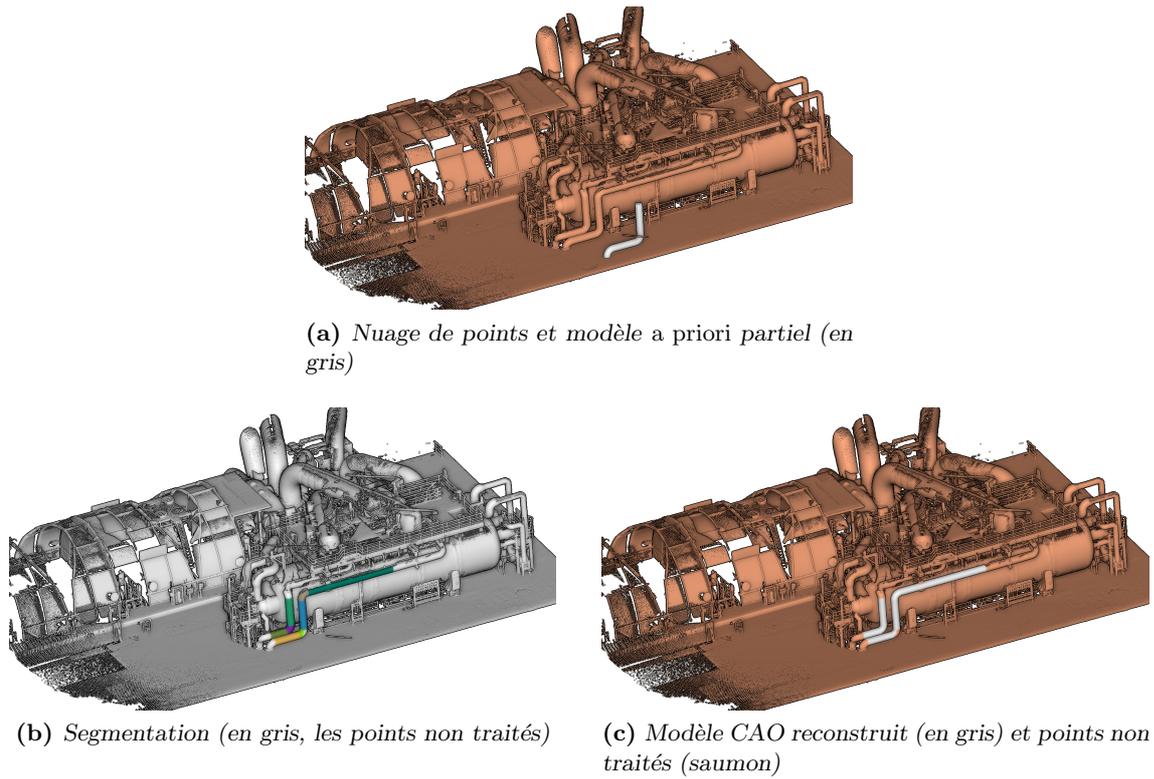
vrent les "trous" du nuage de points. Ainsi la plupart des tuyaux de la scène sont-ils ici fractionnés en cylindres coaxiaux répartis ponctuellement le long des lignes de tuyauterie. Ce résultat, s'il n'est pas esthétiquement idéal, n'est pas faux dans le sens où il correspond à une réalité des données. L'algorithme que nous proposons ne permet donc pas de combler les données absentes dans tous les cas.

La figure 3.15 quant à elle illustre de comportement de la méthode lorsque l'on cherche à reconnaître, au sein du nuage, les seules parties qui correspondent au modèle CAO *a priori* spécifié. Cette situation requiert de renseigner des tolérances aux changements relativement contraignantes, de manière à ne conserver que les primitives correspondant effectivement aux formes composant la requête. Le cas traité est relativement simple, mais la méthode parvient bien à trouver les quelques tuyauteries semblables au modèle CAO de requête sur cet exemple.

En utilisant un modèle CAO *a priori* plus complet sur cette même scène complexe, on obtient les résultats présentés en figure 3.16. L'algorithme parvient à traiter convenablement un certain nombre d'objets, en particulier les principales tyauteries (cf. figures 3.17 et 3.18). Il peine toutefois à reconstruire l'ensemble de la scène correspondant au modèle CAO *a priori* utilisé. Les rambardes visibles dans la figure 3.17 par exemple ne sont pas reconnues par la méthode. Ces lacunes sont vraisemblablement dues à des difficultés rencontrées par le générateur de primitives à l'égard de ces objets. Il s'agit en effet d'assemblages de cylindres étroits (un rayon de 2 cm, alors que le seuil de bruit utilisé ici est de  $\epsilon = 3$  cm) situés dans une zone du nuage de points relativement complexe du fait du nombre de ses composants et de la qualité de l'échantillonnage. L'absence quasi totale des barreaux de rambardes en cet endroit de la scène semble indiquer que le générateur ne parvient pas à proposer d'objets pertinents. De tels candidats seraient en effet acceptés immédiatement par l'algorithme de minimisation en l'absence de collisions avec d'autres formes.

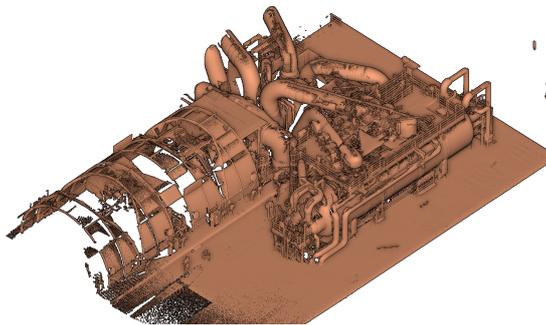
En certains endroits de cette même scène, l'algorithme parvient toutefois à identifier des rambardes plus isolées, et donc plus simples à traiter (cf. figure 3.19). Ces équipements sont en revanche majoritairement traités à l'aide de plans, permettant une représentation approximative du nuage de points à l'échelle traitée.

Bien qu'incomplet, le modèle CAO reconstruit par la méthode est malgré tout de bonne qualité, dans la mesure où les relations entre composants satisfont les règles d'assemblage

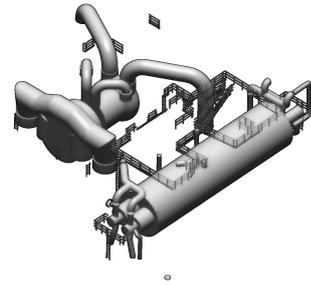


**Figure 3.15:** Reconnaissance d'un modèle CAO a priori partiel. En utilisant les tolérances aux changements appropriées ( $\sigma_p = 3m$  et  $\sigma_o = 10^\circ$ ), l'algorithme de reconstruction adopte bien le comportement d'une méthode de reconnaissance d'objets, puisque seules les parties correspondant effectivement à la requête figurent dans le résultat.

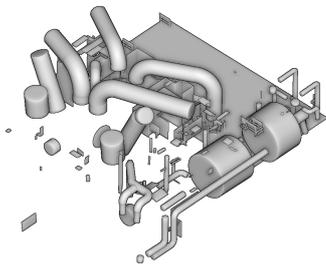
métier et où les équipements reconnus s'appuient effectivement sur le nuage de points. L'orthogonalité des plans permet ainsi typiquement de reconstruire des boîtiers bien formés (cf. figure 3.18).



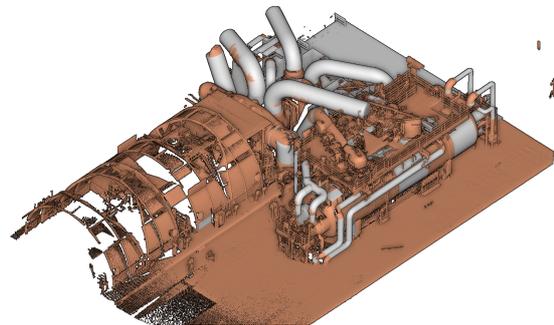
(a) Nuage de points (21916642 points)



(b) Modèle CAO a priori (1674 primitives dont 1207 plans, 416 cylindres et 51 tores)

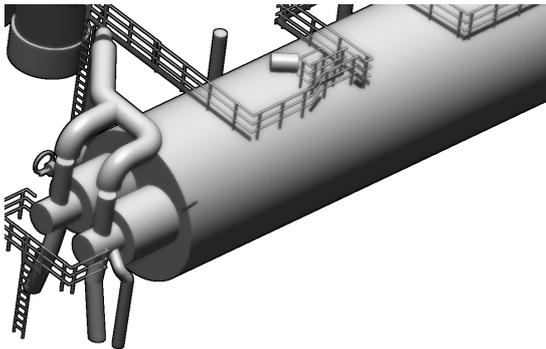


(c) Modèle CAO reconstruit (389 primitives dont 286 plans, 83 cylindres et 20 tores)

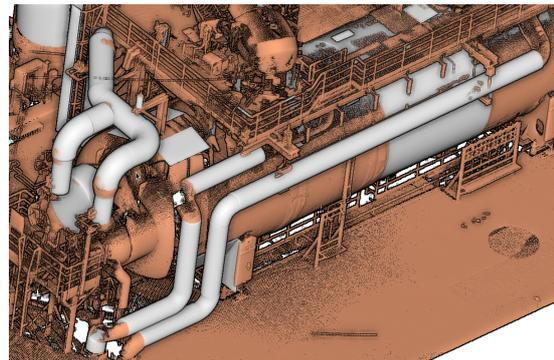


(d) Modèle CAO reconstruit (en gris) et points non traités (saumon)

**Figure 3.16:** Reconstruction d'une scène complexe. La méthode permet ici de retrouver des tuyaux initialement absents du modèle CAO a priori. Elle ne parvient toutefois pas à traiter les rambardes pourtant présentes dans la scène a priori.

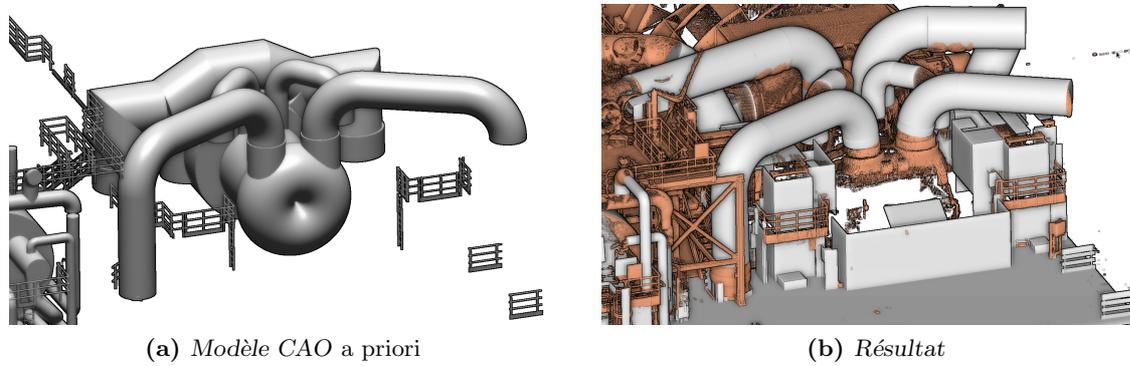


(a) Modèle CAO a priori

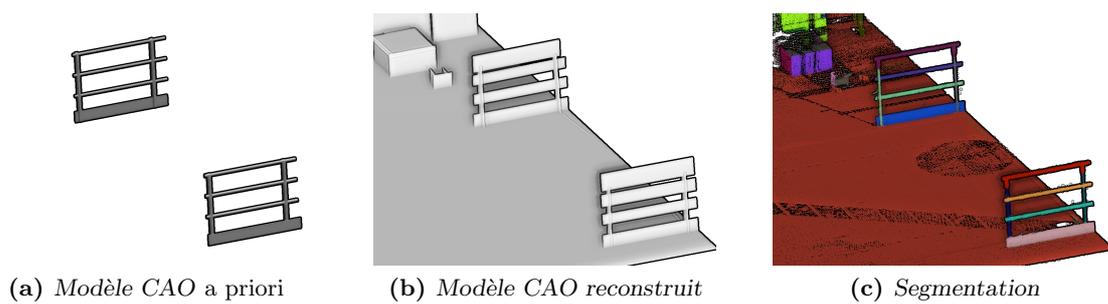


(b) Résultat

**Figure 3.17:** L'algorithme semble en mesure de traiter convenablement les quelques lignes de tuyauterie de la scène, qu'elles soient présentes dans le modèle CAO a priori ou non. Les rambardes sur cet exemple ne sont toutefois pas traitées, du fait de la petite taille des éléments qui les composent.



**Figure 3.18:** La méthode parvient ici à retrouver des éléments effectivement présents dans le nuage de points qui étaient pourtant absents du modèle CAO a priori  $\mathcal{M}_0$ . Mais la présence d'objets de référence semblables dans  $\mathcal{M}_0$  permet leur reconstruction. Le fait que la surface des plans ne soit pas prise en compte dans la mesure de dissimilarité permet notamment la détection de ces nouveaux composants massifs.



**Figure 3.19:** Gestion des rambardes. L'algorithme réussit ici à identifier des rambardes. Toutefois, leur reconstruction s'effectue à l'aide de plan au lieu des cylindres initialement représentés dans le modèle CAO a priori.

# Chapter 4

## Génération de primitives

### Sommaire

---

<b>4.1 Principe</b> . . . . .	<b>139</b>
4.1.1 Échantillons aléatoires . . . . .	139
4.1.2 Recours à l' <i>a priori</i> . . . . .	140
<b>4.2 Création stochastique de primitives</b> . . . . .	<b>141</b>
4.2.1 Localiser la zone de création . . . . .	141
4.2.2 Sélection des points restants . . . . .	143
4.2.3 Création à partir des points . . . . .	149
4.2.4 Analyse du processus et amélioration . . . . .	149
<b>4.3 Borner les primitives</b> . . . . .	<b>154</b>
4.3.1 Limites grossières . . . . .	155
4.3.2 Limites effectives . . . . .	155
4.3.3 Bornes du cylindre . . . . .	155
4.3.4 Bornes du plan . . . . .	157
4.3.5 Taille des tranches et cellules . . . . .	159
<b>4.4 Calcul de la couverture</b> . . . . .	<b>160</b>
<b>4.5 Résultats</b> . . . . .	<b>161</b>
4.5.1 Calcul des limites de cylindres . . . . .	161
4.5.2 Adéquation vis-à-vis des données et de l' <i>a priori</i> . . . . .	162

---

---

### Préambule

---

La méthode de recherche de la configuration optimale proposée au chapitre 3 suppose l'existence d'un "oracle" permettant la création aléatoire d'objets. Cet outil externe permet de piocher des éléments au sein de l'espace des primitives géométriques de manière à fournir des candidats qui seront assemblés, petit à petit, en la solution la plus probable au sens des attentes du problème.

Toutefois, cette génération ne peut se faire sans une certaine méthode, puisque la création parfaitement aléatoire de primitives n'a que très peu de chances d'aboutir à l'obtention d'un objet qui figure effectivement dans la solution recherchée. Pour permettre la convergence rapide vers la solution optimale, la méthode de génération de primitives doit ainsi favoriser l'apparition des formes qui composent effectivement cette solution, et donc construire préférentiellement les objets dont la probabilité a posteriori est élevée (section 4.1).

Pour être pertinente, une primitive doit à la fois épouser les surfaces décrites par le nuage de points, mais aussi être proche d'une des primitives du modèle CAO a priori. Nous proposons donc d'utiliser ces deux sources de données pour biaiser le processus de création en faveur des objets les plus aptes à figurer dans la solution recherchée. Les primitives générées sont ainsi construites à partir de points sélectionnés dans le nuage, cette sélection étant guidée par la mise à disposition d'une forme issue du modèle CAO a priori (section 4.2).

Une fois un candidat créé, il est ensuite nécessaire de délimiter son extension dans l'espace (section 4.3). En effet, par défaut, les candidats générés sont supposés infinis: le cylindre s'étend sur une longueur non limitée, de même que le plan n'est pas délimité en longueur ni en largeur. Le nuage de points peut alors être utilisé pour identifier la section sur laquelle ces objets sont effectivement définis. Lors de ce processus, on peut de plus identifier la proportion de la surface des objets recouverte par les points du nuage, de manière à exclure tout candidat qui ne satisferait pas une adéquation minimale aux données observées et ainsi éviter des calculs inutiles au processus d'assemblage (section 4.4).

La méthode proposée pour le processus de génération permet de constater, en pratique, un biais plus ou moins prononcé en faveur des primitives les plus pertinentes (section 4.5), selon la situation dans laquelle se trouve la primitive de référence issue du modèle a priori vis-à-vis des formes qui lui correspondent dans le nuage de points.

---

## 4.1 Principe

La génération aléatoire de primitives géométriques pour la construction incrémentale du modèle CAO permet en théorie l'exploration de toute configuration de  $\Omega$ . La gestion des primitives générées est déléguée à un processus tiers (BDMCMC, jump-dynamic ou algorithme glouton mentionnés au chapitre 3) chargé d'assembler les formes en un modèle CAO qui tend à s'améliorer au fil des primitives proposées. En apparence, les processus de proposition et d'acceptation des solutions semblent donc parfaitement indépendants.

Or, il existe une infinité de formes pouvant être générées. La probabilité d'obtenir par tirage purement aléatoire une primitive appartenant à la configuration optimale est donc quasi nulle, et l'obtention d'un modèle CAO pertinent requiert alors en théorie une quantité infinie de tentatives. Pour permettre la reconstruction, il est donc indispensable de tenir compte, dans la génération des primitives, de la probabilité que l'on cherche à minimiser. Il s'agit ainsi de biaiser la génération aléatoire de manière à favoriser l'apparition de formes aptes à optimiser la probabilité cible, et ainsi permettre la reconstruction en une quantité raisonnable de tentatives.

L'introduction de biais dans un processus stochastique est une idée apparaissant dans certains travaux existants. R. Schnabel et al. [SWK07] proposent par exemple une approche permettant d'augmenter considérablement l'efficacité de l'algorithme RANSAC, en modifiant le choix des points servant à la génération des primitives de manière à encourager la sélection de points échantillonnant une primitive commune. X. Descombes et al. [DMZ08] quant à eux proposent de générer chaque primitive selon une probabilité dépendant de sa pertinence vis-à-vis des données, augmentant ainsi la fréquence d'apparition des formes conformes aux données.

### 4.1.1 Échantillons aléatoires

Il ressort des travaux existants que la génération des primitives doit *a minima* favoriser la création de formes qui correspondent effectivement aux données. Pour ce faire, la procédure de proposition de candidats de l'algorithme RANSAC (cf. section 1.3.3 - algorithme 1) semble pertinente. Il s'agit de choisir aléatoirement une quantité restreinte de points du nuage suffisante pour déterminer de manière non ambiguë la primitive qui décrit ces données. Cette approche possède deux avantages majeurs:

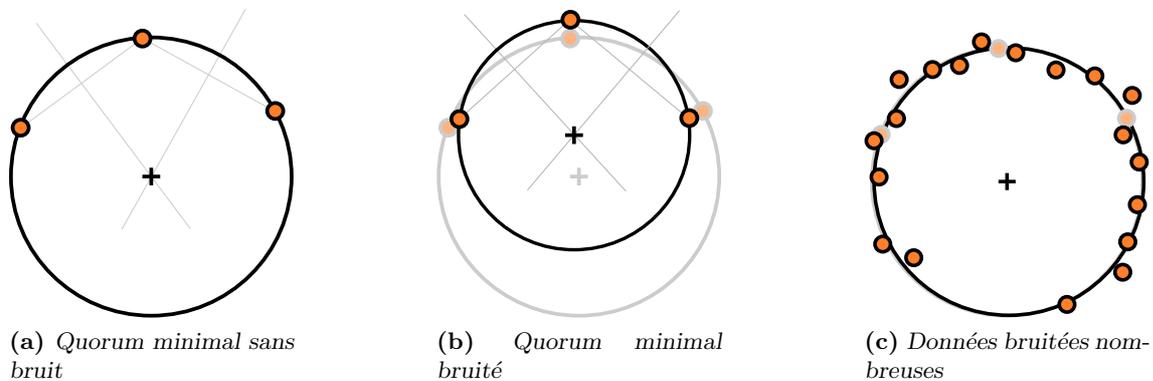
- puisque les primitives sont créées à partir de points du nuage, les candidats générés garantissent une attache aux données minimale.
- la quantité de primitives pouvant être générées est finie et dénombrable du fait de la nature combinatoire de la sélection des points. Ceci permet de discrétiser naturellement l'espace des solutions, en le restreignant à l'ensemble des primitives s'appuyant sur des points du nuage.

Ce simple processus rend nettement plus improbable l'apparition de primitives non pertinentes, et le calcul du modèle CAO reconstruit par génération stochastique devient ainsi envisageable.

Un des principaux défauts des approches de RANSAC existantes tient au fait qu'elles se basent sur le tirage aléatoire de quorums de points minimaux. Les formes sont en effet générées à partir de la plus petite quantité de points permettant de contraindre de manière unique les paramètres libres de la primitive à instancier. Les normales associées

aux points sont d'ailleurs parfois utilisées pour permettre de diminuer le nombre de points à tirer, puisqu'elles apportent des contraintes supplémentaires permettant d'instancier les primitives. En effet, plus on sélectionne de points dans le nuage, moins il est probable que le quorum défini décrive une surface commune (cf. section 1.3.3), et il est donc souvent nécessaire de restreindre au maximum la quantité de points requis, à des fins d'efficacité.

Or, il est évident que l'utilisation d'une faible quantité de points induit une fragilité excessive du processus de génération de primitives vis-à-vis du bruit du nuage de points. En effet, la moindre erreur sur les données du quorum impacte considérablement la primitive qu'il engendre, diminuant ainsi la probabilité de voir apparaître une forme qui s'ajuste effectivement sur les données. A l'inverse, le fait de créer des formes approximant une grande quantité de points tous choisis sur une même surface induirait, par effet de "moyennage", une robustesse du processus de génération vis-à-vis du bruit du nuage. Il faudrait toutefois prendre garde à ce qu'aucun *outlier*<sup>1</sup> ne vienne se glisser parmi les points sélectionnés, puisque cela perturberait considérablement la primitive engendrée par le sous-ensemble de points.



**Figure 4.1:** Effet du bruit sur la construction de primitives géométriques. Sur cet exemple, on voit que l'incertitude sur la position des points peut avoir un effet significatif sur l'erreur des primitives estimées à partir d'un faible nombre de points, alors que le fait de considérer un grand nombre de données permet de garantir la robustesse vis-à-vis du bruit.

#### 4.1.2 Recours à l'*a priori*

Outre l'utilisation des données observées, les données *a priori* peuvent elles aussi permettre d'augmenter significativement la pertinence générale des formes candidates proposées. Tout comme le suggèrent R. Schnabel et al. [SWK07], il semble judicieux d'intégrer au processus de sélection des points certains procédés visant à augmenter la probabilité de générer une forme qui soit effectivement dans le nuage. Pour ce faire, nous pouvons nous appuyer sur le modèle CAO *a priori* supposé, par hypothèse, être similaire au nuage de points à quelques modifications près. Cet *a priori* peut ainsi être utilisé pour guider le choix des points du nuage permettant la création de primitives.

Le gain de fiabilité introduit par l'utilisation de l'*a priori* pour guider la sélection des points permet d'en choisir un grand nombre pour la génération des primitives. Comme nous le mentionnions précédemment, l'augmentation de la quantité de points tirés permet de renforcer la pertinence des primitives créées vis-à-vis des données, en atténuant l'effet du bruit du nuage de points sur le processus de génération.

<sup>1</sup>Rappelons qu'un *outlier* est un point ne correspondant pas au processus d'échantillonnage de la surface d'intérêt.

D'autre part, le recours à l'*a priori* permet de générer préférentiellement des candidats proches des formes du modèle CAO de référence  $\mathcal{M}_0$ , satisfaisant ainsi les attentes vis-à-vis du problème de reconstruction tel que nous l'avons posé.

Finalement, on peut donc poser le problème de la manière suivante: *étant donnée une primitive  $\mathcal{S}^i$  issue du modèle CAO a priori  $\mathcal{M}_0$ , sélectionner un sous-ensemble de points du nuage  $\mathcal{P}$  qui décrit une primitive semblable à  $\mathcal{M}_0$ , aux transformations tolérées près.* Le processus traitant de ce problème définit un noyau de proposition  $q(\mathcal{S}^i, \mathcal{P})$  pertinent pour la recherche de la configuration optimale (cf. algorithme 5). Le recours aux points du nuage permet de générer des formes aptes à minimiser l'énergie d'attache aux données  $H_D$ , tandis que la contrainte de similarité vis-à-vis des primitives *a priori* permet de favoriser l'apparition de primitives minimisant l'énergie géométrique  $H_G$ .

## 4.2 Création stochastique de primitives

L'algorithme de reconstruction 5 que nous avons proposé en section 3.3.2 indique, lors de chaque étape, une primitive  $\mathcal{S}^i$  issue du modèle CAO *a priori* sur laquelle le noyau de proposition  $q$  peut s'appuyer pour guider la génération d'un candidat.

Rappelons que les scènes représentées par le modèle CAO *a priori*  $\mathcal{M}_0$  et le nuage de points  $\mathcal{P}$  sont semblables par hypothèse, dans le sens où les éléments qui se correspondent entre les deux scènes sont similaires à certaines transformations près. En particulier, la probabilité de voir survenir une modification sur un élément diminue lorsque l'amplitude de cette modification augmente, conformément à ce qu'exprime l'énergie géométrique  $H_G$  (cf. section 2.3.2). On distingue notamment les trois composantes définissant une modification (déplacement, désaxage et changement géométrique), chacune étant associée à une tolérance quant à l'amplitude des changements pouvant survenir.

L'algorithme que nous proposons pour permettre la génération d'une primitive candidate pertinente vis-à-vis de  $\mathcal{S}^i$  et du nuage de points  $\mathcal{P}$  procède en trois temps. Il s'agit tout d'abord de localiser la partie du nuage de points à partir de laquelle la primitive candidate  $\mathcal{S}$  va être créée. Pour ce faire, on sélectionne un premier point du nuage en cherchant à favoriser le choix de points échantillonnant une surface pertinente vis-à-vis de  $\mathcal{S}$ . Dans un second temps, on isole dans le voisinage de ce premier point un sous-ensemble paraissant décrire une même surface. La création de la primitive est ensuite réalisée par le calcul de la primitive approximant au mieux ce sous-ensemble.

### 4.2.1 Localiser la zone de création

Pour augmenter les chances de générer une primitive  $\mathcal{S}$  pertinente, on doit privilégier dans la mesure du possible l'exploration des parties du nuage de points qui:

- sont proches de la primitive de référence  $\mathcal{S}^i$
- présentent une orientation conforme à l'orientation de  $\mathcal{S}^i$
- échantillonnent une surface géométriquement équivalente à  $\mathcal{S}^i$

Tous les points du nuage ne se valent donc pas en matière de pertinence vis-à-vis de l'*a priori*  $\mathcal{S}^i$ . Toutefois, aucune partie du nuage ne peut être exclue pour la génération de la primitive candidate, puisque tout candidat généré à partir du nuage de points est potentiellement une "version transformée" de la primitive  $\mathcal{S}^i$ . En effet, la probabilité d'une

modification liant un candidat  $\mathcal{S}$  à la référence  $\mathcal{S}^i$  est strictement positive (donc non nulle) quelle que soit l'amplitude de cette modification. Bien que l'on cherche à privilégier la création de formes semblables à  $\mathcal{S}^i$ , on ne peut donc interdire l'apparition d'aucune forme.

### Probabilité sur les points du nuage

On définit ainsi en tout point  $\mathbf{p}$  du nuage une probabilité  $P_S(\mathbf{p}|\mathcal{S}^i)$  quantifiant la pertinence de  $\mathbf{p}$  vis-à-vis de la primitive de référence  $\mathcal{S}^i$ .

$$P_S(\mathbf{p}|\mathcal{S}^i) = e^{-\left(\frac{d(\mathcal{S}^i, \mathbf{p})}{\sigma_p}\right)^2 - \left(\frac{o(\mathcal{S}^i, \mathbf{p})}{\sigma_o}\right)^2} \quad (4.1)$$

Rappelons que  $d(\mathcal{S}^i, \mathbf{p})$  désigne la distance séparant le point  $\mathbf{p}$  de  $\mathcal{S}^i$ . La fonction  $o(\mathcal{S}^i, \mathbf{p})$  correspond au défaut d'alignement de la normale  $\mathbf{n}_{\mathcal{P}}(\mathbf{p})$  associée à  $\mathbf{p}$  vis-à-vis de l'orientation de  $\mathcal{S}$ . Elle s'exprime comme un angle et son expression dépend du type de la primitive  $\mathcal{S}^i$ .

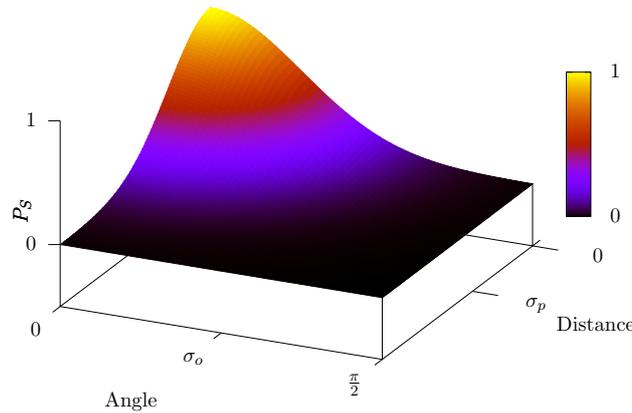
Si  $\mathcal{S}^i$  est un cylindre dont l'axe est dirigé selon le vecteur unitaire  $\mathbf{a}_{\mathcal{S}^i}$ :

$$o(\mathcal{S}^i, \mathbf{p}) = \left| \arccos(|\mathbf{n}_{\mathcal{P}}(\mathbf{p}) \cdot \mathbf{a}_{\mathcal{S}^i}|) - \frac{\pi}{2} \right|$$

Si  $\mathcal{S}^i$  est un plan de vecteur normal  $\mathbf{n}_{\mathcal{S}^i}$ :

$$o(\mathcal{S}^i, \mathbf{p}) = \arccos(|\mathbf{n}_{\mathcal{P}}(\mathbf{p}) \cdot \mathbf{n}_{\mathcal{S}^i}|)$$

Les paramètres  $\sigma_p$  et  $\sigma_o$  quant à eux définissent respectivement la tolérance au déplacement et au désaxage des primitives géométriques reconstruites vis-à-vis de leur correspondances dans le modèle CAO *a priori*  $\mathcal{M}_0$ . Ils apparaissent d'ores et déjà dans l'expression de l'énergie géométrique définie en équation 2.24.



**Figure 4.2:** Probabilité  $P_S(\mathcal{S}^i, \mathbf{p})$  du point  $\mathbf{p}$  vis-à-vis de la primitive de référence  $\mathcal{S}^i \in \mathcal{M}_0$  en fonction de la distance  $d(\mathcal{S}, \mathbf{p})$  et de l'angle  $o(\mathcal{S}^i, \mathbf{p})$ .

La probabilité associée à un point du nuage augmente donc lorsque ce point se rapproche de la primitive  $\mathcal{S}^i$ , et lorsque sa normale se conforme à l'orientation de  $\mathcal{S}$ . L'évolution de cette probabilité est de plus cohérente avec l'évolution de l'énergie  $H_G$ , puisqu'elles reposent toutes deux sur les mêmes tolérances et sur l'étude de grandeurs similaires (défauts de position et d'orientation).

Bien que nous utilisions les critères liés à la position et l'orientation de la primitive, nous ne tenons pas compte de sa géométrie. Ceci tient au fait que l'évaluation fiable d'un descripteur géométrique dans le nuage de points repose sur l'étude locale d'un échantillon suffisamment vaste pour apporter une information convenable au sujet d'une unique surface. Il s'agit donc de proposer une méthode s'adaptant à une échelle parfaite pour permettre le calcul d'un descripteur pertinent. Or, cette tâche est très complexe. On aurait par exemple pu s'appuyer sur l'évaluation des courbures dans le nuage de points, mais les quelques méthodes testées sur nos données se sont révélées peu satisfaisantes, du fait du bruit et des défauts de densité du nuage. Nous préférons donc nous passer de cette information plutôt que d'utiliser une donnée erronée.

### Tirage d'un premier point

Avant d'envisager de tirer tous les points destinés à permettre la création d'une primitive, nous proposons dans un premier temps d'identifier une "graine"  $\mathbf{p}_{i_0} \in \mathcal{P}$  pertinente vis-à-vis de la référence  $\mathcal{S}^i$ . À partir de cette graine, on pourra par la suite sélectionner les points restants parmi ceux échantillonnant la même surface que  $\mathbf{p}_{i_0}$ .

La sélection de  $\mathbf{p}_{i_0}$  est effectuée en simulant un tirage dans  $\mathcal{P}$  dont la densité de probabilité est proportionnelle à  $P_S$ . Ce tirage peut être effectué [CG95] au moyen d'une simple méthode d'acceptation-rejet (algorithme 7), ou de l'algorithme de Metropolis-Hastings (cf. algorithme 3 section 3.2.1). Chacun de ces algorithmes peut tester une quantité importante de points avant de fournir une graine. Tout dépend notamment du noyau de proposition utilisé pour l'exploration des données. Nous proposons ici d'utiliser un noyau uniforme sur le nuage de points.

---

**Algorithme 7** : Tirage par acceptation-rejet basé sur l'exploration uniforme des données.

---

**Données** :  $\mathbb{Q}$  : ensemble de données  
 $P$  : probabilité définie sur les éléments de  $\mathbb{Q}$ , dont la densité n'est connue qu'à un facteur près

**Résultat** :  $q \in \mathbb{Q}$  choisi avec la probabilité  $P$

```

1 Tant que VRAI faire                                     /* Boucle infinie */
2   Choisir un élément  $q \in \mathbb{Q}$  selon une loi uniforme;
3   Générer un réel  $u \in ]0, 1]$  selon une loi uniforme;
4   Si  $u \leq P(q)$  alors
5     | Retourner  $q$ 

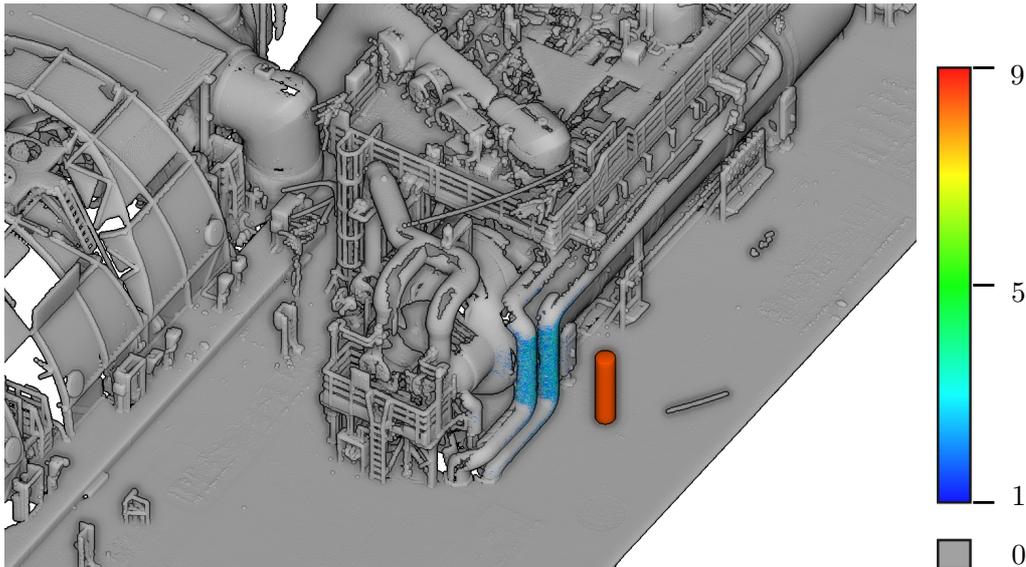
```

---

#### 4.2.2 Sélection des points restants

Une fois la graine  $\mathbf{p}_{i_0}$  sélectionnée, il faut ensuite choisir un certain nombre de points représentant la même surface que celle dont  $\mathbf{p}_{i_0}$  est issue, sous l'hypothèse que  $\mathbf{p}_{i_0}$  représente effectivement un objet du même type que  $\mathcal{S}^i$  (cylindre ou plan). Pour assurer la pertinence de la primitive candidate créée, il est important qu'aucun *outlier* ne figure parmi les points retenus, puisqu'un tel élément induirait une erreur significative sur l'estimation des paramètres du candidat.

Comme le remarquent R. Schnabel et al. [SWK07], plus les points sont choisis proches les uns des autres, plus la probabilité pour qu'ils décrivent une même surface est élevée. À ce titre, il est judicieux de choisir les points dans un même voisinage localisé autour



**Figure 4.3:** Résultat du tirage de graines dans un nuage de points. 100000 tirages ont été réalisés dans un nuage de 22 millions de points, et les couleurs représentées ici correspondent au nombre de sélections de chacun des points (en gris, les points qui n'ont pas été tirés). Le cylindre rouge correspond à la référence  $\mathcal{S}^i$  utilisée ici.

de la graine  $\mathbf{p}_{i_0}$ . Toutefois, pour porter une information géométrique suffisante, les points doivent être suffisamment répartis sur la surface de l'objet à deviner. La taille du voisinage doit donc être suffisante pour permettre de capturer des points qui contraignent convenablement la géométrie du candidat que l'on va générer. Idéalement, il est par exemple souhaitable de recueillir des points de part et d'autre d'un même cylindre pour définir de manière fiable son rayon.

Ces deux contraintes antagonistes sont telles qu'il est quasiment impossible d'assurer l'obtention d'un voisinage ne contenant que des points sur une même surface et qui porte une information géométrique suffisante dans le même temps. On préférera donc surévaluer légèrement la taille du voisinage dans un premier temps. Les *outliers* contenus dans ce voisinage pourront par la suite être évincés en appliquant une méthode d'analyse de la surface qu'il représente.

### Voisinage

On souhaite identifier une zone de l'espace qui englobe un grand nombre de points échantillonnant la même surface que  $\mathbf{p}_{i_0}$ . Naturellement, pour que ce voisinage soit aussi opportun que possible, il est nécessaire de tenir compte de la nature de la surface que l'on cherche à reconstruire : le processus de construction d'un voisinage englobant un plan est différent de celui englobant un cylindre.

En tous les cas, il nous faut estimer *a priori* l'étendue de ce voisinage dans l'espace. Pour ce faire, il semble judicieux d'utiliser la taille de  $\mathcal{S}^i$  puisque l'on cherche à construire une forme qui lui ressemble ( $\mathbf{p}_{i_0}$  a d'ailleurs été choisi dans cette optique).

**Cas du cylindre** Si  $\mathcal{S}^i$  est un cylindre, on recherche donc un cylindre dans le nuage de points. En ce cas, la taille du voisinage peut être estimée à partir du rayon de  $\mathcal{S}^i$ .

Plusieurs approches permettent de traiter le problème. Tout d'abord, concernant la nature du voisinage, plusieurs choix s'offrent à nous. Le voisinage le plus naturel est le

voisinage sphérique, c'est-à-dire l'ensemble des points dont la distance à un centre  $\mathbf{c}$  est inférieure à un rayon  $r$  :

$$\mathcal{N}_{S[r]}(\mathbf{c}, \mathcal{P}) = \{\mathbf{p} \in \mathcal{P} \mid \|\mathbf{p} - \mathbf{c}\| \leq r\} \quad (4.2)$$

Une alternative consiste en l'utilisation de voisinages cylindriques, c'est-à-dire l'ensemble des points dont la distance radiale à un axe engendré par un centre  $\mathbf{c}$  et un vecteur directeur unitaire  $\mathbf{a}$  est inférieure à un rayon  $r$ , et dont la distance à  $\mathbf{c}$  le long de cet axe est inférieure à une demi hauteur  $h$ :

$$\mathcal{N}_{C[\mathbf{a}, r, h]}(\mathbf{c}, \mathcal{P}) = \{\mathbf{p} \in \mathcal{P} \mid \|(\mathbf{p} - \mathbf{c}) \times \mathbf{a}\| \leq r \text{ et } |(\mathbf{p} - \mathbf{c}) \cdot \mathbf{a}| \leq h\} \quad (4.3)$$

Quel que soit le voisinage choisi, il est nécessaire de déterminer les paramètres qui le définissent, comme le centre  $\mathbf{c}$  et le rayon  $r$  (plus le vecteur directeur  $\mathbf{a}$  et la hauteur  $h$  pour le voisinage cylindrique). Dans un premier temps, pour s'assurer que le voisinage englobe au mieux la surface à laquelle appartient la graine  $\mathbf{p}_{i_0}$ , nous proposons de décaler virtuellement  $\mathbf{p}_{i_0}$  le long de sa normale  $\mathbf{n}_{\mathcal{P}}(\mathbf{p}_{i_0})$  afin de positionner le point à l'intérieur de l'éventuel cylindre qu'il échantillonne. Ce faisant, on renforce la pertinence du voisinage considéré puisque l'on diminue les chances d'englober des *outliers*, et il devient possible d'intégrer dans le processus de construction du candidat des points diamétralement opposés à  $\mathbf{p}_{i_0}$  sur la surface cylindrique considérée. On augmente ainsi la quantité et la pertinence de l'information géométrique portée par les données retenues. Pour effectuer ce décalage, on utilise le rayon  $r_{\mathcal{S}^i}$  du cylindre de référence  $\mathcal{S}^i$  :

$$\mathbf{c} = \mathbf{p}_{i_0} - (D \cdot r_{\mathcal{S}^i} \cdot \mathbf{n}_{\mathcal{P}}(\mathbf{p}_{i_0})) \quad (4.4)$$

$D$  est une variable déterminée en fonction des normales du nuage de points  $\mathcal{P}$  :

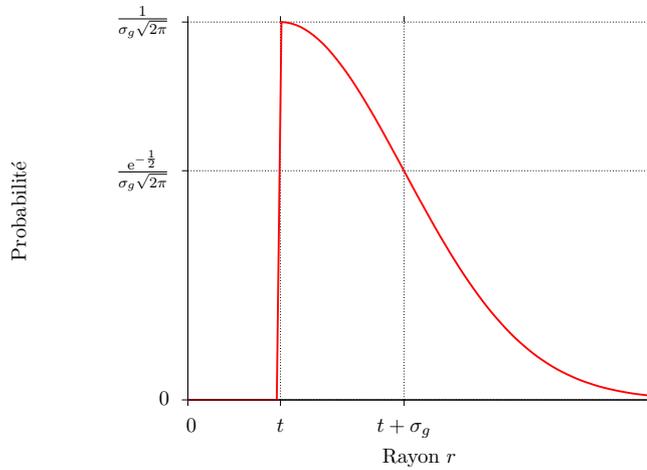
- si les normales ne sont pas orientées convenablement (et sont alors définies uniquement par leur direction, leur sens étant inconnu), deux sens de translation sont envisageables. On choisit alors de définir  $D$  comme étant une variable aléatoire prise équiprobablement dans l'ensemble  $\{-1, 1\}$ . En effet, s'il existait un algorithme capable de déterminer de manière fiable l'orientation de chaque normale, on suppose qu'il aurait été appliqué au nuage de points en guise de précalcul. En l'absence de cette information, on peut raisonnablement estimer que les deux orientations possibles pour chaque normale sont équitablement pertinentes, et on ne peut alors trancher. Chaque tentative de création d'un candidat présente ainsi une chance sur deux d'échec.
- si les normales sont correctement orientées (toutes "pointent vers l'extérieur" des objets échantillonnés), on posera alors simplement  $D = 1$ .

Une fois le centre  $\mathbf{c}$  défini, il s'agit de déterminer le rayon du voisinage. Il nous faut proposer une valeur plus grande que  $r_{\mathcal{S}^i}$ , de manière à englober *a minima* la portion de surface échantillonnée par la graine  $\mathbf{p}_{i_0}$ . De plus, comme nous le disions précédemment, il est préférable de surévaluer dans un premier temps la taille du voisinage. Il n'est pas non plus à exclure que  $\mathbf{p}_{i_0}$  ait été prélevé sur un objet cylindrique dont le rayon est bien supérieur à celui de  $\mathcal{S}^i$ , dans des proportions qu'on ne peut déterminer avant d'avoir reconstruit la surface. Or, le paramètre  $\sigma_g$  (cf. section 2.3.2 - équation 2.24) renseigne sur les changements de géométrie attendus vis-à-vis de l'*a priori*, et donc vis-à-vis du rayon du cylindre  $\mathcal{S}^i$ . On peut ainsi évaluer en termes probabilistes le facteur liant le rayon de

$\mathcal{S}^i$  au rayon de la surface échantillonnée en  $\mathbf{p}_{i_0}$ . Ainsi, on calcule le rayon  $r$  du voisinage comme suit:

$$r = |G| + r_{\mathcal{S}^i} + 2r_{\mathcal{S}} \sin\left(\frac{\theta(\mathbf{p}_{i_0})}{2}\right) + \eta\epsilon \quad (4.5)$$

Rappelons que  $\eta$  et  $\epsilon$  définissent le bruit dans le nuage de points (cf. section 2.3.1 - équation 2.18). L'angle  $\theta(\mathbf{p}_{i_0})$  désigne l'incertitude portant sur l'évaluation de la normale  $\mathbf{n}_{\mathcal{P}}(\mathbf{p}_{i_0})$  en  $\mathbf{p}_{i_0}$ . Les trois termes de droite permettent d'assurer que le voisinage englobera la section cylindrique portant  $\mathbf{p}_{i_0}$  si le rayon de celle-ci est inférieur ou égal à  $r_{\mathcal{S}^i}$ , malgré les incertitudes sur la position des points et sur les normales évaluées. La variable  $G$  est quant à elle une variable aléatoire réelle suivant la loi normale centrée d'écart type  $\sigma_g$  (cf. figure 4.4 - la loi normale peut être simulée de manière efficace en utilisant par exemple le générateur de Box-Müller), permettant ainsi d'anticiper l'éventuel grossissement des cylindres du nuage correspondant à  $\mathcal{S}^i$ .



**Figure 4.4:** Densité de probabilité pour le choix du rayon de voisinage. Ici,  $t$  désigne la valeur minimale pouvant être prise par le rayon  $r$ , à savoir  $t = \eta\epsilon + r_{\mathcal{S}^i} \left(1 + 2 \sin\left(\frac{\theta(\mathbf{p}_{i_0})}{2}\right)\right)$ .

Si l'on choisit de recourir aux voisinages cylindriques, il faut de plus déterminer la direction de l'axe  $\mathbf{a}$  ainsi que la hauteur  $h$  du voisinage. Le vecteur directeur  $\mathbf{a}_{\mathcal{S}^i}$  de l'axe du cylindre  $\mathcal{S}^i$  peut être retenu pour définir l'axe du voisinage. Quant à la hauteur, elle ne peut réellement être évaluée *a priori* puisque la longueur des cylindres dans la scène est essentiellement déterminée par les connexions entre éléments, qui peuvent varier grandement avec la position de ceux-ci. De sorte qu'il est certainement plus pertinent de déterminer la hauteur de manière à fixer le ratio d'aspect du cylindre. On pourra typiquement prendre arbitrairement une hauteur  $h = 4r$  (ou tout autre facteur semblant pertinent sur la base de constatations empiriques).

En pratique, nous avons constaté sur les exemples testés que les voisinages sphériques permettent généralement d'obtenir de meilleurs résultats que les voisinages cylindriques. Sauf mention contraire, on considérera donc par la suite que les voisinages sphériques sont utilisés.

**Cas du plan** Considérons maintenant le cas où  $\mathcal{S}^i$  est un plan. Les points échantillonnant la même surface que  $\mathbf{p}_{i_0}$  se trouvent dans un plan orthogonal à la normale  $\mathbf{n}_{\mathcal{P}}(\mathbf{p}_{i_0})$

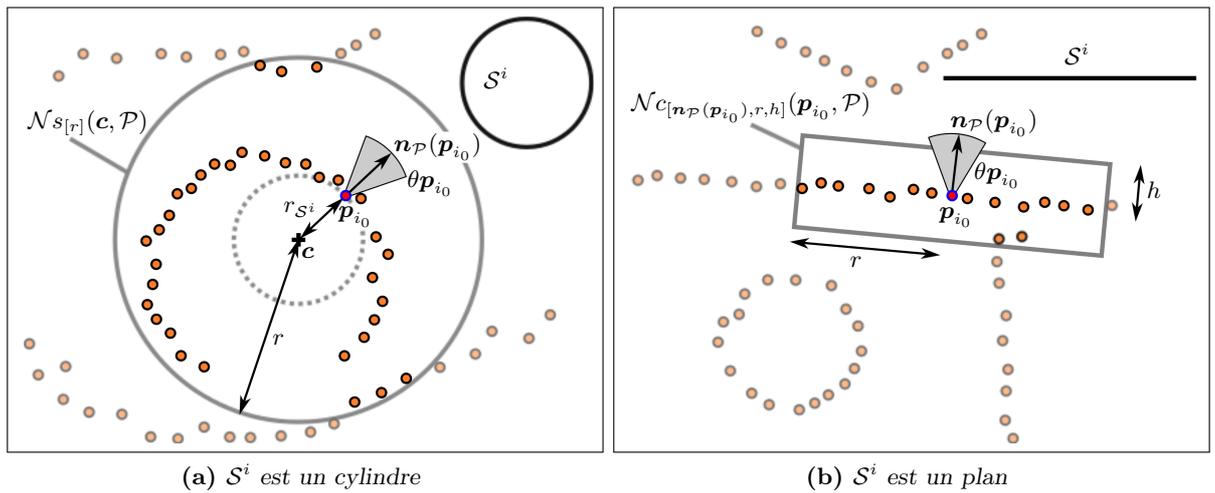
associée à  $\mathbf{p}_{i_0}$ , à l'erreur d'estimation de  $\mathbf{n}_{\mathcal{P}}(\mathbf{p}_{i_0})$  près. On se propose donc de retenir, dans cette situation, le voisinage cylindrique  $\mathcal{N}_{c_{[a,r,h]}(\mathbf{c}, \mathcal{P})}$  (équation 4.3) centré en  $\mathbf{c} = \mathbf{p}_{i_0}$  et engendré par le vecteur directeur  $\mathbf{a} = \mathbf{n}_{\mathcal{P}}(\mathbf{p}_{i_0})$ . Il reste alors à déterminer le rayon  $r$  et la hauteur  $h$  de ce voisinage.

Le rayon du voisinage doit permettre de représenter l'étendue spatiale du plan dans la scène, sa surface en quelque sorte. Or, cette surface ne peut pas vraiment être évaluée *a priori*, puisque les limites d'un plan se définissent essentiellement à partir de son interaction avec les autres primitives. On peut toutefois utiliser la surface  $w_{\mathcal{S}^i} \times l_{\mathcal{S}^i}$  de  $\mathcal{S}^i$  pour proposer une estimation très grossière du rayon  $r$ , faute de mieux. En faisant en sorte que l'aire de la section circulaire du voisinage cylindrique soit identique à l'aire du plan  $\mathcal{S}^i$ , on obtient:

$$r = \frac{\sqrt{w_{\mathcal{S}^i} \times l_{\mathcal{S}^i}}}{\pi} \quad (4.6)$$

Quant à la hauteur, elle doit tenir compte de la somme du bruit dans le nuage de points et de l'erreur d'estimation angulaire  $\theta(\mathbf{p}_{i_0})$  de  $\mathbf{n}_{\mathcal{P}}(\mathbf{p}_{i_0})$ .

$$h = \eta\epsilon + r \tan(\theta(\mathbf{p}_{i_0})) \quad (4.7)$$



**Figure 4.5:** Voisinage de la graine  $\mathbf{p}_{i_0}$  pour la construction de primitives candidates. Quel que soit le cas considéré, la méthode tient compte des incertitudes sur la position des points ainsi que sur l'estimation des normales aux points (cône d'incertitude) pour l'estimation de la taille des voisinages. Ces voisinages optimisent la quantité de points pertinents, bien qu'il soit possible qu'ils contiennent tout de même de faibles quantités d'outliers.

### Première estimation

Les voisinages proposés permettent de retenir une partie du nuage de points sensée représenter la surface du candidat que l'on se propose de construire. Il n'est toutefois pas exclu qu'il existe, parmi les points voisins retenus, une certaine quantité de points n'échantillonnant pas la surface ciblée. Il nous faut donc utiliser une méthode de reconnaissance de primitive qui soit robuste à ces éventuels *outliers*. L'algorithme du RANSAC proposé par R. Schnabel et al. [SWK07] semble être un outil approprié pour résoudre ce problème (cf. section 1.3.3). On peut en effet raisonnablement supposer que la surface la plus représentée par les points du voisinage est celle à laquelle appartient  $\mathbf{p}_{i_0}$ , par construction (cf.

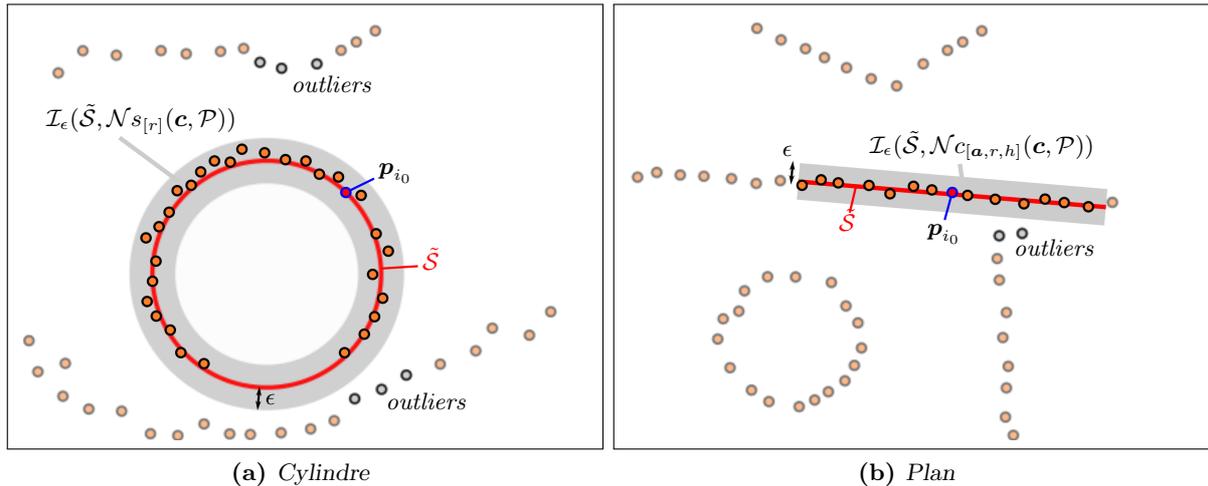
figure 4.5). En outre, la quantité de primitives représentées dans ce voisinage doit être faible. Dans ces conditions, l'algorithme du RANSAC permet de retrouver, après une faible quantité d'essais, la primitive échantillonnée par la plus grande quantité de points, c'est-à-dire la primitive à laquelle appartient  $\mathbf{p}_{i_0}$  si les hypothèses précédentes sont vérifiées.

On peut de plus contraindre la primitive reconnue, de manière à ce qu'elle passe par le point  $\mathbf{p}_{i_0}$ . Pour ce faire, il suffit de faire en sorte que chaque quorum du RANSAC contienne  $\mathbf{p}_{i_0}$ . L'intérêt de cette démarche tient au fait qu'elle permet de fixer une contrainte fiable, et diminue ainsi la dimension de l'espace de recherche. Typiquement, pour générer un plan, seuls deux points doivent être choisis aléatoirement, puisque le troisième est fixé ( $\mathbf{p}_{i_0}$ ). Ceci permet en théorie d'accélérer considérablement la convergence de l'algorithme (cf. section 1.3.3).

Dans la situation décrite ici, il n'est pas nécessaire de recourir à l'échantillonnage par cellule d'*octree* proposée par les auteurs de [SWK07], puisque les points appartiennent déjà tous à un voisinage pertinent. En revanche, la méthode proposée pour l'accélération du calcul de score permet certaines économies de ressources, et peut donc être utilisée. On utilise pour le RANSAC le seuil de bruit  $\epsilon$  introduit en section 2.3.1 - équation 2.18.

La primitive  $\tilde{\mathcal{S}}$  obtenue à l'issue de ce processus fournit une estimation grossière de la forme approximant au mieux les données. Le RANSAC permet en effet de calculer une forme minimisant une erreur simpliste, basée sur le décompte des *inliers*. Cette métrique n'est pas la plus appropriée, mais permet en revanche une évaluation très rapide pour la comparaison de la qualité des formes au sens de cet ajustement. Bien qu'elle puisse être améliorée, la primitive résultante permet toutefois d'exclure les *outliers* du voisinage  $\mathcal{N}$ , en ne retenant que ses *inliers*:

$$\mathcal{I}_\epsilon(\tilde{\mathcal{S}}, \mathcal{N}) = \{\mathbf{p} \in \mathcal{N} \mid d(\tilde{\mathcal{S}}, \mathbf{p}) \leq \epsilon\} \quad (4.8)$$



**Figure 4.6:** Première estimation  $\tilde{\mathcal{S}}$  du candidat généré à partir du voisinage issu de la graine  $\mathbf{p}_{i_0}$ . Partant de cette estimation, on retient l'ensemble des *inliers*  $\mathcal{I}_\epsilon(\tilde{\mathcal{S}}, \mathcal{N})$  que l'on utilisera par la suite pour raffiner cette estimation grossière.

### 4.2.3 Création à partir des points

Il est important, pour la suite du processus, que chaque candidat généré soit ajusté au mieux sur les données. En effet, compte tenu du coût de l'évaluation des candidats dans le cadre de la recherche du maximum *a posteriori*, et de la sensibilité de la mesure d'adéquation aux données (cf. section 2.3.1), une erreur mal maîtrisée dans l'évaluation des paramètres d'un candidat pourrait entraîner son refus.

Il est donc préférable de prendre le temps de raffiner chaque candidat en calculant les paramètres réalisant l'ajustement géométrique optimal sur l'ensemble  $\mathcal{I}_\epsilon(\tilde{\mathcal{S}}, \mathcal{N})$ , afin de maximiser la pertinence du candidat vis-à-vis des données. Pour ce faire, on utilise l'approche habituelle de minimisation au sens des moindres carrés, de manière à construire le candidat :

$$\mathcal{S} = \underset{\mathcal{C}}{\operatorname{argmin}} \sum_{\mathbf{p} \in \mathcal{I}_\epsilon(\tilde{\mathcal{S}}, \mathcal{N})} d(\mathcal{C}, \mathbf{p})^2 \quad (4.9)$$

$d(\mathcal{C}, \mathbf{p})$  désignant la distance Euclidienne du point  $\mathbf{p}$  à la primitive  $\mathcal{C}$  (cf. annexe A).

Dans le cas du plan, le problème peut être résolu via la décomposition en valeurs et vecteurs propres de la matrice de variance-covariance du nuage de points  $\mathcal{I}_\epsilon(\tilde{\mathcal{S}}, \mathcal{N})$  (cf. section 1.2.3).

Pour le cylindre en revanche, la distance est une fonction non linéaire des paramètres de la primitive. Comme le propose par exemple T. Chaperon [Cha02], ce problème d'optimisation peut être résolu à l'aide de l'algorithme de Levenberg-Marquardt. Cette méthode permet, partant d'une solution initiale, de progresser itérativement vers la solution optimale en se basant sur les propriétés différentielles de la fonction à optimiser (se référer à la thèse de T. Chaperon [Cha02] pour une formulation des distances et ses dérivées pour divers types de primitives, dont le cylindre, en vue de l'optimisation). Il repose sur un paramètre d'amortissement conférant à l'algorithme un comportement interpolant l'algorithme de Gauss-Newton et la descente de gradients. Tout comme ces deux méthodes, l'algorithme de Levenberg-Marquardt n'est pas à l'abri de rester piégé dans un minimum local de la fonction cible. Pour obtenir un optimum global, il est donc nécessaire de lui fournir une estimation initiale relativement proche de la solution effective. L'estimation  $\tilde{\mathcal{S}}$  calculée lors du filtrage des *inliers* fournit une bonne approximation de la solution, et peut donc être utilisée pour initialiser l'algorithme.

Chaque itération de l'algorithme de résolution nécessite l'évaluation de la fonction cible et de ses dérivées. Or, ces fonctions se formulent comme des sommes sur l'ensemble des points de  $\mathcal{I}_\epsilon(\tilde{\mathcal{S}}, \mathcal{N})$ . Pour accélérer la convergence de l'algorithme, il peut être intéressant de ne conserver dans l'équation 4.9 qu'une quantité restreinte de points choisis aléatoirement parmi  $\mathcal{I}_\epsilon(\tilde{\mathcal{S}}, \mathcal{N})$  afin de diminuer le coût de l'algorithme de résolution.

La primitive  $\mathcal{S}$  satisfaisant l'équation 4.9 constitue finalement le candidat proposé par notre générateur de primitives.

### 4.2.4 Analyse du processus et amélioration

#### Processus probabilisé

Comme nous le mentionnions précédemment, le processus de création de primitives à partir de points sélectionnés dans le nuage permet de ne générer qu'une quantité finie dénombrable de primitives. Le nombre d'objets que l'on peut construire s'obtient en

calculant le nombre de combinaisons de plus de  $n$  points au sein du nuage ( $n$  désignant le nombre minimal de points nécessaires à la création d'une primitive de type donné), après décompte des combinaisons distinctes engendrant une même primitive. Cette quantité est bien entendu déraisonnable, mais elle permet de s'assurer que les objets ne s'appuyant sur aucun point n'ont aucune chance d'être créés.

Ce processus définit ainsi un espace probabilisé dont l'univers des réalisables est l'ensemble des formes pouvant être créées à partir de points de  $\mathcal{P}$ . La probabilité associée à l'apparition d'une primitive  $\mathcal{S}$  correspond à la probabilité de sélectionner un ensemble de points tels que  $\mathcal{S}$  vérifie l'équation 4.9 vis-à-vis de ces points. La formulation exacte de cette probabilité est extrêmement complexe à obtenir, puisqu'elle met en jeu une multitude de processus (aléatoires ou déterministes) qui se combinent pour obtenir le résultat final. Nous ne sommes donc pas en mesure de fournir une probabilité pour toute primitive, et la méthode proposée ne se prête donc pas à l'exploration stochastique dans le cadre de méthodes de Monte Carlo à base de Chaînes de Markov (pour assurer la réversibilité des chaînes, on doit pouvoir donner la probabilité de voir apparaître un élément). Elle convient toutefois parfaitement à la reconstruction de modèles CAO via l'algorithme 5 que nous proposons en section 3.3.2.

### Pertinence des candidats

La pertinence des primitives générées vis-à-vis des données repose avant tout sur le choix de la graine  $\mathbf{p}_{i_0}$ . En effet, si la graine est sélectionnée sur une surface ne correspondant pas à la primitive de référence  $\mathcal{S}^i \in \mathcal{M}_0$ , il est probable que le résultat généré ne soit pas cohérent vis-à-vis du nuage de points, puisque le reste du processus de création repose sur l'hypothèse selon laquelle la surface à reconstruire est similaire à  $\mathcal{S}^i$ . Inversement, si la graine appartient à une surface pertinente vis-à-vis de  $\mathcal{S}^i$ , il est très probable que la primitive issue du processus de création s'appuie effectivement sur le nuage de points, du fait des diverses étapes visant à s'assurer de ce fait (choix du voisinage, suppression des *outliers* et ajustement fin sur les données restantes).

Or, plus les tolérances  $\sigma_p$  et  $\sigma_o$  vis-à-vis des changements sont élevées, plus il est probable que la graine sélectionnée soit sur une surface ne correspondant pas à  $\mathcal{S}^i$ . La pertinence des formes générées dépend ainsi directement des changements attendus entre les scènes représentées par l'*a priori*  $\mathcal{M}_0$  et le nuage de points  $\mathcal{P}$ , et l'exploration de l'espace des primitives sera évidemment d'autant plus efficace que  $\mathcal{M}_0$  est semblable à  $\mathcal{P}$  (intuitivement, plus l'*a priori* est fiable, plus on peut se reposer dessus et plus les formes proposées seront elles-mêmes fiables).

### Biais de l'échantillonnage

L'approche de génération de primitives que nous proposons ici présente toutefois un biais non négligeable dû à l'échantillonnage des objets par les points de  $\mathcal{P}$ . En effet, la probabilité de tirer une graine à la surface d'un objet représenté dans le nuage de points s'exprime comme la somme des probabilités de tirage des points échantillonnant cet objet (puisque les tirages sont indépendants). Or, plus cet objet est imposant, plus la quantité de points à sa surface est élevée. L'hétérogénéité dans la densité des points du nuage (cf. section 1.2.4) participe aussi à ce déséquilibre. Finalement, toutes les primitives représentées dans  $\mathcal{P}$  n'ont donc pas les mêmes chances d'apparaître, et celles contenant le moins de points à leur surface (les plus petites notamment) sont clairement défavorisées.

Pour corriger ce biais, il nous faut modifier la méthode de sélection de la graine en y introduisant un second biais destiné à équilibrer la probabilité de voir apparaître chaque primitive, indépendamment de son échantillonnage. L'idée consiste en la pénalisation de la probabilité associée aux points échantillonnant les primitives les plus représentées dans le nuage de points  $\mathcal{P}$ . Toute la difficulté de cette approche réside dans la nécessité de connaître le nombre de points par primitive pour contrebalancer l'effet de l'échantillonnage des objets. Or, ceci n'est possible que lorsque l'on connaît la reconstruction de  $\mathcal{P}$ , que l'on cherche justement à calculer. Ne disposant pas de ce résultat, on peut cependant proposer une démarche assimilable à un apprentissage basé sur l'historique des primitives proposées jusqu'alors. Pour estimer l'échantillonnage des primitives à l'issue de la reconstruction, il semble effectivement opportun d'utiliser le modèle CAO  $\mathcal{X}$  en cours de construction, fournissant une approximation de plus en plus précise du résultat au fil des itérations.

Dans cette démarche, il faut toutefois prendre garde à ne pas léser les points qui ne seraient pas encore rattachés à une primitive reconstruite. L'exploration du nuage de points doit effectivement faire en sorte d'harmoniser la sélection des points en s'appuyant sur les objets d'ores et déjà reconnus, mais ceux non encore traités par la reconstruction ne portent aucune information, et ne peuvent pourtant pas être ignorés. Faute de connaissances, les points non associés à une primitive ne peuvent donc pas être pénalisés, et tous se valent. Parallèlement, ceux associés à des primitives de  $\mathcal{X}$  doivent tenir compte de l'échantillonnage de leur primitive dans la course à la candidature au statut de graine.

Nous proposons ainsi une modification de l'algorithme d'acceptation-rejet 7 de manière à intégrer le biais d'échantillonnage. Cette méthode, décrite par l'algorithme 8, repose sur la segmentation  $S(\mathcal{P}, \mathcal{X}) = \{\mathcal{P}_0, \dots, \mathcal{P}_{|\mathcal{X}|}\}$  de  $\mathcal{P}$  correspondant à  $\mathcal{X}$ . Cette segmentation se définit comme une partition de  $\mathcal{P}$  telle que:

- Pour tout  $i < |\mathcal{X}|$ ,  $\mathcal{P}_i \subset \mathcal{P}$  désigne l'ensemble des points associés à la primitive  $\mathcal{S}_i \in \mathcal{X}$ .
- $\mathcal{P}_{|\mathcal{X}|}$  représente l'ensemble de points non affectés à une primitive.

On considère qu'un point  $\mathbf{p}$  est associé à une primitive  $\mathcal{S}_i \in \mathcal{X}$  si  $\mathcal{S}_i$  est la forme la plus proche de  $\mathbf{p}$  et si la distance qui sépare le point de la primitive est inférieure au seuil  $\epsilon$ .

Si  $Y$  désigne une variable aléatoire définie sur le nuage de points  $\mathcal{P}$ , la probabilité de voir survenir l'événement  $Y = \mathbf{p}$  en sortie de l'algorithme 8 se définit comme:

$$P(Y = \mathbf{p} | u \leq P_S(\mathbf{p} | \mathcal{S}^i)) \stackrel{\text{Bayes}}{=} \frac{P(u \leq P_S(\mathbf{p} | \mathcal{S}^i) | Y = \mathbf{p}) \times P(Y = \mathbf{p})}{P(u \leq P_S(\mathbf{p} | \mathcal{S}^i))}$$

où, par définition

$$P(u \leq P_S(\mathbf{p} | \mathcal{S}^i)) = \sum_{\mathbf{q} \in \mathcal{P}} P(u \leq P_S(\mathbf{q} | \mathcal{S}^i) | Y = \mathbf{q}) \times P(Y = \mathbf{q})$$

Or, on constate aisément que  $P(u \leq P_S(\mathbf{p} | \mathcal{S}^i)) = P_S(\mathbf{p} | \mathcal{S}^i)$  pour tout  $\mathbf{p}$  du nuage  $\mathcal{P}$ . L'algorithme 8 simule donc la probabilité:

$$P(Y = \mathbf{p} | u \leq P_S(\mathbf{p} | \mathcal{S}^i)) = \frac{P_S(\mathbf{p} | \mathcal{S}^i) \times P(Y = \mathbf{p})}{\sum_{\mathbf{q} \in \mathcal{P}} P_S(\mathbf{q} | \mathcal{S}^i) \times P(Y = \mathbf{q})} \quad (4.10)$$

**Algorithme 8** : Générateur de graine biaisé.

---

**Données** :  $\mathcal{S}^i$ : primitive de référence dans  $\mathcal{M}_0$ .  
 $P_S$  : probabilité mesurant la pertinence d'un point vis-à-vis de  $\mathcal{S}^i$ .  
 $S(\mathcal{P}, \mathcal{X}) = \{\mathcal{P}_0, \dots, \mathcal{P}_{|\mathcal{X}|}\}$  : segmentation du nuage de points  $\mathcal{P}$  associée au modèle reconstruit courant  $\mathcal{X}$ .

**Résultat** :  $\mathbf{p}$  choisi dans  $\mathcal{P}$  selon une probabilité dépendant de  $P_S$  et tenant compte de l'échantillonnage des primitives de  $\mathcal{X}$

```

1 Tant que VRAI faire                                     /* Boucle infinie */
2   Générer un réel  $v \in ]0, 1]$  selon une loi uniforme;
3   Si  $v \leq \frac{|\mathcal{P}_{|\mathcal{X}|}}{|\mathcal{P}|}$  alors
4     // Choix d'un point parmi ceux non encore affectés
4     Choisir un point  $\mathbf{p}$  dans  $\mathcal{P}_{|\mathcal{X}|}$  selon une loi uniforme;
5   Sinon
6     // Choix d'un point sur une primitive
6     Choisir un entier  $k \in [0, |\mathcal{X}|[$  selon une loi uniforme;
7     Choisir un point  $\mathbf{p}$  dans  $\mathcal{P}_k$  selon une loi uniforme;
8   Générer un réel  $u \in ]0, 1]$  selon une loi uniforme;
9   Si  $u \leq P_S(\mathbf{p}|\mathcal{S}^i)$  alors
10  | Retourner  $\mathbf{p}$ 

```

---

Pour analyser le comportement de cet algorithme, on peut ignorer le dénominateur de l'équation 4.10, puisqu'il ne fait office que de terme de normalisation des probabilités (ce terme est constant, quelle que soit la valeur  $\mathbf{p}$  prise par la variable aléatoire  $Y$ ). En revanche, on constate que la probabilité de choisir un point  $\mathbf{p}$  en tant que graine dépend du produit de la probabilité  $P_S(\mathbf{p}|\mathcal{S}^i)$  et de la probabilité  $P(Y = \mathbf{p})$  définissant de noyau de proposition des points candidats. Ce dernier terme nous permet donc de définir la correction du biais d'échantillonnage. Puisque les divers tirages des points candidats sont indépendants, l'expression de  $P(Y = \mathbf{p})$  ne dépend pas du nombre d'itérations de la boucle principale. Cette expression diffère toutefois selon deux situations.

**Si  $\mathbf{p}$  n'est associé à aucune primitive de  $\mathcal{X}$  (donc  $\mathbf{p} \in \mathcal{P}_{|\mathcal{X}|}$ )**

$$P(Y = \mathbf{p}) = \underbrace{\frac{|\mathcal{P}_{|\mathcal{X}|}}{|\mathcal{P}|}}_{\text{ligne 3}} \times \underbrace{\frac{1}{|\mathcal{P}_{|\mathcal{X}|}}}_{\text{ligne 4}}$$

d'où

$$P(Y = \mathbf{p}) = \frac{1}{|\mathcal{P}|} \quad (4.11)$$

La probabilité de choisir un point candidat parmi les points non affectés dans l'algorithme 8 est donc inchangée par rapport à la méthode d'origine. Le fait qu'il existe des points déjà affectés à des primitives est donc complètement transparent aux points non encore traités.

Si  $\mathbf{p}$  est associé à une primitive  $\mathcal{S}_j \in \mathcal{X}$  (donc  $\mathbf{p} \in \mathcal{P}_j$ , avec  $j < |\mathcal{X}|$ )

$$P(Y = \mathbf{p}) = \underbrace{\left(1 - \frac{|\mathcal{P}_{|\mathcal{X}|}}{|\mathcal{P}|}\right)}_{\text{ligne 3}} \times \underbrace{\frac{1}{|\mathcal{X}|}}_{\text{ligne 6}} \times \underbrace{\frac{1}{|\mathcal{P}_j|}}_{\text{ligne 7}} \quad (4.12)$$

Les deux termes de gauche dans le membre de droite de cette équation sont indépendants du point  $\mathbf{p}$  considéré dans le nuage de points. La différence porte sur le terme de droite: en substituant l'équation 4.12 dans l'équation 4.10, on constate que plus une primitive  $\mathcal{S}_j$  est échantillonnée (donc plus  $|\mathcal{P}_j|$  augmente), plus la probabilité individuelle des points qui lui sont associés diminue. Ses points sont donc pénalisés conformément à ce que nous souhaitons.

En revanche, si l'on étudie la probabilité à l'échelle de la primitive, on constate que :

$$\begin{aligned} P(Y \in \mathcal{P}_j | u \leq P_S(\mathbf{p}|\mathcal{S}^i)) &= \sum_{\mathbf{p} \in \mathcal{P}_j} P(Y = \mathbf{p} | u \leq P_S(\mathbf{p}|\mathcal{S}^i)) \\ &= \sum_{\mathbf{p} \in \mathcal{P}_j} \frac{P_S(\mathbf{p}|\mathcal{S}^i) \times P(Y = \mathbf{p})}{Z_0} \\ &= \frac{Z_1}{|\mathcal{P}_j|} \times \sum_{\mathbf{p} \in \mathcal{P}_j} P_S(\mathbf{p}|\mathcal{S}^i) \end{aligned} \quad (4.13)$$

$Z_0$  étant le dénominateur de l'équation 4.10, et

$$Z_1 = \left(1 - \frac{|\mathcal{P}_{|\mathcal{X}|}}{|\mathcal{P}|}\right) \times \frac{1}{Z_0 \times |\mathcal{X}|}$$

On constate ainsi que la méthode proposée a pour principal effet de moyenniser les probabilités des points définis sur une même primitive dans le processus du choix de la graine. Finalement, le processus de création est ainsi harmonisé à l'échelle des objets, indépendamment de leur échantillonnage. Seule compte désormais la pertinence des points à la surface des primitives vis à vis du référent  $\mathcal{S}^i$ , pas leur quantité.

**Remarque** On note que l'algorithme 8 correspond à la répétition d'une expérience de Bernouilli, dont la probabilité de succès vaut:

$$P(u \leq P_S(\mathbf{p}|\mathcal{S}^i)) = \sum_{\mathbf{p} \in \mathcal{P}} P(u \leq P_S(\mathbf{p}|\mathcal{S}^i) | Y = \mathbf{p}) \times P(Y = \mathbf{p}) = Z_0 \quad (4.14)$$

La variable aléatoire  $N$  définie sur  $\mathbb{N}$  comptant le nombre de répétitions de cette expérience avant l'obtention d'un succès suit donc une loi géométrique de paramètre  $Z_0$ . Ceci permet en particulier de déterminer, en termes probabilistes, le nombre moyen d'itérations de l'algorithme 8, via l'espérance de  $N$ :

$$E(N) = \frac{1}{Z_0} \quad (4.15)$$

Cette valeur est en général très grande sur des nuages de points contenant une grande quantité d'éléments dont la probabilité  $P_S$  est faible (intuitivement, plus la proportion d'éléments acceptables est faible, plus les efforts nécessaires pour la recherche de l'un d'entre eux sont conséquents).

### Méthode alternative

Considérons la situation selon laquelle chaque primitive échantillonnée par  $\mathcal{P}$  correspond à une primitive de  $\mathcal{M}_0$  (rappelons que ce n'est pas toujours le cas puisqu'il se peut que des objets de la scène représentée par  $\mathcal{P}$  ne correspondent à aucun élément de  $\mathcal{M}_0$ ). On sait alors que toutes les parties du nuage peuvent être reconstruites en s'appuyant sur des primitives de *l'a priori*. Dans cette situation, le raisonnement que nous proposons pour la génération de primitives candidates peut être inversé. Au lieu de sélectionner une graine correspondant à une primitive de référence  $\mathcal{S}^i \in \mathcal{M}_0$ , on peut en effet se donner un point  $\mathbf{p}_{i_0}$  (choisi par exemple aléatoirement dans le nuage  $\mathcal{P}$ ), et tirer ensuite une primitive  $\mathcal{S}^i$  dans  $\mathcal{M}_0$  en simulant la probabilité  $P_S(\mathcal{S}^j | \mathbf{p}_{i_0})$ . Or, d'après la règle de Bayes:

$$P_S(\mathcal{S}^j | \mathbf{p}_{i_0}) = P_S(\mathbf{p}_{i_0} | \mathcal{S}^j) \frac{P(\mathcal{S}^j)}{P(\mathbf{p}_{i_0})}$$

En l'absence d'*a priori* pour privilégier le choix d'un élément de  $\mathcal{M}_0$ , la probabilité  $P(\mathcal{S}^j)$  est uniforme sur  $\mathcal{M}_0$ . De sorte que le ratio  $P(\mathcal{S}^j)$  sur  $P(\mathbf{p}_{i_0})$  est indépendant de  $\mathcal{S}^j$ , et peut donc être considéré comme une constante que peuvent ignorer les algorithmes simulant le choix d'un élément  $\mathcal{S}^j$  selon la probabilité  $P_S(\mathcal{S}^j | \mathbf{p}_{i_0})$ . En conclusion, il nous suffit donc d'utiliser la vraisemblance  $P_S(\mathbf{p}_{i_0} | \mathcal{S}^j)$  avec  $\mathbf{p}_{i_0}$  fixé pour simuler le choix d'un élément de  $\mathcal{M}_0$  selon la probabilité  $P_S(\mathcal{S}^j | \mathbf{p}_{i_0})$ , en utilisant au choix l'algorithme de Metropolis-Hastings ou l'algorithme d'acceptation-rejet par exemple.

Cette approche inverse possède quelques avantages vis-à-vis de la méthode que nous avons initialement proposée. Étant donné le fait que le nombre de points dans  $\mathcal{P}$  est en général nettement plus élevé que le nombre de primitives de  $\mathcal{M}_0$ , le choix d'une graine au sein de  $\mathcal{P}$  est plus coûteux (en termes de quantité d'opérations du fait de la répétition des essais) que le choix d'une primitive au sein de  $\mathcal{M}_0$ . On s'abstrait de plus du biais introduit par l'échantillonnage des primitives dans  $\mathcal{P}$ . Ce problème n'existe pas dans le cas où les primitives de  $\mathcal{M}_0$  sont choisies après sélection d'un point de  $\mathcal{P}$ .

On obtient ainsi un gain en terme d'efficacité, mais cette optimisation n'est que mineure en pratique puisque le choix de la graine n'est pas nécessairement l'opération la plus lourde dans le processus de création. En pratique, nous préférons utiliser la démarche procédant par choix d'une graine à partir d'un référent, puisqu'elle ne présuppose rien au sujet de  $\mathcal{P}$  et  $\mathcal{M}_0$ , contrairement à cette méthode alternative.

## 4.3 Borner les primitives

Les étapes de génération de primitives décrites dans les sections précédentes n'abordent pas le problème du calcul de la limite des formes. En effet, les plans et cylindres générés sont considérés comme infinis par défaut. L'importance de ce problème est considérable, puisque les limites des primitives générés serviront par la suite au calcul des connexions entre les éléments du modèle reconstruit (rappelons que l'on considère que deux primitives sont considérées comme connectées si leur intersection dans  $\mathbb{R}^3$  est non vide).

Pour calculer ces limites, il nous faut utiliser le nuage de points. On considère alors que la primitive n'est effectivement définie que sur la section où elle est supportée par des points de  $\mathcal{P}$ . On apportera toutefois une distinction dans cette définition du support, donnant lieu à deux calculs de limites.

En effet, certaines primitives qui ne sont pas directement connectées physiquement (typiquement les cylindres reliés par des coudes) doivent malgré tout apparaître comme

dépendantes, puisque liées par un objet intermédiaire. Il est ainsi souhaitable de surévaluer suffisamment l'extension spatiale des primitives pour permettre à ces configurations d'être identifiées comme des connexions. Par ailleurs, on doit se donner une description aussi précise que possible de la section effective des primitives dans le nuage de points, pour permettre la reconstruction de modèles CAO fiables. En effet seules les parties du modèle CAO qui sont effectivement supportées par des points du nuage peuvent être jugées comme étant fiables.

### 4.3.1 Limites grossières

Considérons qu'un point supporte une primitive si sa contribution énergétique d'attache aux données  $h_D$  est négative (cf. section 2.3.1 - équation 2.18), c'est-à-dire si le point correspond effectivement à au processus d'échantillonnage de la primitive. Cette définition permet ainsi à la primitive de s'étendre sur la totalité de la section où elle se trouve soutenue par un point au moins, sur la base de critères géométriques simples (distances et conformité des normales). En ce sens, les limites issues de ce critère sont grossières puisque l'on ne tient pas compte de la qualité réelle de la description de la primitive par le nuage, et une poignée de points (des *outliers* par exemple) peuvent allonger significativement l'étendue spatiale de l'objet généré.

L'intérêt des limites ainsi définies tient au fait qu'elles permettent à des objets partageant des points en commun de se connecter. C'est donc sur la base de ce premier critère que l'on définit les limites des primitives en vue des calculs de connexion.

### 4.3.2 Limites effectives

Les limites grossières mentionnées ci-avant, bien qu'elles soient pertinentes pour le calcul de connexion, ne représentent pas exactement la portion effective de la forme décrite par le nuage de points. Elles en proposent une surévaluation permettant la collision des formes partageant des points du nuage.

Pour cerner les bornes effectives, il nous faut considérer la répartition statistique des points autour de la forme. Sur une section donnée de la primitive, si la proportion de points échantillonnant correctement la forme (apportant une contribution  $h_D$  négative) est significativement supérieure à la quantité de points qui la discréditent (apportant une contribution  $h_D$  positive), on pourra alors considérer que cette section est effectivement supportée par le nuage de points. Dans le cas contraire, la section considérée ne correspond pas à un objet décrit par le nuage de points, et doit donc être ignorée.

Intuitivement, là où les extrémités grossières définissent l'endroit à partir duquel la primitive n'est plus décrite par les points du nuage, les extrémités effectives définissent l'endroit à partir duquel la primitive est mal décrite par les points du nuage.

Cette définition alternative des bornes des primitives doit être conservée en marge du candidat. En pratique, c'est la primitive définie par les limites effectives qui est exportée en tant que résultat, les limites grossières n'étant utilisées qu'en "interne" pour permettre les calculs.

### 4.3.3 Bornes du cylindre

Pour identifier les limites du cylindre  $\mathcal{S}$  vis-à-vis du nuage de points, nous proposons de le découper le long de son axe en tranches contiguës d'épaisseur  $\zeta$ . La tranche  $\mathcal{T}_i$  se définit ainsi comme l'ensemble des points dont la distance à  $\mathcal{S}$  est inférieure au seuil  $\eta\epsilon$  (puisque les points dont la distance excèdent ce seuil ne participent pas à la contribution

énergétique vis-à-vis de  $\mathcal{S}$  et ne sont donc pas pertinents pour la définition de ses bornes - cf. section 2.3.1) et se projetant sur la  $i^{\text{ème}}$  section le long de l'axe de  $\mathcal{S}$  engendré par le centre  $\mathbf{c}_{\mathcal{S}}$  et le vecteur directeur  $\mathbf{a}_{\mathcal{S}}$  :

$$\mathcal{T}_i = \left\{ \mathbf{p} \in \mathcal{P} \mid i \leq \frac{(\mathbf{p} - \mathbf{c}_{\mathcal{S}}) \cdot \mathbf{a}_{\mathcal{S}}}{\zeta} < i + 1 \text{ et } |d(\mathcal{S}, \mathbf{p})| \leq \eta\epsilon \right\} \quad \forall i \in \mathbb{N} \quad (4.16)$$

On distinguera notamment le sous-ensemble des points apportant une contribution négative:

$$\mathcal{T}_i^- = \{ \mathbf{p} \in \mathcal{T}_i \mid h_D(\{\mathcal{S}\}, \mathbf{p}) \leq 0 \} \quad (4.17)$$

Nous considérons de plus que le centre  $\mathbf{c}_{\mathcal{S}}$  du cylindre  $\mathcal{S}$  est positionné de manière à ce que la graine  $\mathbf{p}_{i_0}$  ayant servi à sa construction appartienne à la tranche  $\mathcal{T}_0$  (cf. figure 4.8).

**Limites grossières** Partant de ces diverses définitions, on peut ainsi déterminer les bornes grossières du cylindre en calculant l'ensemble connexe maximal de tranches comprenant  $\mathcal{T}_0$ , et dont toutes les tranches contiennent au moins un point apportant une contribution négative. En particulier, on distingue les deux tranches extrémales  $\mathcal{T}_m$  et  $\mathcal{T}_M$  de cet ensemble connexe, dont les indices  $m$  et  $M$  sont calculés comme suit:

$$m = \min \left\{ k \in \mathbb{N} \mid \forall j \in [k, 0], \left| \mathcal{T}_j^- \right| > 0 \right\} \quad (4.18)$$

$$M = \max \left\{ k \in \mathbb{N} \mid \forall j \in [0, k], \left| \mathcal{T}_j^- \right| > 0 \right\} \quad (4.19)$$

Ces deux tranches correspondent alors aux bornes grossières du cylindre. Entre ces deux extrémités, la section cylindrique est continument supportée par des points. On déduit ainsi la position du centre du  $\mathbf{c}_{\mathcal{S}'}$  cylindre borné  $\mathcal{S}'$

$$\mathbf{c}_{\mathcal{S}'} = \mathbf{c}_{\mathcal{S}} + \left( \frac{\zeta \times (m + M + 1)}{2} \right) \mathbf{a}_{\mathcal{S}} \quad (4.20)$$

ainsi que sa longueur  $l_{\mathcal{S}'}$

$$l_{\mathcal{S}'} = \zeta \times (M - m + 1) \quad (4.21)$$

Le rayon et le vecteur directeur de l'axe de  $\mathcal{S}'$  sont quant à eux identiques à ceux du cylindre infini  $\mathcal{S}$ .

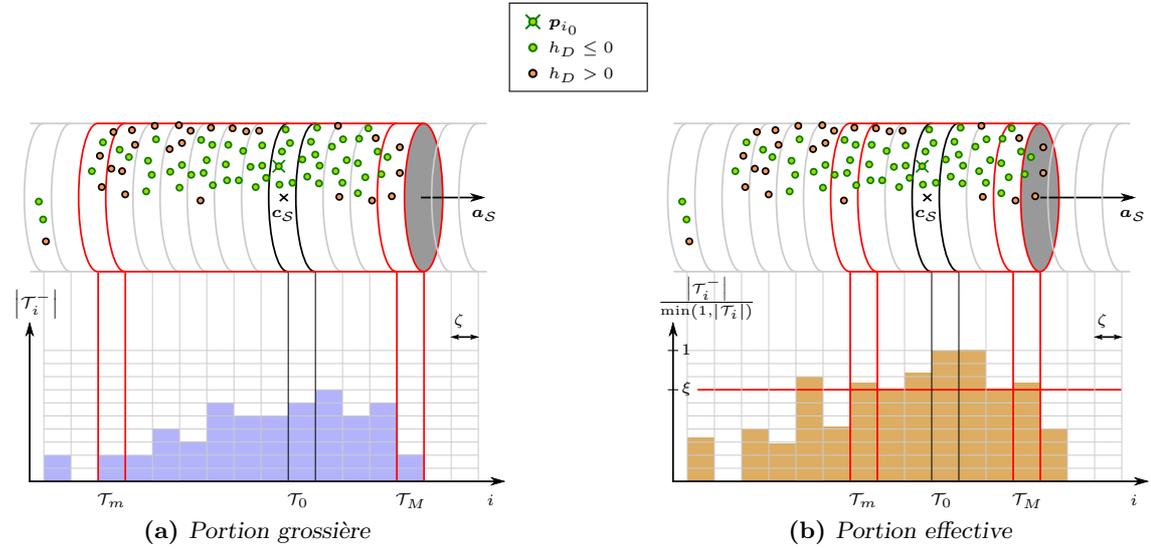
**Limites effectives** Sur le même principe du découpage en tranches, on peut isoler la portion vérifiant le critère établissant la section effective du cylindre. Le cylindre est ainsi effectivement défini sur l'ensemble connexe maximal de tranches comprenant  $\mathcal{T}_0$  et dont toutes les tranches possèdent une proportion minimale de points apportant une contribution négative. Les indices  $m$  et  $M$  des deux tranches extrémales  $\mathcal{T}_m$  et  $\mathcal{T}_M$  de cet ensemble connexe se définissent donc comme suit:

$$m = \min \left\{ k \in \mathbb{N} \mid \forall j \in [k, 0], \frac{\left| \mathcal{T}_j^- \right|}{\max(1, |\mathcal{T}_j|)} \geq \xi \right\} \quad (4.22)$$

$$M = \max \left\{ k \in \mathbb{N} \mid \forall j \in [0, k], \frac{|\mathcal{T}_j^-|}{\max(1, |\mathcal{T}_j|)} \geq \xi \right\} \quad (4.23)$$

Le paramètre réel  $\xi \in ]0, 1]$  permet de moduler la qualité attendue pour la description de chaque tranche par les points. Pour être relativement permissif, nous utilisons typiquement dans nos expériences un seuil de l'ordre de  $\xi = 0.7$ .

Une fois  $m$  et  $M$  déterminés, le centre  $\mathbf{c}_{\mathcal{S}'}$  et la longueur  $l_{\mathcal{S}'}$  du cylindre effectif  $\mathcal{S}'$  peuvent être calculés en réutilisant les équations 4.20 et 4.21 respectivement.



**Figure 4.7:** Calcul des extrémités d'un cylindre candidat. On distingue deux définitions pour les extrémités. La première (gauche) surévalue la longueur du cylindre de manière à permettre le calcul de connexion entre formes. La seconde (droite) estime avec précision ses limites sur la base de la qualité de la description de la primitive par le nuage.

#### 4.3.4 Bornes du plan

Rappelons que, pour simplifier les traitements, nous considérons que les portions planes sont délimitées par un rectangle. La détermination de l'orientation  $\mathbf{d}_{\mathcal{S}}$  d'un plan  $\mathcal{S}$  autour de sa normale  $\mathbf{n}_{\mathcal{S}}$  sur la base de l'analyse du nuage de points étant une tâche relativement complexe, on supposera <sup>2</sup> de plus que les bords de plans sont alignés selon les axes canoniques  $X$ ,  $Y$  et  $Z$  en fonction de l'orientation principale de la normale  $\mathbf{n}_{\mathcal{S}}$ .

Pour évaluer les bornes du plan  $\mathcal{S}$ , on s'appuie sur la construction d'une grille régulière à sa surface. Dans la même idée que pour le traitement du cylindre, chaque cellule carrée  $\mathcal{C}_{(i,j)}$  de coordonnées  $(i, j) \in \mathbb{N}^2$  et de largeur  $\zeta$  dans cette grille se voit affecter l'ensemble des points proches de  $\mathcal{S}$  qui se projettent en son intérieur:

$$\mathcal{C}_{(i,j)} = \left\{ \mathbf{p} \in \mathcal{P} \mid i \leq \frac{w(\mathcal{S}, \mathbf{p})}{\zeta} < i + 1 \text{ et } j \leq \frac{l(\mathcal{S}, \mathbf{p})}{\zeta} < j + 1 \text{ et } d(\mathcal{S}, \mathbf{p}) \leq \eta \epsilon \right\} \quad (4.24)$$

<sup>2</sup>L'hypothèse de l'alignement selon les trois axes canoniques se base sur le constat que la plupart des scènes industrielles sont effectivement structurées selon ces trois directions lorsqu'elles sont exprimées dans le repère approprié (ce qui est souvent le cas en pratique). Le problème est toutefois plus complexe pour les bâtiments à structure annulaire, puisqu'il n'existe dans ce cas aucune autre direction privilégiée que l'axe vertical  $Z$ .

avec

$$\begin{aligned} w(\mathcal{S}, \mathbf{p}) &= \mathbf{d}_{\mathcal{S}} \cdot (\mathbf{p} - \mathbf{c}_{\mathcal{S}}) \\ l(\mathcal{S}, \mathbf{p}) &= (\mathbf{d}_{\mathcal{S}} \times \mathbf{n}_{\mathcal{S}}) \cdot (\mathbf{p} - \mathbf{c}_{\mathcal{S}}) \end{aligned} \quad (4.25)$$

Rappelons que  $\mathbf{n}_{\mathcal{S}}$  désigne le vecteur unitaire normal au plan  $\mathcal{S}$ , tandis que  $\mathbf{c}_{\mathcal{S}}$  correspond à son centre et  $\mathbf{d}_{\mathcal{S}}$  est un vecteur unitaire définissant son orientation autour de  $\mathbf{n}_{\mathcal{S}}$  ( $\mathbf{d}_{\mathcal{S}}$  et  $\mathbf{n}_{\mathcal{S}}$  sont orthogonaux).  $\mathcal{S}$  étant non encore borné, tout point à la surface de ce plan peut être défini comme son centre. On gardera toutefois en tant que centre  $\mathbf{c}_{\mathcal{S}}$  la projection  $\mathbf{p}_{i_0} \downarrow_{\mathcal{S}}$  sur  $\mathcal{S}$  de la graine  $\mathbf{p}_{i_0}$  ayant servi à sa construction, de manière à ce que la cellule  $\mathcal{C}_{(0,0)}$  contienne  $\mathbf{p}_{i_0}$ .

On associe de plus à chaque cellule  $\mathcal{C}_{(i,j)}$  le sous ensemble  $\mathcal{C}_{(i,j)}^-$  correspondant aux points de  $\mathcal{C}_{(i,j)}$  apportant une contribution  $h_D$  négative vis-à-vis de  $\mathcal{S}$ :

$$\mathcal{C}_{(i,j)}^- = \{\mathbf{p} \in \mathcal{C}_{(i,j)} \mid h_D(\{\mathcal{S}\}, \mathbf{p}) \leq 0\} \quad (4.26)$$

**Limites grossières** Le calcul des limites grossières du plan  $\mathcal{S}$  s'effectue en recherchant la composante connexe contenant  $\mathcal{C}_{(0,0)}$  parmi l'ensemble des cellules  $\mathcal{C}_{(i,j)}^-$ . La connexité parmi les cellules se définit sur la base du 8 voisinage: deux cellules de coordonnées respectives  $(a, b) \in \mathbb{N}^2$  et  $(c, d) \in \mathbb{N}^2$  sont voisines si et seulement si  $\max(|a - c|, |b - d|) \leq 1$ .

Le calcul de la composante connexe peut alors être effectué en utilisant un algorithme de croissance de région initialisé à partir de l'ensemble atomique  $\mathbb{B} = \{\mathcal{C}_{(0,0)}\}$  (le principe des algorithmes par croissance de région est exposé en section 1.3.2). À chaque itération, une cellule  $\mathcal{C}_{(i,j)}^-$  peut être ajoutée à la composante connexe  $\mathbb{B}$  si :

- $|\mathcal{C}_{(i,j)}^-| > 0$ ,
- $\mathcal{C}_{(i,j)}^-$  est adjacente à  $\mathbb{B}$  (c'est-à-dire si  $\mathbb{B}$  contient une cellule voisine de  $\mathcal{C}_{(i,j)}^-$ ).

En pratique, la croissance de région peut être traitée efficacement en maintenant la liste des cellules adjacentes à  $\mathbb{B}$  au cours des itérations.

À l'issue du calcul de  $\mathbb{B}$ , on peut finalement déterminer les bornes du plan en identifiant les indices extrémaux  $w$ ,  $W$ ,  $l$  et  $L$  de ses cellules:

$$\begin{aligned} w &= \min \left\{ i \in \mathbb{N} \mid \exists j \in \mathbb{N} \text{ t.q. } \mathcal{C}_{(i,j)}^- \in \mathbb{B} \right\} \\ W &= \max \left\{ i \in \mathbb{N} \mid \exists j \in \mathbb{N} \text{ t.q. } \mathcal{C}_{(i,j)}^- \in \mathbb{B} \right\} \\ l &= \min \left\{ j \in \mathbb{N} \mid \exists i \in \mathbb{N} \text{ t.q. } \mathcal{C}_{(i,j)}^- \in \mathbb{B} \right\} \\ L &= \max \left\{ j \in \mathbb{N} \mid \exists i \in \mathbb{N} \text{ t.q. } \mathcal{C}_{(i,j)}^- \in \mathbb{B} \right\} \end{aligned} \quad (4.27)$$

À partir de ces indices, on peut calculer le centre  $\mathbf{c}_{\mathcal{S}'}$  de la portion grossière de plan  $\mathcal{S}'$

$$\mathbf{c}_{\mathcal{S}'} = \mathbf{c}_{\mathcal{S}} + \zeta \times \left( \left( \frac{w + W + 1}{2} \right) \mathbf{d}_{\mathcal{S}} + \left( \frac{l + L + 1}{2} \right) (\mathbf{d}_{\mathcal{S}} \times \mathbf{n}_{\mathcal{S}}) \right) \quad (4.28)$$

ainsi que sa largeur  $w_{\mathcal{S}'}$

$$w_{\mathcal{S}'} = \zeta \times (W - w + 1) \quad (4.29)$$

et sa longueur  $l_{\mathcal{S}'}$

$$l_{\mathcal{S}'} = \zeta \times (L - l + 1) \quad (4.30)$$

Le vecteur d'orientation et le vecteur normal au plan restent quant à eux inchangés.

**Limites effectives** Les limites effectives du plan peuvent être calculées selon le même principe que les limites grossières, en modifiant toutefois légèrement le critère utilisé pour la croissance de région. Une cellule  $\mathcal{C}_{(i,j)}^-$  peut être ajoutée à la composante connexe  $\mathbb{B}$  si :

- $\mathcal{C}_{(i,j)}^-$  est adjacente à  $\mathbb{B}$
- la proportion de points pertinents est suffisante:  $\frac{|\mathcal{C}_{(i,j)}^-|}{\max(1, |\mathcal{C}_{(i,j)}^-|)} \geq \xi$

le paramètre  $\xi$  ayant été introduit dans le cadre du calcul des limites effectives du cylindre.

La portion effective de plan  $\mathcal{S}''$  est ensuite déterminée en identifiant les coordonnées extrémales des cellules de  $\mathbb{B}$  (équation 4.27), à partir desquelles on peut ensuite calculer le centre, la largeur et la longueur de la portion effective (équations 4.28, 4.29 et 4.30 respectivement).

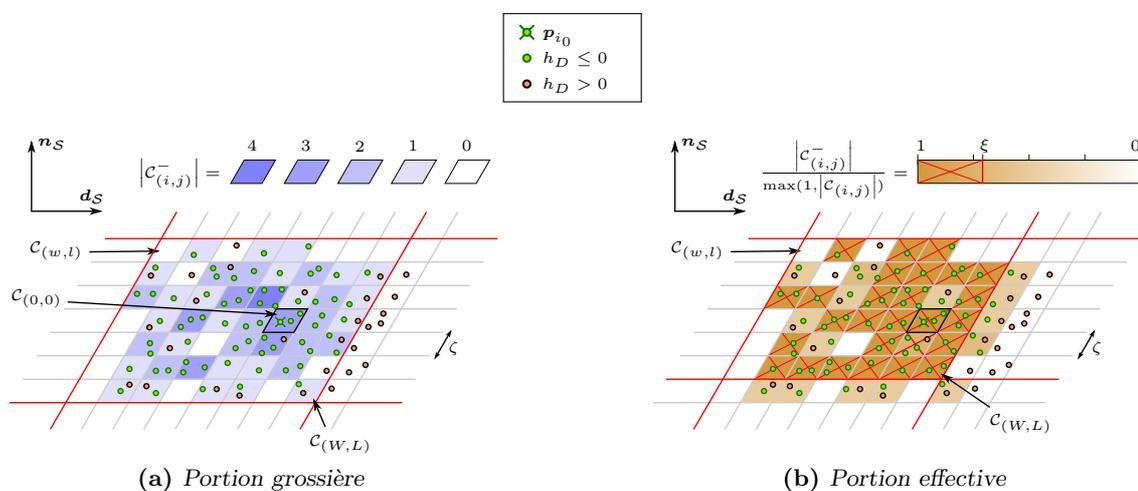


Figure 4.8: Calcul des extrémités d'un plan candidat.

### 4.3.5 Taille des tranches et cellules

La délimitation de cylindres et des plans repose sur un paramètre commun  $\zeta$  spécifiant la taille des cellules (ou tranches) utilisées. Ce paramètre sera de plus utilisé par la suite (section 4.4) pour permettre d'évaluer la couverture de la surface de chaque primitive par les points du nuage.

Le paramètre  $\zeta$  correspond intuitivement à une zone d'influence que possède chaque point à la surface de la primitive. Idéalement, les influences de points voisins doivent se recouvrir, pour signifier que le nuage de points représente bien la totalité de la zone recouverte. Le chevauchement des zones d'influence permet ainsi de délimiter les composantes connexes parmi les points présents à la surface de l'objet. Toutefois, ce paramètre doit aussi être suffisamment petit pour permettre d'estimer finement l'étendue des primitives.

Dans cette optique,  $\zeta$  peut être estimé à partir de la densité des points du nuage. On peut ainsi faire en sorte que chaque cellule à la surface d'un plan se voit attribuer de 3 à 10 points. Pour ce faire, on peut par exemple estimer  $\zeta$  à partir d'une centaine de points sélectionnés aléatoirement dans le nuage. La valeur affectée à  $\zeta$  est alors calculée comme étant la distance moyenne de ces points à leur cinquième (par exemple) plus proche

voisin dans le nuage. Ce processus entièrement automatique permet d'estimer aisément une bonne valeur pour le paramètre, quoiqu'une évaluation visuelle du nuage de points permette d'aboutir aussi simplement à des résultats similaires.

## 4.4 Calcul de la couverture

L'énergie d'attache aux données  $H_D$  (cf. section 2.3.1) intègre un terme de contrôle empêchant l'insertion, dans le modèle CAO reconstruit, de primitives ne vérifiant pas une adéquation minimale vis-à-vis du nuage de points. Cette adéquation se mesure en terme de couverture  $\Gamma(\mathcal{S}, \mathcal{P})$  de la surface de la primitive candidate  $\mathcal{S}$  par le nuage de points. Il s'agit de s'assurer qu'une proportion minimale  $\gamma_0$  de la surface de  $\mathcal{S}$  est effectivement échantillonnée, afin de s'assurer que toutes les primitives candidates à l'insertion dans le modèle s'appuient bien sur les points du nuage. Ce filtrage préalable permet de plus d'économiser des ressources de calcul en détectant au plus tôt les objets qui ne sont manifestement pas suffisamment pertinents pour la reconstruction.

La couverture  $\Gamma$  de la surface d'un objet par des points ne peut évidemment pas être calculée de manière exacte, puisque chaque point ne représente par définition qu'une portion infinitésimale de l'espace. En tant que somme finie de termes infinitésimaux, la proportion de la surface de  $\mathcal{S}$  effectivement représentée par des points serait donc elle-même insignifiante. On peut toutefois approximer  $\Gamma(\mathcal{S}, \mathcal{P})$  en considérant la surface de  $\mathcal{S}$  par morceaux, et en comptabilisant les parties représentées par un point de  $\mathcal{P}$  au moins. Pour être plus précis, nous ne considérons que les points de  $\mathcal{P}$  apportant une contribution  $h_D$  négative vis-à-vis de  $\mathcal{S}$ , puisque seuls ceux-ci correspondent potentiellement au processus d'échantillonnage de  $\mathcal{S}$ .

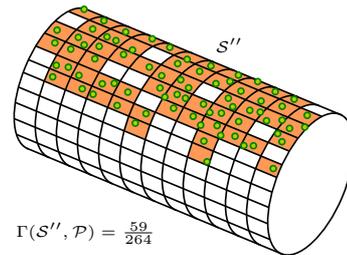
Nous considérons la portion effective  $\mathcal{S}''$  de la primitive  $\mathcal{S}$  pour l'estimation de  $\Gamma(\mathcal{S}, \mathcal{P})$ , puisqu'elle correspond à la seule partie du candidat dont la description par le nuage de points est satisfaisante. De la même manière que pour le calcul des limites des primitives, l'algorithme d'évaluation de la couverture de  $\mathcal{S}''$  repose sur la construction d'une grille régulière à sa surface, dont chaque cellule carrée est de largeur  $\zeta$ . L'ensemble  $\mathcal{C}_{(i,j)}^- \subset \mathcal{P}$  des points de contribution  $h_D$  négative se projetant sur la cellule de coordonnées  $(i, j)$  dans cette grille est donné par les équations 4.26, 4.24 et 4.4 pour le cas du plan. Pour le cylindre, les équations 4.26 et 4.24 définissant le contenu de la cellule restent identiques, et il suffit de modifier l'équation 4.4 définissant les coordonnées surfaciques d'un point  $\mathbf{p}$  comme suit:

$$\begin{aligned} w(\mathcal{S}, \mathbf{p}) &= \text{atan2}((\mathbf{x}_S \times \mathbf{a}_S) \cdot (\mathbf{p} - \mathbf{c}_S), \mathbf{x}_S \cdot (\mathbf{p} - \mathbf{c}_S)) \\ l(\mathcal{S}, \mathbf{p}) &= \mathbf{a}_S \cdot (\mathbf{p} - \mathbf{c}_S) \end{aligned}$$

$\mathbf{x}_S$  étant un vecteur unitaire arbitraire orthogonal au vecteur directeur  $\mathbf{a}_S$  de l'axe du cylindre  $\mathcal{S}$ . Rappelons que  $\mathbf{c}_S$  est le centre du cylindre (en l'occurrence, de la portion effective du cylindre). La fonction trigonométrique  $\text{atan2}(y, x) : \mathbb{R}^2 \rightarrow ]-\pi; \pi]$  quant à elle retourne l'angle entre les vecteurs  $(x, y)$  et  $(1, 0)$  du plan, autrement dit l'argument principal du complexe  $x + iy$ .

Soit  $N$  le nombre de cellules recouvrant la surface de  $\mathcal{S}''$ , et si  $M$  désigne le nombre de cellules dont le contenu  $\mathcal{C}_{(i,j)}^-$  est non vide, on a alors:

$$\Gamma(\mathcal{S}, \mathcal{P}) = \frac{M}{N} \tag{4.31}$$

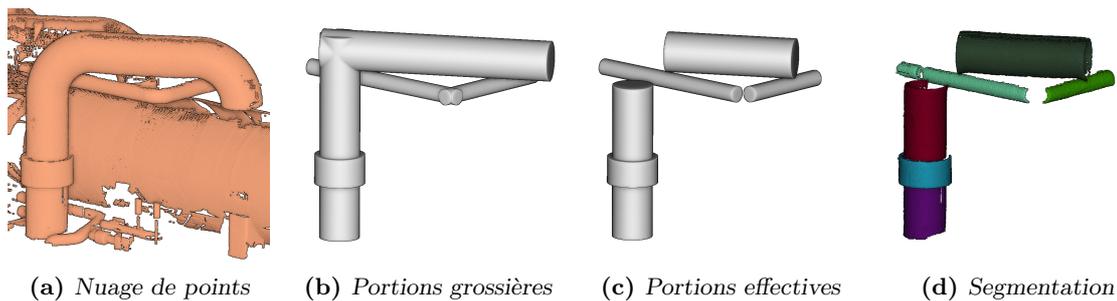


**Figure 4.9:** Estimation de la couverture de la surface d'une primitive. Cette approximation se base sur la construction d'une grille à la surface de l'objet, de manière à comptabiliser la proportion de cellules à l'intérieur desquelles se projette au moins un point apportant une contribution  $h_D$  négative vis-à-vis de l'objet.

## 4.5 Résultats

### 4.5.1 Calcul des limites de cylindres

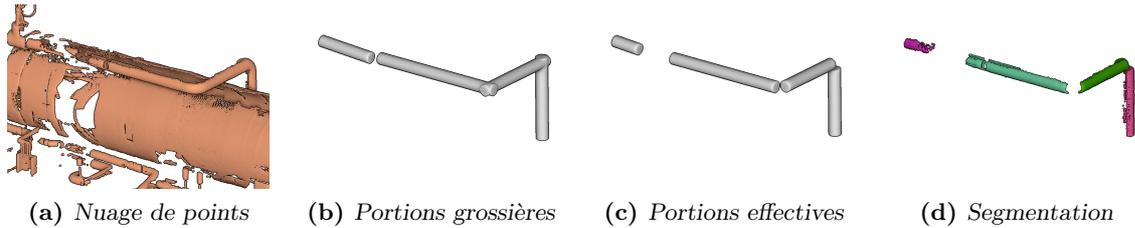
Le calcul des portions grossières permet d'obtenir des objets entrant en collision au niveau de ce qui devrait être leur jonction (figure 4.10b). Ceci suppose notamment que les objets sont suffisamment échantillonnés au niveau de leurs extrémités, pour permettre la surévaluation des portions. Sans quoi les limites des primitives sont fixées à l'endroit du nuage de points où les données sont absentes.



**Figure 4.10:** Résultat du calcul de limite de cylindres dans un nuage de points industriel.

L'absence de données peut en effet occasionner le découpage d'un même objet de la scène en plusieurs primitives distinctes (cf. figure 4.11). En effet, faute de points, l'algorithme d'estimation des portions estime que l'objet n'est plus représenté au delà d'une certaine extrémité, et ne peut deviner que la même primitive se prolonge pourtant après ce "trou" dans les données. La détection et la réparation de ce genre de phénomène au niveau de la génération d'un candidat est un problème relativement complexe, du fait notamment des tailles variées des trous pouvant survenir parmi les données. Il est donc préférable de laisser à l'utilisateur le soin d'identifier et de corriger ces situations au besoin, en guise de post-traitement des résultats (à l'aide d'un éventuel algorithme d'identification de tuyaux alignés).

On constate que la qualité de l'estimation des portions effectives de cylindre est très satisfaisante, en particulier au niveau des coudes joignant deux cylindres connectés en "L" (figures 4.10c et 4.10d). Les primitives sont en effet limitées aux parties cylindriques rectilignes des lignes de tuyauterie, et ne débordent que très peu sur la partie torique correspondant au coude. Ces résultats sont particulièrement intéressants pour les cylindres, puisque les limites effectives des cylindres servent notamment à l'initialisation du



**Figure 4.11:** Influence des données sur l'évaluation des portions de primitives. L'absence de données donne lieu à la séparation du tuyau de gauche en deux parties distinctes.

calcul des tores joignant deux cylindres connectés, en définissant les limites de la zone de récupération des points échantillonnant le coude (cf. section 3.4.1). En pratique, la nuance entre les portions effectives et les portions grossières de plans est négligeable.

#### 4.5.2 Adéquation vis-à-vis des données et de l'*a priori*

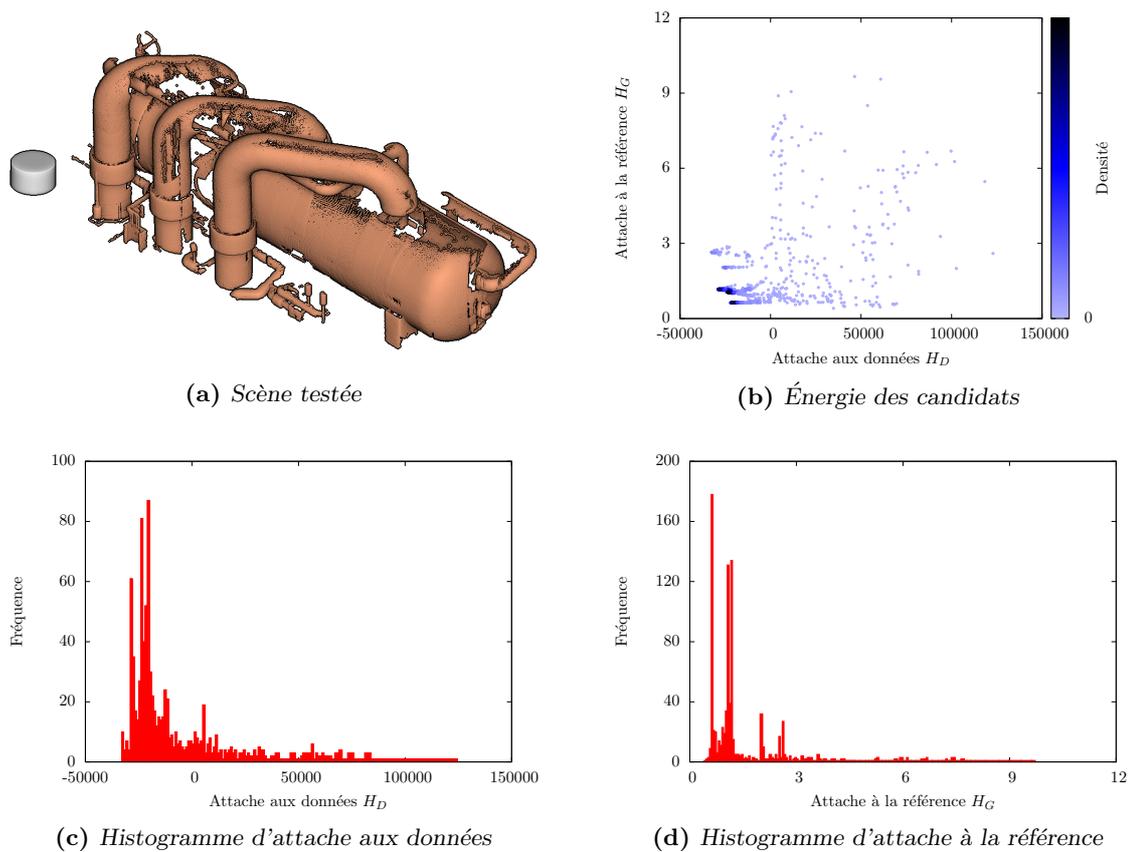
Pour étudier le comportement du générateur de primitives proposé, nous présentons quelques résultats obtenus sur des données réelles (figures 4.12 à 4.18). Étant donnée une primitive de référence  $\mathcal{S}^i$  issue de l'*a priori*, nous nous proposons ici d'étudier la répartition statistique des primitives dans l'espace des énergies  $H_D$  et  $H_G$ , pour vérifier que les objets générés ont effectivement tendance à minimiser ces deux quantités. Nous n'utilisons pas ici le biais de sélection de graine proposé en section 4.2.4, puisque l'assemblage des candidats n'est pas géré ici (on ne conserve donc pas de segmentation du nuage de points nécessaire à la mise en place du biais).

La première figure 4.12 présente un premier exemple relativement favorable, dans la mesure où la primitive de référence fait directement face aux objets qui lui correspondent le plus dans la scène. Il existe toutefois un décalage significatif de  $\mathcal{S}^i$  vis-à-vis de des objets de  $\mathcal{P}$  lui correspondant, ainsi qu'un léger désaxage. Le référent  $\mathcal{S}^i$  est d'ailleurs construit après translation et réorientation d'un objet préalablement reconnu dans  $\mathcal{P}$ , sur cet exemple ainsi que sur ceux présentés par la suite.

En observant les divers graphiques de répartition des énergies  $H_D$  et  $H_G$  associés à cette expérience, on constate qu'une part significative des candidats générés à partir de  $\mathcal{S}^i$  présentent simultanément une attache aux données  $H_D$  et une ressemblance à la référence  $H_G$  satisfaisante. Ceci est particulièrement visible sur la figure 4.14b où l'on constate une grande concentration de points dans la partie inférieure gauche du graphique correspondant aux énergies les plus faibles. On remarque notamment la présence de quelques bassins d'attraction des candidats dans cette zone, dont on peut penser qu'ils correspondent très probablement aux quelques objets de  $\mathcal{P}$  ressemblant à  $\mathcal{S}$  (les 2 tuyaux verticaux de gauche). Cette supposition est confirmée par l'étude visuelle des candidats générés (cf. figure 4.13). On constate effectivement que les objets les plus plébiscités sont relativement semblables à la référence  $\mathcal{S}^i$ .

La figure 4.14 quant à elle présente une situation légèrement plus complexe. La référence  $\mathcal{S}^i$  est choisie et positionnée dans cette expérience de manière à ce qu'il n'existe par d'objet de  $\mathcal{P}$  correspondant à  $\mathcal{S}$  dans son voisinage immédiat. Il existe en effet des "obstacles" entre  $\mathcal{S}^i$  et ses correspondances, et on attend de l'algorithme qu'il parvienne à les contourner pour trouver effectivement les objets du nuage ressemblant à  $\mathcal{S}^i$ .

Sans surprise, on constate que l'algorithme est clairement distrait par ces obstacles, et

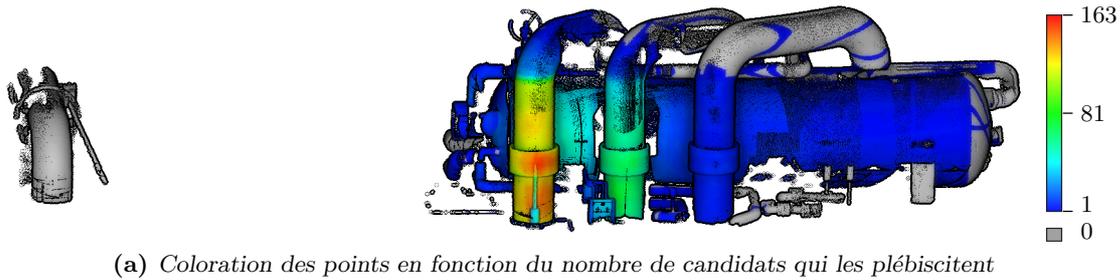


**Figure 4.12:** Répartition statistique des énergies  $H_D$  et  $H_G$  de 1000 candidats générés à partir d'un référent  $\mathcal{S}^i$  (en gris). On utilise ici les tolérances  $\sigma_p = 2m$ ,  $\sigma_o = 15^\circ$  et  $\sigma_g = 0.2 \times r_{\mathcal{S}^i}$ , où  $r_{\mathcal{S}^i}$  désigne le rayon du référent  $\mathcal{S}^i$ . On supprime ici les candidats dont l'énergie géométrique  $H_G$  vis-à-vis de  $\mathcal{S}^i$  dépasse 10, pour permettre la lecture des résultats (il reste alors 931 candidats sur les 1000 générés).

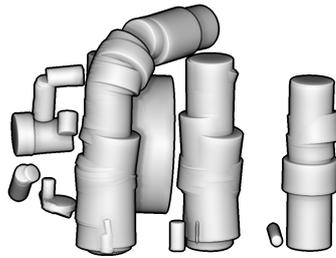
ne parvient pas à cibler aussi bien les primitives recherchées. Les énergies des candidats sont dispersées, et il en est donc de même des candidats. On note toutefois la présence, dans le coin inférieur gauche du diagramme 4.16b de répartition des deux énergies, d'une concentration élevée et très localisée de candidats. De plus, on dénombre 30 candidats générés sur le tuyau que nous visons à reconstruire dans cet exemple (tuyau cheminant sous les 3 conduites verticales en figure 4.15), de qualité variée en matière d'attache aux données.

L'expérience illustrée par la figure 4.16 considère un plan de référence  $\mathcal{S}^i$  au lieu d'un cylindre.

L'analyse de ces résultats est plus complexe, dans la mesure où la taille des plans est calculée de telle sorte que les candidats ne sont considérés que sur la section où ils sont bien définis par le nuage de points. Autrement dit, les plans candidats générés présentent en général une énergie  $H_D$  d'attache aux données très basse, puisqu'on ne conserve que la partie permettant justement de minimiser  $H_D$ . Ce constat ne s'applique pas vraiment aux cylindres. Ces derniers sont en effet eux aussi bornés en étudiant la section la mieux décrite par le nuage de points le long de leur axe. Mais l'intégralité du contour circulaire des cylindres est conservée. Un cylindre ne s'adaptant pas aux données pourra donc se voir pénalisé malgré tout, contrairement au plan qui peut être contracté à volonté. Les



(a) Coloration des points en fonction du nombre de candidats qui les plébiscitent



(b) Visualisation des candidats pertinents vis-à-vis des données (552 cylindres)

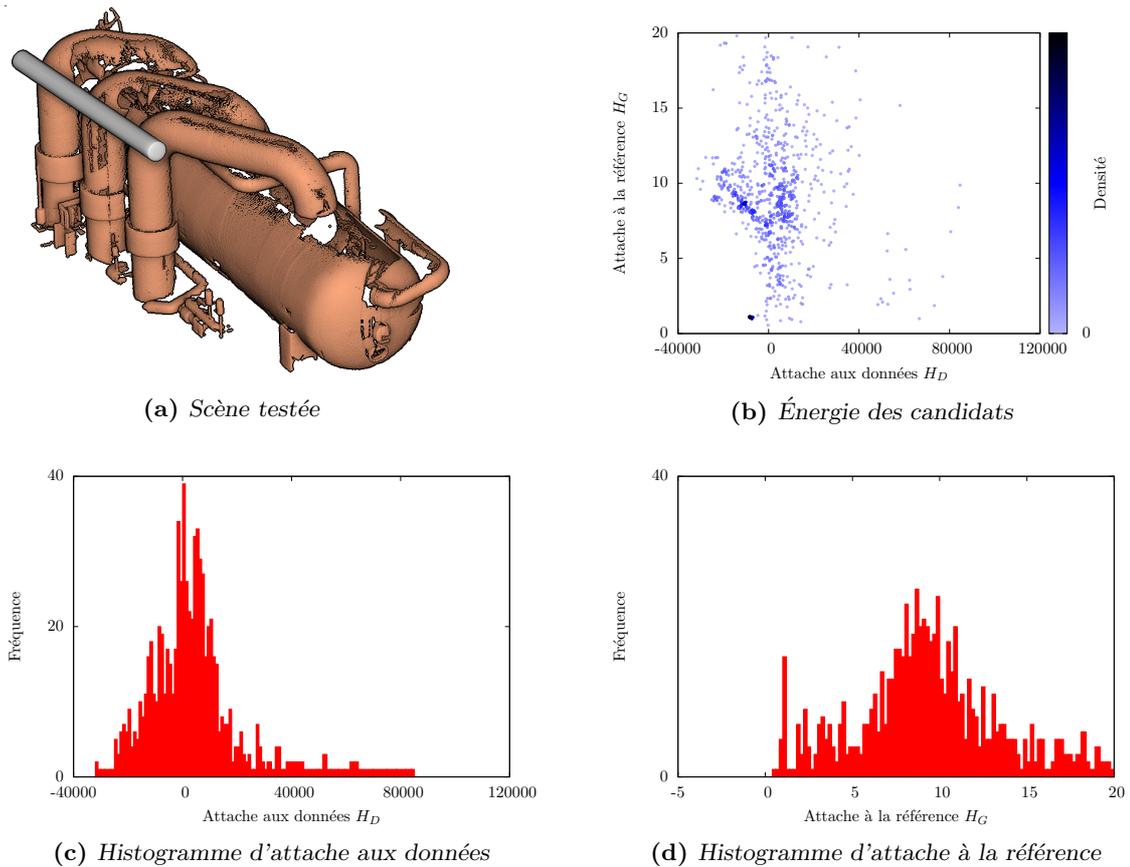
**Figure 4.13:** Visualisation des candidats générés à partir du premier exemple (figure 4.12). Le premier schéma présente la coloration de chaque point en fonction du nombre de candidats qu'il supporte, c'est à dire vis-à-vis desquels la contribution énergétique d'attache aux données  $h_D$  est négative. Le second schéma quant à lui illustre les candidats dont l'énergie d'attache aux données  $H_D$  est négative.

diagrammes de la figure 4.16 confirment cette adaptabilité des plans vis-à-vis des données, de par la quantité des candidats dont l'énergie  $H_D$  est négative.

Cependant, l'historgramme 4.18d de répartition des énergies géométriques  $H_G$  vis-à-vis de  $\mathcal{S}^i$  montre clairement que l'algorithme favorise l'apparition des plans semblables à  $\mathcal{S}$  au sens de  $H_G$ . Ce constat est confirmé par l'étude visuelle des candidats générés (figure 4.17). Ces résultats montrent notamment la limite de la méthode, du fait de la non prise en compte de la géométrie du nuage de points. En effet, le générateur explore les surfaces indépendamment de leur nature, qu'elles soient planaires ou cylindriques. Les flancs de la cuve faisant face au plan sont par exemple souvent choisis pour la création des primitives planaires, puisqu'une partie des points de cette cuve présentent une position et une orientation de la normale les rendant pertinents lors de l'étape du choix de la graine. Il aurait été souhaitable d'utiliser ici une information de géométrie locale du nuage de points pour renforcer la pertinence du choix de la graine, et éviter ainsi les erreurs constatées sur cet exemple.

Le dernier exemple présenté 4.18 correspond à une scène plus complexe que la première. L'échantillonnage des objets y est plus pauvre, puisque le recouvrement des surfaces est incomplet bien que la densité d'échantillonnage et la précision soient plus importantes. Qui plus est, la scène est plus encombrée aux alentours de l'objet. On utilise en outre une tolérance aux déplacements relativement permissive, et le désaxage de  $\mathcal{S}^i$  est considérable.

L'algorithme fait plus difficilement ressortir des tendances parmi les candidats générés sur cet exemple. Les énergies d'attache aux données sont en effet très dispersées. On constate cependant, cette fois encore, que le générateur tend à privilégier l'apparition d'objets présentant une énergie géométrique faible. La difficulté majeure, dans cette scène, tient au fait que le générateur doit compenser le désaxage considérable du cylindre vis-à-vis



**Figure 4.14:** On répète l'expérience illustrée en figure 4.12, en utilisant cette fois-ci un cylindre  $\mathcal{S}^i$  positionné de manière à ce que les points du nuage dont il est le plus proche ne lui correspondent pas. Cependant, il existe une partie du nuage qui lui correspond effectivement, au delà des surfaces les plus proches (tuyau moyen cheminant en dessous des trois plus gros). Après filtrage des candidats dont l'énergie  $H_G$  excède 20, on obtient 740 candidats sur 1000.

des tuyaux de la scène. Le générateur expérimente ici très largement les murs et le sol, qui présentent de nombreux points dont l'orientation vis-à-vis de la référence est relativement acceptable, alors que les tuyaux sont quant à eux plus complexes à atteindre. Malgré tout, en observant la figure 4.19, on constate que la méthode parvient à générer quelques centaines de candidats que l'on peut juger comme étant pertinents étant données les tolérances aux changements très permissives que l'on utilise ici ( $\sigma_p = 3m$  et  $\sigma_o = 30^\circ$ ).

De manière générale, ces résultats confirment la capacité de l'algorithme de génération de primitive guidé par l'*a priori* à favoriser l'apparition d'objets cohérents vis-à-vis de la référence  $\mathcal{S}^i$  choisie. Sa faculté à retrouver effectivement des objets du nuage dépend en revanche de la difficulté de la situation traitée. Sur les exemples favorables, où les différences entre  $\mathcal{S}^i$  et  $\mathcal{P}$  sont raisonnables, les primitives générées s'appuient effectivement sur le nuage de points. Toutefois, lorsque les objets recherchés sont trop éloignés de  $\mathcal{S}^i$  au sens de  $H_G$ , l'algorithme peine à retrouver ces objets noyés parmi l'ensemble potentiellement immense des données.

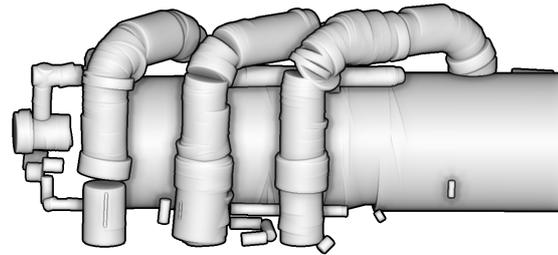
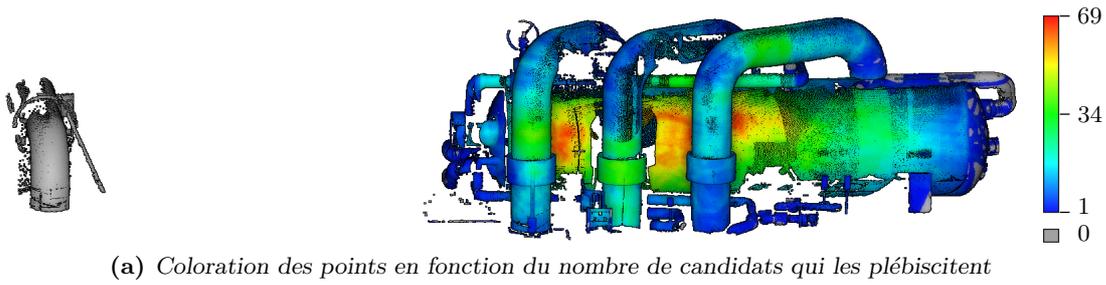


Figure 4.15: Visualisation des candidats générés à partir du second exemple (figure 4.14).

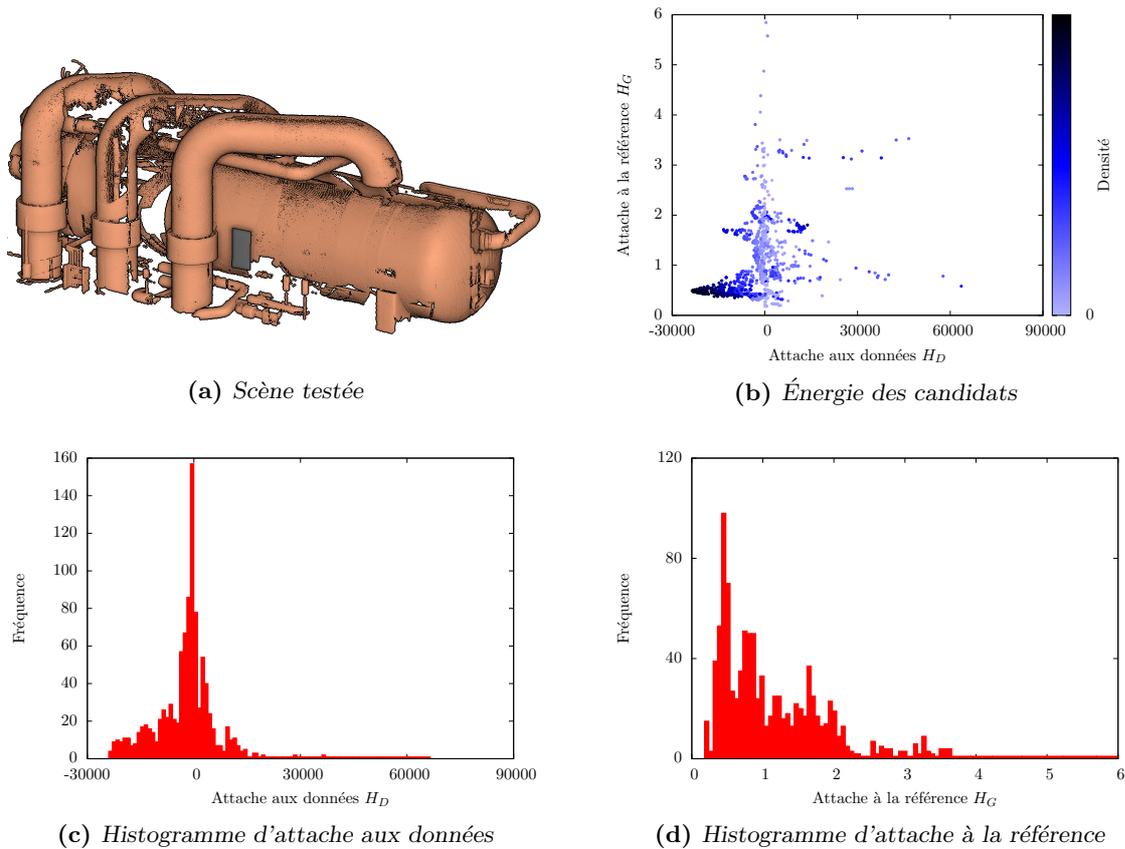
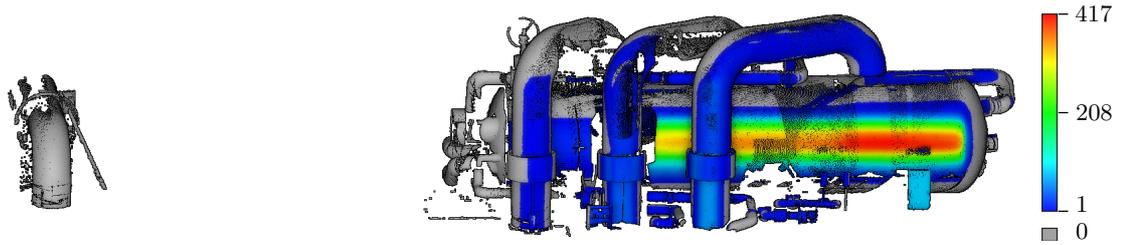
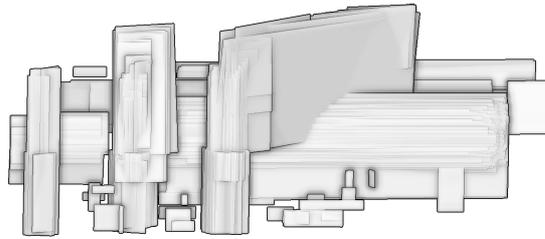


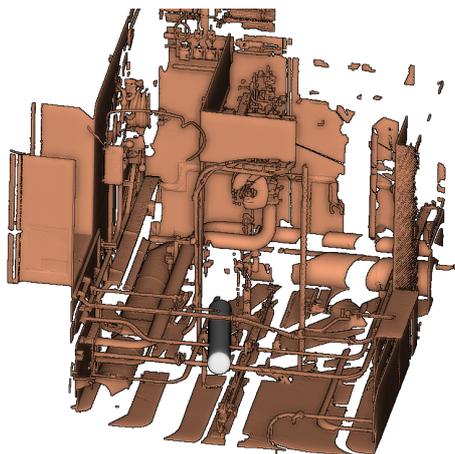
Figure 4.16: La référence  $S^i$  est cette fois-ci un patch rectangulaire (en gris au milieu de la figure représentant la scène). On utilise ici une tolérance au déplacement plus élevée  $H_G = 5m$ , du fait du positionnement de l'objet, et n'appliquons aucun filtrage sur les candidats générés.



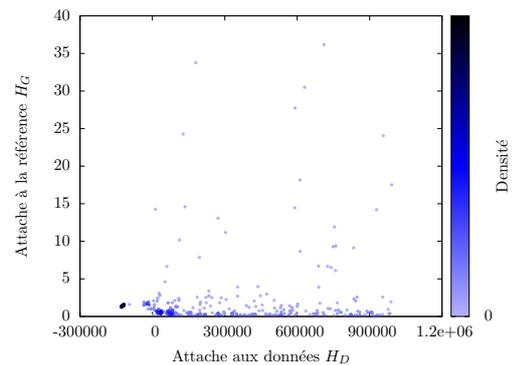
(a) Coloration des points en fonction du nombre de candidats qui les plébiscitent



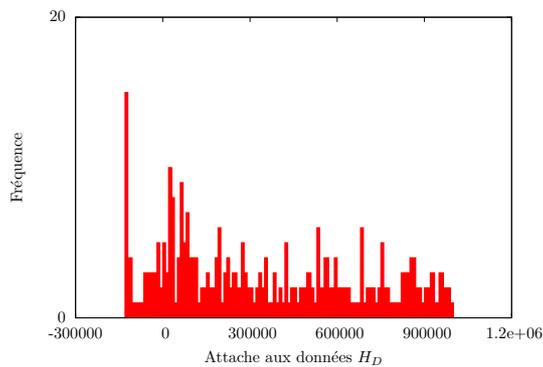
(b) Visualisation de la totalité des candidats générés (1000 plans)

**Figure 4.17:** Visualisation des candidats générés à partir du troisième exemple (figure 4.16).

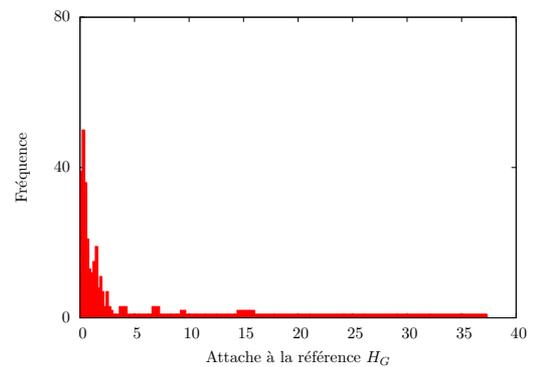
(a) Scène testée



(b) Énergie des candidats

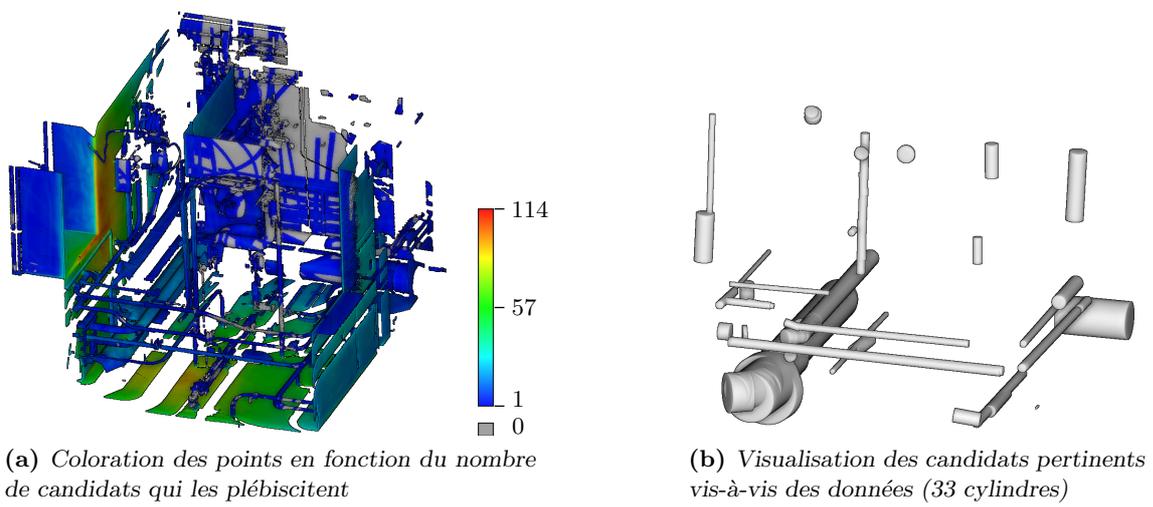


(c) Histogramme d'attache aux données



(d) Histogramme d'attache à la référence

**Figure 4.18:** Expérimentation dans une scène complexe. En filtrant les cylindres générés tels que  $H_D < 10^{-6}$  et  $H_G < 40$ , il reste 268 candidats représentés dans ces graphiques. On utilise ici un seuil  $\sigma_p = 3m$ , soit à peu près la taille de la scène complète, et un seuil  $\sigma_o = 30^\circ$ . Le cylindre de référence  $S^i$  figure au centre de la scène (en gris).



**Figure 4.19:** Visualisation des candidats générés à partir du quatrième exemple (figure 4.18).

# Conclusion

## Bilan

En nous basant sur le formalisme Bayésien d'ores et déjà utilisé dans de multiples travaux, nous avons proposé une formulation probabiliste du problème de reconstruction intégrant, en plus du traditionnel critère d'ajustement aux données du nuage, un certain nombre d'*a priori* quant à la composition géométrique et la structuration du modèle CAO reconstruit. Cette démarche permet en particulier de tirer parti d'un modèle CAO disponible supposé relativement semblable à la solution recherchée. Le résultat calculé de manière à satisfaire l'ensemble des différents critères requis offre des garanties de qualité et de fiabilité que l'on ne retrouve pas dans les approches habituelles d'extraction de formes telles que l'algorithme RANSAC ou les transformées de Hough par exemple ne visant que l'adéquation des surfaces calculées aux données observées.

Pour permettre la recherche du modèle CAO qui optimise la probabilité cible, nous avons proposé une heuristique gloutonne d'assemblage de primitives candidates générées selon un processus stochastique. Cette heuristique n'offre aucune garantie théorique de convergence vers l'optimum global de la fonction de probabilité cible, contrairement à certains algorithmes existants eux aussi basés sur l'exploration stochastique de l'espace des solutions. Elle semble toutefois en mesure de traiter convenablement le problème posé en améliorant la qualité de la solution au fil du processus de reconstruction. Bien que rien ne permette d'affirmer que la solution calculée soit effectivement la meilleure, le modèle CAO reconstruit par cette heuristique s'ajuste malgré tout très bien sur le nuage de points, semble conforme aux contraintes d'assemblage des primitives, et ressemble au modèle CAO *a priori* spécifié pour la résolution du problème.

La vitesse de convergence de l'heuristique de construction repose avant tout sur la capacité du générateur de primitives stochastique à proposer des éléments pertinents participant effectivement à l'amélioration de la solution en cours de reconstruction. Pour permettre la reconstruction d'un modèle CAO "en temps raisonnable" (de l'ordre de quelques jours CPU au plus), il est indispensable de faire en sorte que les primitives géométriques impliquées dans la solution recherchée apparaissent au plus tôt dans la résolution. Nous avons pour cela proposé un processus biaisé de génération de primitives, basé sur la création de formes guidée par le nuage de points et le modèle CAO *a priori*. Cette méthode permet, par construction, de favoriser l'apparition des formes effectivement échantillonnées par le nuage de points, et proches du modèle CAO *a priori*.

Cette méthode permet de ne traiter, parmi l'ensemble des données, que la reconstruction des parties du nuage de points correspondant effectivement au modèle CAO *a priori* que l'on se donne. En effet, nous estimons que les seules reconstructions acceptables sont celles ressemblant aux données, de manière à pouvoir garantir la fiabilité du résultat. L'évaluation de la ressemblance peut toutefois être modulée à l'aide de paramètres spécifiant les tolérances aux changements attendus.

L'approche proposée s'appuie sur une grande quantité de paramètres réglables par l'utilisateur. Nous nous sommes toutefois attachés à ce que la plupart d'entre eux puissent être appréhendés simplement par l'utilisateur, dans la mesure du possible. En effet, les grandeurs inhérentes au traitement du problème que l'on souhaite aborder (en l'occurrence, les grandeurs géométriques telles que les distances et angles) peuvent effectivement avoir un sens évident au regard de l'opérateur souhaitant utiliser la méthode. Ce constat ne peut raisonnablement pas se généraliser et certains algorithmes nécessitent par exemple une initialisation à partir de valeurs dont l'estimation est très complexe (par exemple le recuit simulé). Dans nos travaux, la majorité des variables intervenant dans le processus de reconstruction traitent de distances, angles ou ratio et ces valeurs peuvent être évaluées facilement grâce à leur caractère intuitif, ou grâce à l'évaluation visuelle de la scène à traiter. Nous proposons de plus une approche permettant d'automatiser l'estimation des trois paramètres  $\lambda_D$ ,  $\lambda_G$  et  $\lambda_T$  dont on pense que la détermination par l'utilisateur est impossible.

## Perspectives

Finalement, cette thèse apporte une formulation du problème de reconstruction 3D à partir d'*a priori* ainsi que des outils permettant de résoudre ce problème. Mais certains aspects pourraient être encore approfondis.

Nous nous sommes par exemple essentiellement intéressés aux plans, cylindres et tores selon le rôle qu'il occupent dans les scènes industrielles. Il peut toutefois être envisageable d'étendre la méthode à d'autres formes canoniques pouvant être trouvées dans les centrales. Nous pensons en particulier aux poutrelles métalliques très nombreuses dans les scènes industrielles, et décrites par des formes issues de catalogues. Nous avons en effet considéré en première estimation que ces objets, couramment décrits par des profils extrudés, pouvaient être représentés par des assemblages de plans (cf. annexe A). Mais la prise en compte d'une nouvelle géométrie spécifique pour ces composants semble cependant plus appropriée. La méthode d'assemblage des formes étant d'ores et déjà générique, puisqu'indépendante de la nature des objets manipulés, on pourrait alors étendre les capacités du générateur de primitives pour lui permettre d'instancier de tels objets à partir de points du nuage, et intégrer aux énergies topologique  $H_T$  et géométrique  $H_G$  la prise en compte de ce nouveau type de primitive. Cette remarque vaut pour tout nouveau type d'objet. Il faut toutefois prendre garde à la combinatoire induite par la gestion des interactions entre objets dans l'énergie topologique  $H_T$ , et faire en sorte que le nombre de types de primitives manipulés reste tout de même raisonnable.

De plus, la méthode proposée souffre d'une lenteur qui pourrait certainement être atténuée.

Le principal handicap rencontré par l'approche est la multiplication des traitements au niveau des points du nuage. Le fait de diminuer la quantité des points permettrait ainsi d'accélérer significativement le processus de reconstruction. Il s'agit toutefois de ne pas

écarter des données que l'on souhaiterait traiter par la suite. Un des premiers traitements envisageable serait alors l'isolement des seules données qui nous intéressent réellement dans le problème, à savoir les parties du nuage qui diffèrent de l'*a priori*. Nous avons en effet abordé ce sujet au chapitre 2, en proposant d'écarter préalablement les données liées au génie civil. Cette remarque peut s'étendre à l'ensemble des données communes au nuage de point et au modèle CAO *a priori*: il n'est pas nécessaire de chercher à reconstruire des formes qui sont initialement représentées parfaitement par le modèle CAO *a priori*. Une mise en correspondance préalable entre les points du nuage et les surfaces du modèles CAO *a priori*, sur la base des travaux de D. Girardeau-Montaut [GM06], pourrait ainsi permettre de ne conserver que les parties distinctes entre les deux scènes, et ainsi simplifier considérablement le problème.

Une autre source de complexité de l'algorithme est la détection des connexions entre les formes. Chaque nouveau candidat à l'insertion dans le modèle CAO doit en effet être confronté à l'ensemble des formes déjà reconnues pour identifier les collisions. Cette approche naïve peut être considérablement accélérée en ayant recours à des structures appropriées telles que les hiérarchies de volumes englobants (*Bounding Volume Hierarchy*), de manière à limiter le nombre de tests effectués pour la recherche des collisions. Il faut toutefois prendre garde à ce que ces structures puissent être maintenues à moindre effort lors des multiples ajouts et suppressions de formes réalisés par l'algorithme.

Le rendement du générateur de primitives quant à lui pourrait aussi être significativement amélioré, en tenant compte de la géométrie locale du nuage de points dans le choix de la graine. En effet, le premier point servant à la construction de candidat est uniquement sélectionné sur la base de critères de position et d'orientation, mais la géométrie est ignorée faute de descripteurs fiables permettant de guider ce choix. En tenant compte de ce troisième critère, on pourrait alors améliorer la pertinence de la sélection de graine, et donc la pertinence des formes générées. Typiquement, pour la génération de plans, on souhaiterait avantager les points dont le voisinage semble effectivement planaire. De même que pour la génération de cylindres, il serait souhaitable de tenir compte de la courbure par exemple dans le choix de la graine (pour générer un petit cylindre, privilégier les points associés aux plus fortes courbures). Or cela suppose que l'on sait estimer de manière fiable un descripteur géométrique dans le nuage de points, ce qui semble relativement complexe en pratique sur les nuages de points qui nous intéressent ici.

En outre, la génération et l'évaluation des candidats sont des procédures qui se prêtent parfaitement à la parallélisation. Sur les ordinateurs actuels possédant pour la plupart plus de 4 cœurs (et les unités comptant quelques centaines de cœur, comme les cartes graphiques par exemple), on pourrait ainsi bénéficier d'un facteur d'accélération intéressant. La gestion de l'assemblage des candidats constitue toutefois un goulot d'étranglement puisque cette étape est difficilement parallélisable. Cependant ce traitement est significativement plus rapide que la génération et l'évaluation de la qualité des primitives proposées, et ne devrait *a priori* pas être gênant.

La seconde amélioration que nous avons envisagée, sans avoir pu l'étudier en détail, consisterait en l'apprentissage des règles métier de connexion entre les formes à partir du modèle CAO disponible. Nous avons en effet postulé une certaine quantité de règles d'assemblage à partir de constatations génériques quant à la composition des scènes. Ces règles peuvent s'avérer contraignantes et pénaliser des configurations qui, puisqu'elles ne sont pas prévues parmi les possibilités envisagées *a priori*, se trouvent exclues du modèle CAO reconstruit malgré le fait qu'elles puissent être pertinentes. Or le modèle CAO disponible *a priori* peut lui-même être représentatif de l'ensemble des connexions que l'on

peut attendre dans une scène industrielle. Ainsi, au lieu d'utiliser une énergie topologique fixée à l'avance, cette énergie pourrait être forgée à partir des connexions disponibles dans le modèle CAO *a priori*. On pourrait typiquement s'appuyer sur une formulation à base de Gaussiennes comme nous l'avons fait, en positionnant les centres et les amplitudes de manière à favoriser les configurations apparaissant effectivement dans le modèle CAO *a priori*.

Pour améliorer la complétude des résultats (la propension à détecter l'ensemble des formes qui devraient figurer dans la solution optimale), il peut être intéressant de séparer le traitement des formes de l'*a priori* selon leur échelle notamment. Nous avons en effet constaté que les plus petits éléments sont difficilement détectés parmi les grands ensembles de données. Le traitement préalable des plus gros équipements, puis la suppression des données reconnues dans le nuage de points pourraient ainsi permettre de faciliter la recherche des plus petits éléments dans un second temps.

Une des dernières pistes que nous aurions souhaité explorer s'articule autour de la prise en compte de la variabilité du bruit et de la densité des points du nuage. En effet, comme la plupart des travaux d'analyse de nuages de points, nous utilisons des paramètres uniques s'appliquant à la totalité du nuage. Or, comme nous le mentionnons au chapitre 1, le bruit et la densité sont deux grandeurs extrêmement variables sur l'ensemble des données. Et la prise en compte plus fine de ces caractéristiques permettrait certainement de renforcer la pertinence des traitements proposés, en particulier pour la détection des plus petits éléments dont la taille coïncide quasiment avec le seuil de bruit  $\epsilon$ , ou la taille des cellules  $\zeta$ .

# Primitives géométriques pour la construction CAO

## Sommaire

---

<b>A.1 Plans</b> . . . . .	<b>175</b>
A.1.1 Cas du plan infini . . . . .	175
A.1.2 Outils géométriques . . . . .	175
<b>A.2 Cylindres</b> . . . . .	<b>176</b>
A.2.1 Cas du cylindre infini . . . . .	177
A.2.2 Outils géométriques . . . . .	177
<b>A.3 Tores à section circulaire</b> . . . . .	<b>178</b>
A.3.1 Outils géométriques . . . . .	178
<b>A.4 Cônes</b> . . . . .	<b>180</b>
<b>A.5 Sphéroïde</b> . . . . .	<b>180</b>
<b>A.6 Divers</b> . . . . .	<b>181</b>

---

---

### Préambule

---

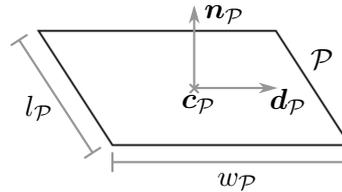
*Dans les modèles CAO de scène industrielles, la plupart des éléments sont communément décrits à l'aide de plans, cylindres, cônes, tores (circulaires ou rectangulaires), sphéroïdes, et surfaces d'extrusion. Cette annexe détaille les primitives géométriques habituellement utilisées dans les modèles CAO d'installations industrielles, en explicitant une paramétrisation de chacune et détaillant quelques outils pour le calcul s'appliquant aux formes auxquelles nous nous intéressons plus particulièrement dans cette thèse.*

---

## A.1 Plans

La plupart des surfaces planes représentent le bâti (sols, murs, plafonds, ...). Toutefois, certains boîtiers ou conduits d'aération peuvent aussi être représentés par des assemblages de plans orthogonaux.

Afin de faciliter la représentation et la manipulation de ces objets, nous choisissons de délimiter les surfaces planaires par des bords rectangulaires. Bien que cette description ne soit pas toujours parfaitement adaptée, elle permet de représenter les plans sans avoir à considérer une représentation complexe des bordures. Une fois intégré dans un modèle CAO complet, il est envisageable d'adopter une représentation plus complexe du plan pour laquelle les bords seraient définis à partir des interactions entre chaque plan et son environnement (bords circulaires, lignes brisées, etc. en fonction de la situation).



Patch planaire rectangulaire	
Paramètre	Description
$\mathbf{n}_P$	Vecteur unitaire normal au plan $\mathcal{P}$ .
$\mathbf{c}_P$	Position de $\mathcal{P}$ . Si le plan est borné (largeur et longueur finies), on prendra le centre de masse. Dans le cas contraire, tout point du plan peut être utilisé, et la position du plan peut alors être décrite par sa seule distance à l'origine (nombre réel).
$\mathbf{d}_P$	Vecteur unitaire orthogonal à la normale $\mathbf{n}_P$ décrivant l'orientation de $\mathcal{P}$ autour de $\mathbf{n}_P$ lorsque le plan est borné. Si le plan est infini, ce paramètre n'a aucune utilité.
$w_P$	Largeur de $\mathcal{P}$ .
$l_P$	Longueur de $\mathcal{P}$ .

### A.1.1 Cas du plan infini

Le plan infini se caractérise par son absence de limites. On considère alors que  $w_P = \infty$  et  $l_P = \infty$ . Dans ces conditions, tout point appartenant au plan suffit à définir sa position  $\mathbf{c}_P$ , et l'orientation  $\mathbf{d}_P$  n'a plus lieu d'être. Finalement, au lieu des 8 paramètres libres nécessaires à la description d'un patch rectangulaire plan, le plan infini peut être décrit par les 3 paramètres libres définissant son équation: deux paramètres angulaires définissant sa normale unitaire  $\mathbf{n}_P$  et un paramètre définissant sa distance à l'origine correspondant à sa position  $\mathbf{c}_P$  (qui est alors le point du plan le plus proche de l'origine).

### A.1.2 Outils géométriques

**Distance** La distance séparant un point  $\mathbf{p}$  d'un plan  $\mathcal{P}$  infini est la suivante:

$$d(\mathbf{p}, \mathcal{P}) = |(\mathbf{p} - \mathbf{c}_P) \cdot \mathbf{n}_P| \quad (\text{A.1})$$

Lorsque le plan  $\mathcal{P}$  est borné, l'expression de la distance est alors plus complexe:

$$d(\mathbf{p}, \mathcal{P}) = \sqrt{d_Z(\mathbf{p}, \mathcal{P})^2 + \max(0, |d_X(\mathbf{p}, \mathcal{P}) - w_P|)^2 + \max(0, |d_Y(\mathbf{p}, \mathcal{P}) - l_P|)^2} \quad (\text{A.2})$$

avec:

$$\begin{aligned} d_Z(\mathbf{p}, \mathcal{P}) &= \mathbf{n}_{\mathcal{P}} \cdot (\mathbf{p} - \mathbf{c}_{\mathcal{P}}) \\ d_X(\mathbf{p}, \mathcal{P}) &= \mathbf{d}_{\mathcal{P}} \cdot (\mathbf{p} - \mathbf{c}_{\mathcal{P}}) \\ d_Y(\mathbf{p}, \mathcal{P}) &= (\mathbf{d}_{\mathcal{P}} \times \mathbf{n}_{\mathcal{P}}) \cdot (\mathbf{p} - \mathbf{c}_{\mathcal{P}}) \end{aligned}$$

**Vecteur normal** Le vecteur normal au plan  $\mathcal{P}$  en tout point  $\mathbf{p}$  du plan (à l'exception des bords où elle n'est pas définie) est par définition:

$$\mathbf{n}_{\mathcal{P}}(\mathbf{p}) = \mathbf{n}_{\mathcal{P}} \quad (\text{A.3})$$

**Projection** La projection d'un point  $\mathbf{p}$  sur un plan  $\mathcal{P}$  infini peut s'exprimer comme suit:

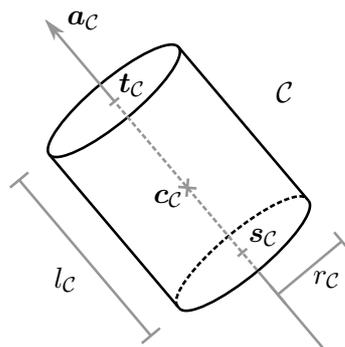
$$\mathbf{p}_{\mathcal{P}}^{\downarrow} = \mathbf{p} - \mathbf{n}_{\mathcal{P}} ((\mathbf{p} - \mathbf{c}_{\mathcal{P}}) \cdot \mathbf{n}_{\mathcal{P}}) \quad (\text{A.4})$$

Lorsque le plan  $\mathcal{P}$  est borné, la projection se définit alors comme suit:

$$\mathbf{p}_{\mathcal{P}}^{\downarrow} = \mathbf{c}_{\mathcal{P}} + d_Z(\mathbf{p}, \mathcal{P}) \cdot \mathbf{n}_{\mathcal{P}} + \min(d_X(\mathbf{p}, \mathcal{P}), w_{\mathcal{P}}) \cdot \mathbf{d}_{\mathcal{P}} + \min(d_Y(\mathbf{p}, \mathcal{P}), l_{\mathcal{P}}) \cdot (\mathbf{d}_{\mathcal{P}} \times \mathbf{n}_{\mathcal{P}}) \quad (\text{A.5})$$

## A.2 Cylindres

Les cylindres sont massivement présents dans les scènes industrielles, et décrivent principalement les composants rectilignes de tuyauterie. Les cylindres peuvent aussi représenter certains équipements majeurs (GSS, générateurs de vapeur, cuve, ...), ainsi que les rambarde et les échelles.



Section de cylindre droit	
Paramètre	Description
$r_{\mathcal{C}}$	Rayon du cylindre $\mathcal{C}$ .
$\mathbf{a}_{\mathcal{C}}$	Vecteur unitaire représentant la direction de l'axe de $\mathcal{C}$ .
$\mathbf{c}_{\mathcal{C}}$	Point de passage de l'axe de $\mathcal{C}$ . Lorsque la longueur de $\mathcal{C}$ est finie, on prendra $\mathbf{c}_{\mathcal{C}}$ comme étant le centre de masse de $\mathcal{C}$ . Dans le cas contraire, tous point sur l'axe peut être utilisé.
$l_{\mathcal{C}}$	Longueur du cylindre tronqué $\mathcal{C}$ (distance entre ses deux extrémités).

### A.2.1 Cas du cylindre infini

Tout comme le plan infini, le cylindre infini se caractérise par son absence de limites. On considère alors que  $l_C = \infty$ , et tout point se situant sur la droite définissant l'axe du cylindre suffit à définir sa position. Au lieu des 7 paramètres libres nécessaires à la définition d'un cylindre tronqué, 5 paramètres libres suffisent à définir un cylindre infini: deux paramètres angulaires pour la direction de son axe  $\mathbf{a}_C$ , un paramètre pour son rayon  $r_C$  et deux paramètres pour sa position  $\mathbf{c}_C$ .

### A.2.2 Outils géométriques

**Distance** La distance séparant un point  $\mathbf{p}$  d'un cylindre infini  $\mathcal{C}$  est la suivante:

$$d(\mathbf{p}, \mathcal{C}) = \|\mathbf{a}_C \times (\mathbf{p} - \mathbf{c}_C)\| - r_C \quad (\text{A.6})$$

On note que cette distance est signée. Elle est en particulier négative lorsque le point se situe à l'intérieur du cylindre, positive sinon.

Lorsque le cylindre  $\mathcal{C}$  est borné, il est nécessaire d'intégrrer ses limites au calcul. On considérera que la surface du cylindre est définie par sa surface cylindrique d'une part, et les disques bornant le cylindre d'autre part. Posons

$$\begin{aligned} t(\mathbf{p}, \mathcal{C}) &= |(\mathbf{p} - \mathbf{c}_C) \cdot \mathbf{a}_C| - \frac{l_C}{2} \\ s(\mathbf{p}, \mathcal{C}) &= \|\mathbf{a}_C \times (\mathbf{p} - \mathbf{c}_C)\| - r_C \end{aligned}$$

$t(\mathbf{p}, \mathcal{C})$  définit la distance séparant  $\mathbf{p}$  du plan (infini) bornant  $\mathcal{C}$  dont il est le plus proche, et est positif si  $\mathbf{p}$  est à l'extérieur de  $\mathcal{C}$ , négatif sinon.  $s(\mathbf{p}, \mathcal{C})$  définit la distance au cylindre infini.

On a alors

$$d(\mathbf{p}, \mathcal{C}) = \begin{cases} \max(t(\mathbf{p}, \mathcal{C}), s(\mathbf{p}, \mathcal{C})) & \text{si } t(\mathbf{p}, \mathcal{C}) < 0 \text{ et } s(\mathbf{p}, \mathcal{C}) < 0 \\ \sqrt{\max(0, t(\mathbf{p}, \mathcal{C}))^2 + \max(0, s(\mathbf{p}, \mathcal{C}))^2} & \text{sinon} \end{cases} \quad (\text{A.7})$$

Cette distance possède les mêmes propriétés que la distance au cylindre infini: elle est négative lorsque le point se situe à l'intérieur du cylindre tronqué, positive sinon.

**Vecteur normal** Le vecteur normal au cylindre tronqué  $\mathcal{C}$  au point  $\mathbf{p}$  se trouvant à la surface de  $\mathcal{C}$  est:

$$\mathbf{n}_C(\mathbf{p}) = \begin{cases} \mathbf{u}(\mathbf{p}, \mathcal{C}) & \text{si } t(\mathbf{p}, \mathcal{C}) < 0 \\ \text{sgn}((\mathbf{p} - \mathbf{c}_C) \cdot \mathbf{a}_C) \cdot \mathbf{a}_C & \text{si } s(\mathbf{p}, \mathcal{C}) < 0 \\ \text{indéfinie} & \text{sinon} \end{cases} \quad (\text{A.8})$$

avec

$$\mathbf{u}(\mathbf{p}, \mathcal{C}) = \frac{\mathbf{a}_C \times ((\mathbf{p} - \mathbf{c}_C) \times \mathbf{a}_C)}{\|\mathbf{a}_C \times ((\mathbf{p} - \mathbf{c}_C) \times \mathbf{a}_C)\|}$$

et

$$\text{sgn}(x) = \begin{cases} -1 & \text{si } x < 0 \\ 1 & \text{sinon} \end{cases}$$

Pour le cylindre infini, la normale peut être calculée via la formule précédente en constatant que  $t(\mathbf{p}, \mathcal{C}) < 0$  dans ce cas (on a alors  $\mathbf{n}_C(\mathbf{p}) = \mathbf{u}(\mathbf{p}, \mathcal{C})$ ).

**Projection** Pour calculer la projection d'un point  $\mathbf{p}$  sur un cylindre tronqué  $\mathcal{C}$ , il faut considérer deux cas distincts. Soit le point est à l'intérieur du cylindre ( $s(\mathbf{p}, \mathcal{C}) < 0$  et  $t(\mathbf{p}, \mathcal{C}) < 0$ ), auquel cas:

$$\mathbf{p}_{\mathcal{C}}^{\downarrow} = \begin{cases} \mathbf{c}_{\mathcal{C}} + ((\mathbf{p} - \mathbf{c}_{\mathcal{C}}) \cdot \mathbf{a}_{\mathcal{C}}) \mathbf{a}_{\mathcal{C}} + r_{\mathcal{C}} \mathbf{u}(\mathbf{p}, \mathcal{C}) & \text{si } t(\mathbf{p}, \mathcal{C}) < s(\mathbf{p}, \mathcal{C}) \\ \mathbf{p} - [\text{sgn}((\mathbf{p} - \mathbf{c}_{\mathcal{C}}) \cdot \mathbf{a}_{\mathcal{C}}) t(\mathbf{p}, \mathcal{C})] \mathbf{a}_{\mathcal{C}} & \text{si } s(\mathbf{p}, \mathcal{C}) \leq t(\mathbf{p}, \mathcal{C}) \end{cases} \quad (\text{A.9})$$

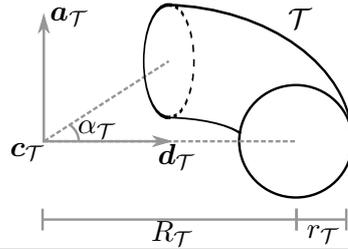
soit le point est à l'extérieur du cylindre ( $s(\mathbf{p}, \mathcal{C}) \geq 0$  ou  $t(\mathbf{p}, \mathcal{C}) \geq 0$ ):

$$\mathbf{p}_{\mathcal{C}}^{\downarrow} = \mathbf{c}_{\mathcal{C}} + \max\left(-\frac{l_{\mathcal{C}}}{2}, \min\left(\frac{l_{\mathcal{C}}}{2}, (\mathbf{p} - \mathbf{c}_{\mathcal{C}}) \cdot \mathbf{a}_{\mathcal{C}}\right)\right) \mathbf{a}_{\mathcal{C}} + [r_{\mathcal{C}} + \min(0, s(\mathbf{p}, \mathcal{C}))] \mathbf{u}(\mathbf{p}, \mathcal{C}) \quad (\text{A.10})$$

Le cas du cylindre infini peut être traité en constatant que  $t(\mathbf{p}, \mathcal{C}) < s(\mathbf{p}, \mathcal{C}) < 0$  dans ce cas.

### A.3 Tores à section circulaire

La principale fonction des tores est la description des coudes dans les lignes de tuyauterie. Toutefois, ils peuvent aussi représenter de manière marginale les poignées d'ouverture/fermeture de valves.



Section de tore circulaire	
Paramètre	Description
$r_{\mathcal{T}}$	Rayon mineur du tore $\mathcal{T}$ (rayon de la section circulaire).
$R_{\mathcal{T}}$	Rayon majeur de $\mathcal{T}$ .
$\mathbf{a}_{\mathcal{T}}$	Vecteur unitaire représentant la direction de l'axe de $\mathcal{T}$ (axe de révolution).
$\mathbf{d}_{\mathcal{T}}$	Vecteur unitaire, orthogonal à l'axe $\mathbf{a}_{\mathcal{T}}$ , indiquant le centre de la première extrémité.
$\mathbf{c}_{\mathcal{T}}$	Centre de $\mathcal{T}$ .
$\alpha_{\mathcal{T}}$	Angle d'ouverture de $\mathcal{T}$ (angle de révolution, correspondant à l'angle entre les deux extrémités).

#### A.3.1 Outils géométriques

**Distance** Si l'on ne considère pas l'angle d'ouverture  $\alpha_{\mathcal{T}}$  du tore  $\mathcal{T}$  (autrement dit,  $\alpha_{\mathcal{T}} = 2\pi$ ), la distance d'un point  $\mathbf{p}$  à  $\mathcal{T}$  est alors:

$$d(\mathbf{p}, \mathcal{T}) = \sqrt{(R_{\mathcal{T}} - \|\mathbf{a}_{\mathcal{T}} \times (\mathbf{p} - \mathbf{c}_{\mathcal{T}})\|)^2 + ((\mathbf{p} - \mathbf{c}_{\mathcal{T}}) \cdot \mathbf{a}_{\mathcal{T}})^2} - r_{\mathcal{T}} \quad (\text{A.11})$$

Cette distance est signée: elle est négative lorsque  $\mathbf{p}$  est à l'intérieur du tore, positive sinon. Par la suite, on utilisera  $s(\mathbf{p}, \mathcal{T})$  pour désigner la distance séparant le point  $\mathbf{p}$  du tore complet associé au tore  $\mathcal{T}$  potentiellement partiel ( $\alpha_{\mathcal{T}} < 2\pi$ ).

Pour intégrer les limites du tore partiel, définissons les deux disques  $\mathcal{D}_1$  et  $\mathcal{D}_2$  de rayon  $r_{\mathcal{T}}$  bornant le tore caractérisés par leurs centres respectifs  $\mathbf{e}_{\mathcal{T}}^1$  et  $\mathbf{e}_{\mathcal{T}}^2$ , ainsi que leur vecteurs normaux  $\mathbf{v}_{\mathcal{T}}^1$  et  $\mathbf{v}_{\mathcal{T}}^2$ :

$$\begin{aligned} \mathbf{e}_{\mathcal{T}}^1 &= \mathbf{c}_{\mathcal{T}} + R_{\mathcal{T}} \mathbf{d}_{\mathcal{T}} \\ \mathbf{v}_{\mathcal{T}}^1 &= \mathbf{d}_{\mathcal{T}} \times \mathbf{a}_{\mathcal{T}} \\ \mathbf{e}_{\mathcal{T}}^2 &= \mathbf{c}_{\mathcal{T}} + R_{\mathcal{T}} (\cos(\alpha_{\mathcal{T}}) \mathbf{d}_{\mathcal{T}} + \sin(\alpha_{\mathcal{T}}) (\mathbf{a}_{\mathcal{T}} \times \mathbf{d}_{\mathcal{T}})) \\ \mathbf{v}_{\mathcal{T}}^2 &= \cos(\alpha_{\mathcal{T}}) (\mathbf{a}_{\mathcal{T}} \times \mathbf{d}_{\mathcal{T}}) - \sin(\alpha_{\mathcal{T}}) \mathbf{d}_{\mathcal{T}} \end{aligned} \quad (\text{A.12})$$

La surface de  $\mathcal{T}$  défini sur l'angle  $\alpha_{\mathcal{T}}$  correspond alors à la surface torique ainsi qu'à ces deux disques.

La distance séparant un point  $\mathbf{p}$  d'un disque  $\mathcal{D}$  de rayon  $r$ , centré en  $\mathbf{e}$  de vecteur normal  $\mathbf{v}$  est:

$$d(\mathbf{p}, \mathcal{D}) = \sqrt{\max(0, ((\mathbf{p} - \mathbf{e}) \times \mathbf{v}) - r)^2 + ((\mathbf{p} - \mathbf{e}) \cdot \mathbf{v})^2} \quad (\text{A.13})$$

Introduisons un prédicat  $O(\mathbf{p}, \mathcal{T})$  permettant de vérifier si  $\mathbf{p}$  est en dehors des limites induites par l'angle d'ouverture  $\alpha_{\mathcal{T}}$  et  $\mathbf{d}_{\mathcal{T}}$ . L'expression de ce prédicat dépend de la valeur de  $\alpha_{\mathcal{T}}$ . Si  $\alpha_{\mathcal{T}} < \pi$

$$O(\mathbf{p}, \mathcal{T}) = \begin{cases} \text{VRAI} & \text{si } (\mathbf{p} - \mathbf{e}_{\mathcal{T}}^1) \cdot \mathbf{v}_{\mathcal{T}}^1 \geq 0 \text{ ou } (\mathbf{p} - \mathbf{e}_{\mathcal{T}}^2) \cdot \mathbf{v}_{\mathcal{T}}^2 \geq 0 \\ \text{FAUX} & \text{sinon} \end{cases} \quad (\text{A.14})$$

En revanche, si  $\alpha_{\mathcal{T}} \geq \pi$

$$O(\mathbf{p}, \mathcal{T}) = \begin{cases} \text{VRAI} & \text{si } (\mathbf{p} - \mathbf{e}_{\mathcal{T}}^1) \cdot \mathbf{v}_{\mathcal{T}}^1 \geq 0 \text{ et } (\mathbf{p} - \mathbf{e}_{\mathcal{T}}^2) \cdot \mathbf{v}_{\mathcal{T}}^2 \geq 0 \\ \text{FAUX} & \text{sinon} \end{cases} \quad (\text{A.15})$$

Partant des différents éléments introduits ci-avant, on obtient alors la distance du point  $\mathbf{p}$  au tore partiel  $\mathcal{T}$ :

$$d(\mathbf{p}, \mathcal{T}) = \begin{cases} \min(d(\mathbf{p}, \mathcal{D}_1), d(\mathbf{p}, \mathcal{D}_2)) & \text{si } O(\mathbf{p}, \mathcal{T}) \\ s(\mathbf{p}, \mathcal{T}) & \text{si } \neg O(\mathbf{p}, \mathcal{T}) \text{ et } s(\mathbf{p}, \mathcal{T}) \geq 0 \\ \max(s(\mathbf{p}, \mathcal{T}), d(\mathbf{p}, \mathcal{D}_1), d(\mathbf{p}, \mathcal{D}_2)) & \text{sinon} \end{cases} \quad (\text{A.16})$$

**Vecteur normal** Le vecteur normal au tore partiel  $\mathcal{T}$  en  $\mathbf{p}$  est:

$$\mathbf{n}_{\mathcal{T}}(\mathbf{p}) = \begin{cases} \mathbf{v}_{\mathcal{T}}^1 & \text{si } d(\mathbf{p}, \mathcal{D}_1) = 0 \text{ et } s(\mathbf{p}, \mathcal{T}) < 0 \\ \mathbf{v}_{\mathcal{T}}^2 & \text{si } d(\mathbf{p}, \mathcal{D}_2) = 0 \text{ et } s(\mathbf{p}, \mathcal{T}) < 0 \\ \frac{\mathbf{p} - \mathbf{q}(\mathbf{p}, \mathcal{T})}{\|\mathbf{p} - \mathbf{q}(\mathbf{p}, \mathcal{T})\|} & \text{si } s(\mathbf{p}, \mathcal{T}) = 0 \text{ et } d(\mathbf{p}, \mathcal{D}_1) \neq 0 \text{ et } d(\mathbf{p}, \mathcal{D}_2) \neq 0 \\ \text{indéfinie} & \text{sinon} \end{cases} \quad (\text{A.17})$$

où

$$\mathbf{q}(\mathbf{p}, \mathcal{T}) = \mathbf{c}_{\mathcal{T}} + R_{\mathcal{T}} \mathbf{u}(\mathbf{p}, \mathcal{T})$$

avec

$$\mathbf{u}(\mathbf{p}, \mathcal{T}) = \frac{\mathbf{a}_{\mathcal{T}} \times ((\mathbf{p} - \mathbf{c}_{\mathcal{T}}) \times \mathbf{a}_{\mathcal{T}})}{\|\mathbf{a}_{\mathcal{T}} \times ((\mathbf{p} - \mathbf{c}_{\mathcal{T}}) \times \mathbf{a}_{\mathcal{T}})\|}$$

**Projection** La projection du point  $\mathbf{p}$  sur le disque  $\mathcal{D}$  centré en  $\mathbf{e}$  normal à  $\mathbf{t}$  de rayon  $r$  s'exprime comme suit:

$$\mathbf{p}_{\mathcal{D}}^{\downarrow} = \min(R_{\mathcal{T}}, \|(\mathbf{p} - \mathbf{e}) \times \mathbf{t}\|) \left( \frac{\mathbf{t} \times (\mathbf{p} - \mathbf{e}) \times \mathbf{t}}{\|\mathbf{t} \times (\mathbf{p} - \mathbf{e}) \times \mathbf{t}\|} \right) \quad (\text{A.18})$$

On en déduit l'expression de la projection d'un point  $\mathbf{p}$  sur un tore partiel  $\mathcal{T}$ . Considérons tout d'abord le cas où  $\mathbf{p}$  est en dehors de  $\mathcal{T}$  (c'est à dire  $O(\mathbf{p}, \mathcal{T})$  est vérifié et  $s(\mathbf{p}, \mathcal{T}) < 0$ ):

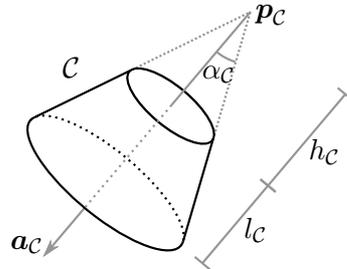
$$\mathbf{p}_{\mathcal{T}}^{\downarrow} = \begin{cases} \mathbf{p}_{\mathcal{D}_1}^{\downarrow} & \text{si } O(\mathbf{p}, \mathcal{T}) \text{ et } d(\mathbf{p}, \mathcal{D}_1) < d(\mathbf{p}, \mathcal{D}_2) \\ \mathbf{p}_{\mathcal{D}_2}^{\downarrow} & \text{si } O(\mathbf{p}, \mathcal{T}) \text{ et } d(\mathbf{p}, \mathcal{D}_2) < d(\mathbf{p}, \mathcal{D}_1) \\ \mathbf{q}(\mathbf{p}, \mathcal{T}) + r_{\mathcal{T}} \left( \frac{\mathbf{p} - \mathbf{q}(\mathbf{p}, \mathcal{T})}{\|\mathbf{p} - \mathbf{q}(\mathbf{p}, \mathcal{T})\|} \right) & \text{sinon} \end{cases} \quad (\text{A.19})$$

Lorsque  $\mathbf{p}$  est à l'intérieur de  $\mathcal{T}$ :

$$\mathbf{p}_{\mathcal{T}}^{\downarrow} = \begin{cases} \mathbf{p}_{\mathcal{D}_1}^{\downarrow} & \text{si } d(\mathbf{p}, \mathcal{D}_1) \geq d(\mathbf{p}, \mathcal{D}_2) \text{ et } d(\mathbf{p}, \mathcal{D}_1) > s(\mathbf{p}, \mathcal{T}) \\ \mathbf{p}_{\mathcal{D}_2}^{\downarrow} & \text{si } d(\mathbf{p}, \mathcal{D}_2) \geq d(\mathbf{p}, \mathcal{D}_1) \text{ et } d(\mathbf{p}, \mathcal{D}_2) > s(\mathbf{p}, \mathcal{T}) \\ \mathbf{q}(\mathbf{p}, \mathcal{T}) + r_{\mathcal{T}} \left( \frac{\mathbf{p} - \mathbf{q}(\mathbf{p}, \mathcal{T})}{\|\mathbf{p} - \mathbf{q}(\mathbf{p}, \mathcal{T})\|} \right) & \text{sinon} \end{cases} \quad (\text{A.20})$$

## A.4 Cônes

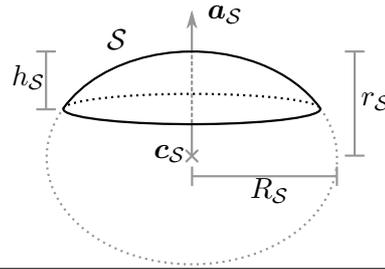
Les cônes servent quasi exclusivement à décrire les réducteurs dans les lignes de tuyauterie. Leur intérêt est donc relativement limité, puisqu'ils sont finalement peu présents contrairement aux autres types de primitives, et ont un rôle secondaire dans la structure des scènes.



Section de cône de révolution	
Paramètre	Description
$\alpha_{\mathcal{C}}$	Demi angle du cône $\mathcal{C}$ .
$\mathbf{a}_{\mathcal{C}}$	Vecteur unitaire représentant la direction de l'axe de $\mathcal{C}$ (axe de révolution).
$\mathbf{p}_{\mathcal{C}}$	Apex (sommet) de $\mathcal{C}$ .
$h_{\mathcal{T}}$	Hauteur de tronçage de $\mathcal{C}$ (distance entre l'extrémité supérieure et l'apex).
$l_{\mathcal{T}}$	Longueur de la section de cône $\mathcal{C}$ (distance entre ses deux extrémités).

## A.5 Sphéroïde

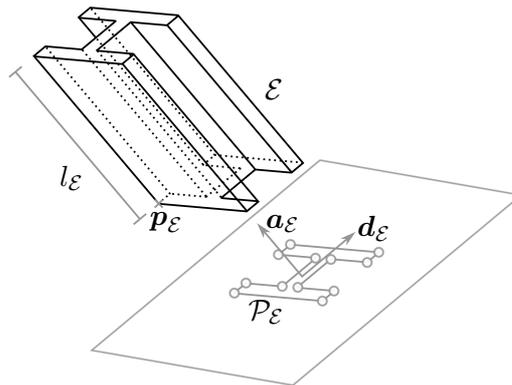
Les seuls composant sphéroïdaux significatifs sont les extrémités de cuve.



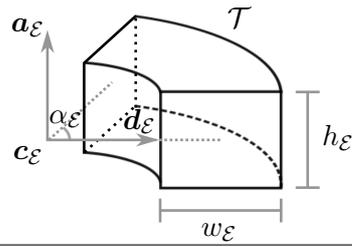
Calotte sphéroïdale	
Paramètre	Description
$\mathbf{a}_S$	Vecteur unitaire représentant la direction de l'axe de $\mathcal{S}$ (axe de révolution).
$\mathbf{c}_S$	Centre de $\mathcal{S}$ .
$r_S$	Rayon de $\mathcal{S}$ le long de son axe $\mathbf{a}_S$ .
$R_S$	Rayon de $\mathcal{S}$ dans le plan perpendiculaire à son axe $\mathbf{a}_S$ .
$h_S$	Hauteur de la calotte sphéroïdale (plus grande distance entre la calotte et sa base).

## A.6 Divers

Il existe d'autres types de primitives qui peuvent être utilisés pour décrire divers composants. Typiquement, les tores rectangulaires servent de coude dans les conduits d'aération et certains solides générés par extrusion de profil servent à décrire des poutres (IPN, IPE, IPEA, ...). On note toutefois que les profils extrudés peuvent être définis comme des assemblages de plans. De même, les tores à section rectangulaires peuvent être construits à partir de cylindres (joints interne et externe) et de plans.



Profil extrudé	
Paramètre	Description
$\mathbf{a}_E$	Vecteur unitaire représentant la direction d'extrusion de $\mathcal{E}$ (génératrice).
$\mathbf{d}_E$	Vecteur unitaire orthogonal à l'axe $\mathbf{a}_E$ , définissant l'orientation de $\mathcal{E}$ autour de $\mathbf{a}_E$ dans l'espace.
$\mathcal{P}_E$	Liste de points 2D définissant le contour extrudé.
$\mathbf{p}_E$	Position 3D du premier point du contour extrudé.
$l_E$	Longueur de l'extrusion (distance entre ses extrémités).



Section de tore rectangulaire	
Paramètre	Description
$w_{\mathcal{T}}$	Largeur de la section rectangulaire du tore $\mathcal{T}$ .
$h_{\mathcal{T}}$	Hauteur de la section rectangulaire de $\mathcal{T}$ .
$R_{\mathcal{T}}$	Rayon de $\mathcal{T}$ .
$\mathbf{a}_{\mathcal{T}}$	Vecteur unitaire représentant la direction de l'axe de $\mathcal{T}$ (axe de révolution).
$\mathbf{d}_{\mathcal{T}}$	Vecteur unitaire, orthogonal à l'axe $\mathbf{a}_{\mathcal{T}}$ , indiquant le centre de la première extrémité.
$\mathbf{c}_{\mathcal{T}}$	Centre de $\mathcal{T}$ .
$\alpha_{\mathcal{T}}$	Angle d'ouverture de $\mathcal{T}$ (angle de révolution, correspondant à l'angle entre les deux extrémités).

# Appendix **B**

## Collisions entre primitives géométriques

### Sommaire

---

<b>B.1 Plan - Plan</b> . . . . .	185
<b>B.2 Cylindre - Cylindre</b> . . . . .	186
<b>B.3 Cylindre - Plan</b> . . . . .	188

---

---

### Préambule

---

*La détection des connexions entre primitives géométriques composant un modèle CAO repose notamment sur l'identification des collisions survenant entre ces objets. En effet, nous considérons dans nos travaux que deux primitives distinctes sont connectées si leur intersection dans  $\mathbb{R}^3$  est non vide, ou si elles plébiscitent toutes deux un même point du nuage.*

*Cette annexe aborde la question de la détection des collisions. Il s'agit de décider, étant donnée un couple de primitives, si leur intersection est vide ou non. Bien que simple en apparence, la réponse à cette question peut demander une étude poussée de la situation.*

---

## B.1 Plan - Plan

La détection de la collision entre plans est relativement simple. Considérons tout d'abord le cas des plans infinis. Deux plans non bornés  $\mathcal{P}_i$  et  $\mathcal{P}_j$  sont disjoints si et seulement si ils sont parallèles et non confondus, par définition. Ce constat trivial de non-intersection peut être établi en vérifiant simplement que:

$$\mathbf{n}_{\mathcal{P}_i} \cdot \mathbf{n}_{\mathcal{P}_j} \neq 1 \quad \text{ou} \quad (\mathbf{c}_{\mathcal{P}_i} - \mathbf{c}_{\mathcal{P}_j}) \cdot \mathbf{n}_{\mathcal{P}_j} \neq 0$$

où  $\mathbf{n}_{\mathcal{P}}$  et  $\mathbf{c}_{\mathcal{P}}$  désignent respectivement le vecteur unitaire normal au plan  $\mathcal{P}$  et un point quelconque sur ce même plan.

Lorsque les plans sont bornés (rappelons que l'on considère alors des patches rectangulaires plans - cf. annexe A), le test de collision se complexifie très légèrement. On peut alors s'appuyer sur un test connu de certaines communautés telles que la simulation ou les jeux vidéos, basé sur le théorème dit de "l'axe séparateur". Ce théorème stipule que deux volumes convexes plongés dans un espace quelconque sont disjoints si et seulement si il existe un hyperplan de l'espace qui les sépare, c'est-à-dire tel que les deux volumes se situent de part et d'autre cet hyperplan sans le toucher. La direction orthogonale à un tel hyperplan est appelée axe séparateur.

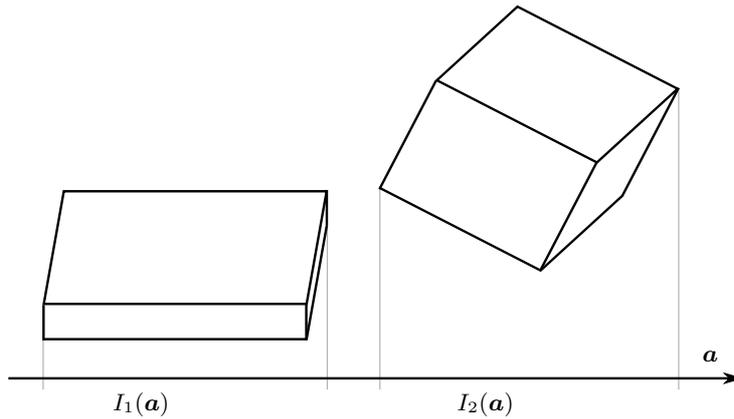
En particulier, si deux polyèdres convexes dans  $\mathbb{R}^3$  sont disjoints, il existe un axe séparateur orthogonal à une face d'un des deux polyèdres ou à deux arêtes prises sur chacun des polyèdres. Si aucune des directions ainsi définies ne constitue un axe séparateur, on peut alors conclure que les polyèdres sont disjoints. Dans le cas contraire, il y a collision.

Dans la situation qui nous intéresse ici, un patch rectangulaire plan peut être considéré comme un parallélépipède d'épaisseur nulle. La méthode de l'axe séparateur peut donc s'appliquer directement pour détecter la collision des plans en testant 15 directions seulement, puisqu'il n'y a dans une telle scène qu'un maximum de  $3 + 3 = 6$  faces distinctes et  $3 \times 3 = 9$  paires d'arêtes distinctes. Si  $\mathbf{d}_{\mathcal{P}}$  désigne le vecteur d'orientation d'un patch rectangulaire  $\mathcal{P}$ , on devra alors tester les 15 directions suivantes:

à partir des faces:	à partir des arêtes
$\mathbf{n}_{\mathcal{P}_i}$	$\mathbf{n}_{\mathcal{P}_i} \times \mathbf{n}_{\mathcal{P}_j}$
$\mathbf{n}_{\mathcal{P}_j}$	$\mathbf{n}_{\mathcal{P}_i} \times \mathbf{d}_{\mathcal{P}_j}$
$\mathbf{d}_{\mathcal{P}_i}$	$\mathbf{d}_{\mathcal{P}_i} \times \mathbf{n}_{\mathcal{P}_j}$
$\mathbf{d}_{\mathcal{P}_j}$	$\mathbf{n}_{\mathcal{P}_i} \times (\mathbf{n}_{\mathcal{P}_j} \times \mathbf{d}_{\mathcal{P}_j})$
$\mathbf{n}_{\mathcal{P}_i} \times \mathbf{d}_{\mathcal{P}_i}$	$(\mathbf{n}_{\mathcal{P}_i} \times \mathbf{d}_{\mathcal{P}_i}) \times \mathbf{n}_{\mathcal{P}_j}$
$\mathbf{n}_{\mathcal{P}_j} \times \mathbf{d}_{\mathcal{P}_j}$	$\mathbf{d}_{\mathcal{P}_i} \times (\mathbf{n}_{\mathcal{P}_j} \times \mathbf{d}_{\mathcal{P}_j})$
	$(\mathbf{n}_{\mathcal{P}_i} \times \mathbf{d}_{\mathcal{P}_i}) \times \mathbf{d}_{\mathcal{P}_j}$
	$\mathbf{d}_{\mathcal{P}_i} \times \mathbf{d}_{\mathcal{P}_j}$
	$(\mathbf{n}_{\mathcal{P}_i} \times \mathbf{d}_{\mathcal{P}_i}) \times (\mathbf{n}_{\mathcal{P}_j} \times \mathbf{d}_{\mathcal{P}_j})$

Soient  $(\mathbf{s}_0^i, \mathbf{s}_1^i, \mathbf{s}_2^i, \mathbf{s}_3^i)$  et  $(\mathbf{s}_0^j, \mathbf{s}_1^j, \mathbf{s}_2^j, \mathbf{s}_3^j)$  les sommets définissant les bords rectangulaires des plans  $\mathcal{P}_i$  et  $\mathcal{P}_j$  respectivement. L'axe supporté par le vecteur  $\mathbf{a}$  est séparateur si  $I_i(\mathbf{a}) \cap I_j(\mathbf{a}) = \emptyset$ , où  $I_k(\mathbf{a})$  désigne l'intervalle obtenu par projection orthogonale des sommets de  $\mathcal{P}_k$  (avec  $k \in \{i, j\}$ ) sur la direction  $\mathbf{a}$ :

$$I_k(\mathbf{a}) = \left[ \min_l (\mathbf{s}_l^k \cdot \mathbf{a}), \max_l (\mathbf{s}_l^k \cdot \mathbf{a}) \right]$$



**Figure B.1:** Détection de collisions de boîtes par test de l'axe séparateur. Sur cet exemple, l'axe  $\mathbf{a}$  perpendiculaire à une face de l'objet de gauche sépare les deux pavés, dont on peut donc conclure qu'ils sont disjoints. Comme les plans peuvent être considérés comme des pavés d'épaisseur nulle, ce test peut être utilisé pour détecter la collision de plans.

## B.2 Cylindre - Cylindre

La détection de la collision de cylindres est quant à elle nettement plus complexe. Si les cylindres sont de longueur infinie, le test est trivial puisqu'il suffit de calculer la distance  $\chi$  séparant leurs axes (le détail de ce calcul de  $\chi$  est présenté en section 2.3.3, équation 2.35) afin de déterminer si cette distance est inférieure à la somme de leurs rayons. Ainsi les cylindres infinis  $\mathcal{C}_i$  et  $\mathcal{C}_j$  sont en collision si et seulement si  $d \leq r_{\mathcal{C}_i} + r_{\mathcal{C}_j}$ .

Si les cylindres  $\mathcal{C}_i$  et  $\mathcal{C}_j$  sont bornés et si les cylindres infinis qui les supportent sont en collision (sans quoi les cylindres sont disjoints quoiqu'il en soit), il faut alors distinguer les différentes situations pouvant être rencontrées en fonction des positions relatives des disques faisant office d'extrémité pour les cylindres. On distingue ainsi deux cas en fonction de la situation des primitives l'une par rapport à l'autre.

### Cas standard: les cylindres s'intersectent au niveau de leurs flancs

Ce cas peut être repéré en étudiant les points  $\mathbf{q}_i$  et  $\mathbf{q}_j$  situés respectivement sur les axes de  $\mathcal{C}_i$  et  $\mathcal{C}_j$ , correspondant aux points minimisant la distance entre ces axes (cf. équation 2.33). Si les cylindres s'intersectent au niveau de leur flancs, alors ces points se faisant face se situent tous deux à l'intérieur du cylindre à partir duquel ils sont définis.

Si la projection de  $\mathbf{q}_i$  sur le cylindre  $\mathcal{C}_j$  est à l'intérieur de  $\mathcal{C}_j$ , ou réciproquement si la projection de  $\mathbf{q}_j$  sur le cylindre  $\mathcal{C}_i$  est à l'intérieur de  $\mathcal{C}_i$ , on peut alors conclure que les cylindres sont en collision puisqu'ils partagent un point de  $\mathbb{R}^3$  en leur intérieur (en considérant que la surface fait aussi partie de l'intérieur). Dans le cas contraire, les cylindres ne s'intersectent pas au niveau des flancs et il est alors nécessaire de tester leurs extrémités (ci-après).

Notons au passage que ce test permet aussi de détecter l'intersection pour certaines des situations où la collision implique les extrémités, et sa simplicité permet alors d'éviter un excès de calculs.

### Cas atypique : la collision implique les disques extrémaux

Le test précédent ne suffit pas à traiter tous les cas possibles. S'il échoue, il est alors nécessaire d'étudier le rapport entre les disques extrémaux d'un cylindre et le second cylindre. Supposons que l'on s'intéresse au premier disque extrémal de  $\mathcal{C}_i$  centré en  $\mathbf{s}_{\mathcal{C}_i} = \mathbf{c}_{\mathcal{C}_i} + \frac{l_{\mathcal{C}_i}}{2} \mathbf{a}_{\mathcal{C}_i}$ , de rayon  $r_{\mathcal{C}_i}$  et défini dans le plan de vecteur normal  $\mathbf{a}_{\mathcal{C}_i}$ . Il s'agit alors de tester si ce disque est en collision avec le second cylindre  $\mathcal{C}_j$ .

On distingue alors deux sous-cas distincts en fonction de l'endroit où le disque intersecte  $\mathcal{C}_j$ .

**Sous-cas 1: la collision survient uniquement au niveau du flanc de  $\mathcal{C}_j$**  Pour traiter ce cas il nous faut identifier le point  $\mathbf{r}$  sur le disque qui soit le plus proche de l'axe de  $\mathcal{C}_j$ . Si  $\mathbf{r}$  est à l'intérieur de  $\mathcal{C}_j$ , alors il y a collision.

Le point  $\mathbf{r}$  est calculé grâce à l'intersection  $\mathbf{p}$  de l'axe de  $\mathcal{C}_j$  avec le plan portant le disque étudié.

$$\mathbf{r} = \mathbf{s}_{\mathcal{C}_i} + \min(r_{\mathcal{C}_i}, \|\mathbf{p} - \mathbf{s}_{\mathcal{C}_i}\|) \cdot \left( \frac{\mathbf{p} - \mathbf{s}_{\mathcal{C}_i}}{\|\mathbf{p} - \mathbf{s}_{\mathcal{C}_i}\|} \right)$$

avec

$$\mathbf{p} = \mathbf{c}_{\mathcal{C}_j} + \left( \frac{\mathbf{a}_{\mathcal{C}_i} \cdot \mathbf{s}_{\mathcal{C}_i} - \mathbf{a}_{\mathcal{C}_i} \cdot \mathbf{c}_{\mathcal{C}_j}}{\|\mathbf{a}_{\mathcal{C}_i} \cdot \mathbf{a}_{\mathcal{C}_j}\|} \right) \cdot \mathbf{a}_{\mathcal{C}_j}$$

Dans le cas où le point  $\mathbf{p}$  est indéfini (lorsque  $\mathbf{a}_{\mathcal{C}_i} \cdot \mathbf{a}_{\mathcal{C}_j} = 0$ , c'est à dire lorsque les axes des cylindres sont orthogonaux), on calcule  $\mathbf{r}$  en projetant le centre  $\mathbf{s}_{\mathcal{C}_i}$  du disque sur l'axe de  $\mathcal{C}_j$ , puis en reprojétant le résultat  $\mathbf{r}^0$  sur le disque:

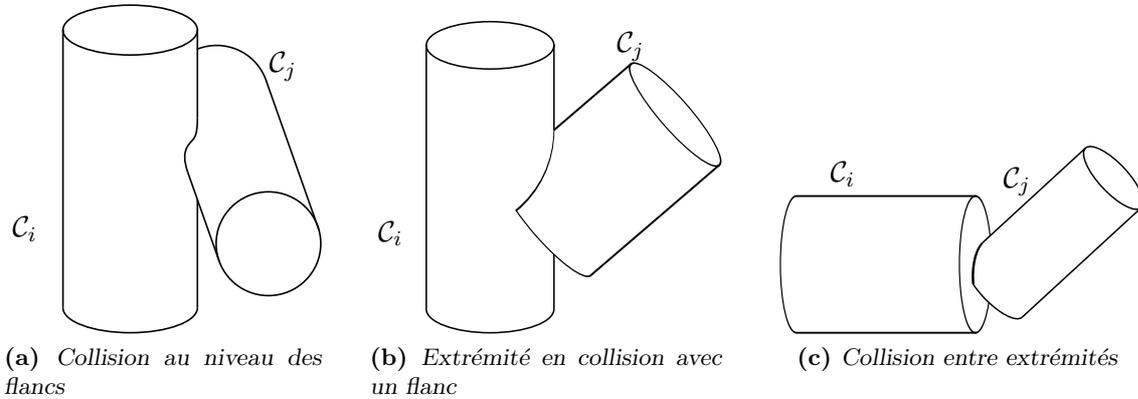
$$\begin{aligned} \mathbf{r}^0 &= \mathbf{c}_{\mathcal{C}_j} + ((\mathbf{s}_{\mathcal{C}_i} - \mathbf{c}_{\mathcal{C}_j}) \cdot \mathbf{a}_{\mathcal{C}_j}) \cdot \mathbf{a}_{\mathcal{C}_j} \\ \mathbf{r}^1 &= \mathbf{r}^0 - ((\mathbf{r}^0 - \mathbf{s}_{\mathcal{C}_i}) \cdot \mathbf{a}_{\mathcal{C}_i}) \cdot \mathbf{a}_{\mathcal{C}_i} \\ \mathbf{r} &= \mathbf{s}_{\mathcal{C}_i} + \min(r_{\mathcal{C}_i}, \|\mathbf{r}^1 - \mathbf{s}_{\mathcal{C}_i}\|) \cdot \left( \frac{\mathbf{r}^1 - \mathbf{s}_{\mathcal{C}_i}}{\|\mathbf{r}^1 - \mathbf{s}_{\mathcal{C}_i}\|} \right) \end{aligned}$$

**Sous-cas 2: la collision implique les disques extrémaux de  $\mathcal{C}_j$**  Pour traiter le dernier cas envisageable, il faut étudier la collision du disque avec chacun des deux disques extrémaux de  $\mathcal{C}_j$ .

Deux disques plongés dans  $\mathbb{R}^3$  sont en collision si la droite issue de l'intersection des plans qui les supportent les coupe en deux segments non disjoints. L'algorithme de la détection de collision de deux disques centrés en  $\mathbf{c}_1$  et  $\mathbf{c}_2$ , de rayons  $r_1$  et  $r_2$  est donc le suivant:

1. calculer la droite d'intersection des plans supportant les disques à tester. Soient  $\mathbf{p}$  un point de cette droite et  $\mathbf{d}$  son vecteur directeur.
2. identifier les racines de l'équation du second degré d'inconnue  $x$ :  $\|\mathbf{c}_1 - (\mathbf{p} + x\mathbf{d})\| = r_1^2$ . Ces racines définissent un intervalle  $I_1$  éventuellement vide ou singulier selon qu'il existe aucune, une ou deux solutions à l'équation posée. Elles correspondent à l'intersection de la droite  $(\mathbf{p}, \mathbf{d})$  avec le cercle centrée en  $\mathbf{c}_1$  de rayon  $r_1$  sachant que la droite appartient au même plan que ce disque.
3. répéter l'étape 2 de manière à calculer l'intervalle  $I_2$  correspondant à l'intersection de la droite  $(\mathbf{p}, \mathbf{d})$  avec le cercle de centre  $\mathbf{c}_2$  de rayon  $r_2$ .
4. les cercles sont en collision si et seulement si  $I_1 \cap I_2 \neq \emptyset$ .

**Test de tous les disques extrémaux** Ces tests doivent être répétés pour le second disque extrémal de  $\mathcal{C}_i$  vis-à-vis de  $\mathcal{C}_j$ , puis pour les deux disques définissant les extrémités de  $\mathcal{C}_j$  vis-à-vis de  $\mathcal{C}_i$ .



**Figure B.2:** Différents cas à traiter pour la détection de la collision de cylindres. Si l'étude de chacun de ces cas ne permet pas de conclure à l'intersection des formes, alors elles sont disjointes.

### Bilan des différents cas

Dans le cas où la totalité des tests précédents pour toutes situations mentionnées n'auraient pas conclu la présence de collision, on peut alors déduire que les cylindres bornés  $\mathcal{C}_i$  et  $\mathcal{C}_j$  sont disjointes.

## B.3 Cylindre - Plan

Dans la même idée qu'un couple de plans infinis, un cylindre et un plan tous deux infinis s'intersectent à moins qu'ils ne soient parallèles et suffisamment éloignés l'un de l'autre. Ainsi, le plan infini  $\mathcal{P}$  et le cylindre  $\mathcal{C}$  sont disjointes si et seulement si  $\mathbf{n}_P \cdot \mathbf{a}_C = 0$  et  $|(\mathbf{c}_C - \mathbf{c}_P) \cdot \mathbf{n}_P| > r_C$ .

Pour traiter le cas des primitives bornées, nous recourons à une approche en deux temps.

### Première étape : couper les bordures du plan

Seule la partie du plan  $\mathcal{P}$  définie entre les extrémités du cylindre  $\mathcal{C}$  est susceptible d'intersecter ce dernier. Nous nous proposons donc, dans un premier temps, de supprimer les parties superflues pour étudier le problème. Pour chacun des deux plans portant les extrémités du cylindre  $\mathcal{C}$ , on étudie son intersection avec les arêtes délimitant  $\mathcal{P}$ . Notons que, du fait de la convexité des bornes de  $\mathcal{P}$ , deux arêtes de  $\mathcal{P}$  au plus sont susceptibles de couper le plan étudié, et que l'opération de suppression décrite ici préserve cette propriété de convexité. En parcourant les arêtes, on identifie ainsi au maximum deux nouveaux points appartenant simultanément à la bordure de  $\mathcal{P}$  et au plan délimitant  $\mathcal{C}$  que l'on étudie. Ces nouveaux sommets sont insérés dans la définition de la bordure de  $\mathcal{P}$ .

Une fois l'opération de mise à jour de la bordure effectuée pour chacun des deux plans extrémaux de  $\mathcal{C}$ , on supprime les sommets ne se situant pas entre ces plans extrémaux, tout

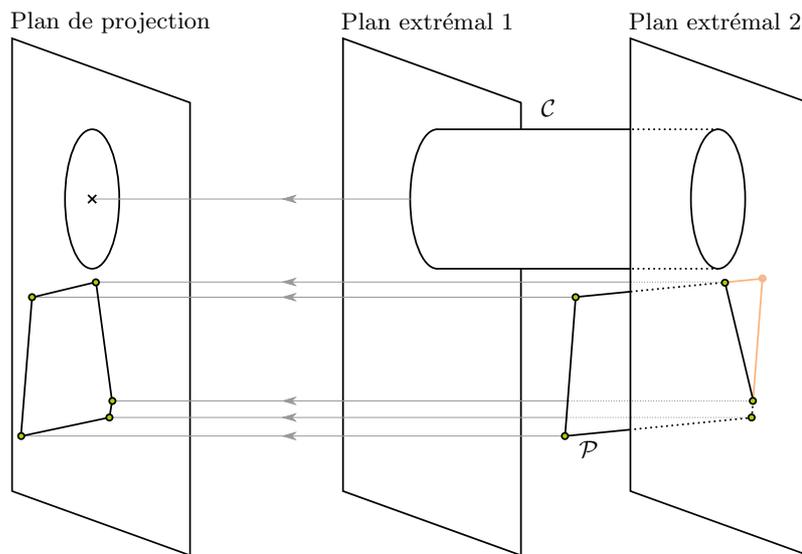
en conservant l'ordre de la liste définissant le contour de  $\mathcal{P}$  (la suppression d'un sommet entraîne ainsi implicitement la création d'une nouvelle arête reliant son prédécesseur et son successeur dans la liste). Si la bordure de  $\mathcal{P}$  obtenue à l'issue de cette opération est vide de sommets, alors les primitives  $\mathcal{P}$  et  $\mathcal{C}$  sont disjointes.

Le bordures de  $\mathcal{P}$  ainsi mises à jour permettent d'étudier le problème de la collision en deux dimensions, puisque nous ne conservons ici que l'information pertinente pour ce test.

### Seconde étape : projection et résolution en 2D

Considérons le polygone obtenu après projection des sommets définissant la bordure de  $\mathcal{P}$  dans le plan orthogonal à  $\mathcal{C}$ . Les primitives sont alors disjointes si et seulement si ce polygone et le cercle générateur du cylindre sont disjointes dans ce plan.

Ce test en 2D peut être effectué en deux temps. Tout d'abord, il s'agit de vérifier que le centre du cercle générateur ne se situe pas à l'intérieur du polygone. Si c'est le cas, les primitives sont en collision. Dans le cas contraire, il nous faut calculer la distance du centre du cercle générateur au polygone (cette distance se définit comme la plus petite des distances séparant le centre et les segments délimitant le polygone). Si cette distance est inférieure au rayon du cylindre, alors il y a collision. Dans le cas contraire, les primitives sont disjointes.



**Figure B.3:** Mécanismes de la détection de collision plan-cylindre. Après avoir supprimé (orange) les parties de la bordure du plan se situant hors de la zone de collision potentielle, les primitives sont projetées dans un plan orthogonal au cylindre. L'étude de l'intersection du cercle et du polygone obtenu dans ce plan est relativement simple et permet de conclure, sur cet exemple, que les primitives sont disjointes.



# Appendix **C**

## Accès rapide aux points

### Sommaire

---

<b>C.1 Arbres de subdivision</b> . . . . .	<b>193</b>
C.1.1 <i>Octree</i> . . . . .	193
C.1.2 <i>KD-tree</i> . . . . .	193
<b>C.2 Requête de plus proche voisin</b> . . . . .	<b>195</b>
<b>C.3 Voisinage d'une primitive</b> . . . . .	<b>195</b>

---

---

### Préambule

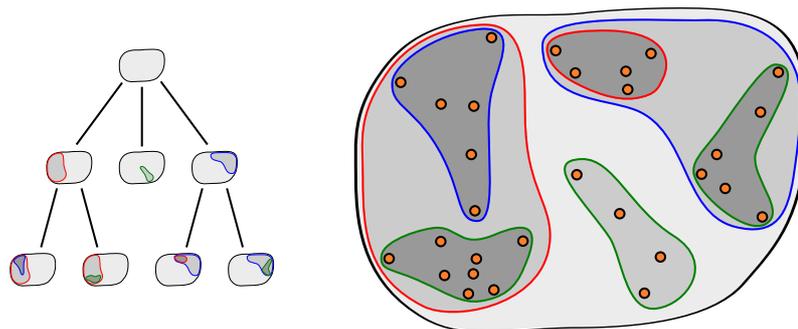
---

*Une part considérable des traitements effectués sur les nuages de points se résume à des calcul de voisinages de points (le calcul des vecteurs normaux en est un exemple). L'approche naïve basé sur le parcours de la totalité du nuage de points pour en extraire les voisins d'un point requête n'est pas envisageable en pratique, puisque trop longue à réaliser. D'autant plus que le nombre de requêtes adressées au nuage est en général important. Cette annexe présente quelques unes des approches couramment utilisées pour réduire considérablement la complexité temporelle de ce type de requête.*

---

## C.1 Arbres de subdivision

Pour s'affranchir algorithmiquement du surcôt introduit par les opérations de recherche d'éléments, il est courant d'associer au nuage de points une structure de découpage hiérarchique de l'espace qu'il occupe. Ces structures peuvent être représentées sous forme d'arborescences (cf. figure C.1), dont la traversée permet d'exclure au plus tôt l'exploration de grandes quantités de points qui ne sont pas pertinents vis-à-vis de la requête adressée au nuage de points. Les deux structures les plus couramment utilisées dans ce domaine sont l'*octree* et le *KD-tree*, toutes deux généralisables à un espace  $\mathbb{R}^d$  de dimension  $d$  quelconque.



**Figure C.1:** Principe de la subdivision spatiale hiérarchique. La subdivision peut être représentée sous la forme d'un arbre. Chaque nœud  $y$  représente un ensemble de points. Les fils d'un nœud quant à eux représentent la séparation de ce nœud en sous-ensembles disjoints. On parlera de "cellule" pour désigner le nœud d'un arbre, et son extension spatiale.

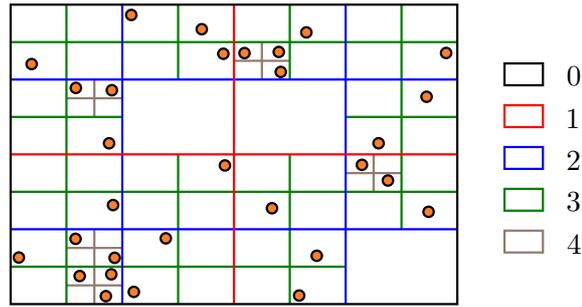
### C.1.1 *Octree*

L'*octree* propose un découpage récursif de l'espace en pavés similaires. Cette structure s'appuie généralement sur la boîte englobante du nuage de points, faisant office de cellule initiale pour le découpage. La construction de l'*octree* partitionne ensuite chaque pavé du niveau courant en 8 cellules identiques ( $2^d$  cellules en dimension  $d$ ) parmi lesquelles seules celles contenant effectivement des données sont conservées. Ce processus est ensuite répété jusqu'à l'obtention de cellules ne contenant qu'un point, ou de cellules de suffisamment petites pour l'application souhaitée. La profondeur maximale d'un *octree* est en général dépendante de cette taille minimale spécifiée pour les cellules.

Il existe différentes manières de stocker efficacement un *octree* [BWK02, DD04, GM06]. D. Girardeau-Montaut [GM06] propose par exemple d'encoder chaque cellule de l'*octree* à l'aide d'un identifiant unique dont la décomposition binaire permet de déterminer la position de la cellule dans la structure. Chaque cellule peut ainsi être adressée en temps constant.

### C.1.2 *KD-tree*

Le *KD-tree*, proposé par J. Bentley [Ben75], repose sur une approche de partitionnement binaire de l'espace, en alternant les axes de coupe. Chaque étape de subdivision sépare un ensemble de points en deux sous-ensembles disjoints séparés par un plan. Le plan de coupe d'une subdivision est orthogonal aux plans de coupe des deux subdivisions parentes (*i.e.* les deux subdivisions successives ayant engendré le sous-ensemble considéré). Habituellement,

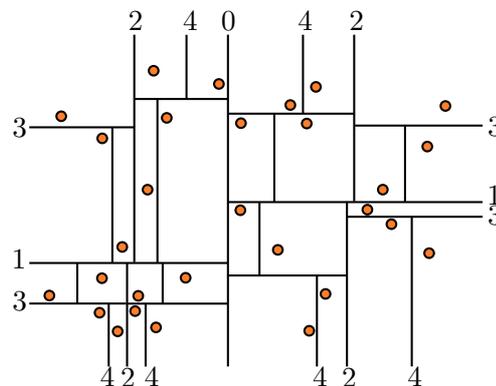


**Figure C.2:** Découpage d'un nuage de points en quadtree (équivalent 2D de l'octree). Les couleurs correspondent à un niveau de coupe

les axes canoniques  $x$ ,  $y$  et  $z$  sont utilisés pour déterminer les directions de coupe: le plan de coupe du premier niveau de subdivision est orthogonal à  $x$ , puis les plans au second niveau sont orthogonaux à  $y$ , ceux du troisième niveau sont orthogonaux à  $z$ , les coupes du quatrième niveau se font à nouveau selon  $x$ , etc.

L'efficacité de cette structure est essentiellement déterminée par la stratégie adoptée concernant le choix des plans de coupe. L'approche la plus simple consiste en la séparation de chaque ensemble de points en deux sous-ensembles de même cardinal (dans la mesure du possible). Le placement du plan de coupe est alors déterminé par la médiane des points projetés sur l'axe de coupe courant. Contrairement au découpage en *octree* qui nécessite de spécifier une taille minimale de cellule pour garantir l'arrêt de la subdivision, le découpage en *KD-tree* par coupe médiane s'arrête spontanément puisque le nombre d'éléments par cellule diminue nécessairement au cours du processus. L'intérêt majeur de cette approche tient au fait qu'elle garantit l'obtention d'un arbre de subdivision équilibré: chaque feuille contient un unique point, et chaque cellule interne possède deux fils de profondeur égale (à une unité près). La profondeur totale du *KD-tree* associé à un nuage de  $n$  points est alors  $\lceil \log_2(n) \rceil$ .

En pratique, est est préférable d'arrêter la subdivision lorsque la cellule courante contient une quantité de points jugée suffisante. Nous avons ainsi constaté, sur la plupart des cas que nous avons testés, qu'une trentaine de points par feuille de l'arbre de subdivision permettent d'obtenir une efficacité optimale dans les temps de réponse aux requêtes de voisinage.



**Figure C.3:** Découpage d'un nuage de points en KD-tree, par coupe médiane. Les hyperplans de coupe se trouvent sur cet exemple entre le point médian et le suivant. Les chiffres indiquent le niveau de la subdivision à laquelle ils sont associés. Les coupes alternent hyperplans verticaux (niveaux pairs) et horizontaux (niveaux impairs).

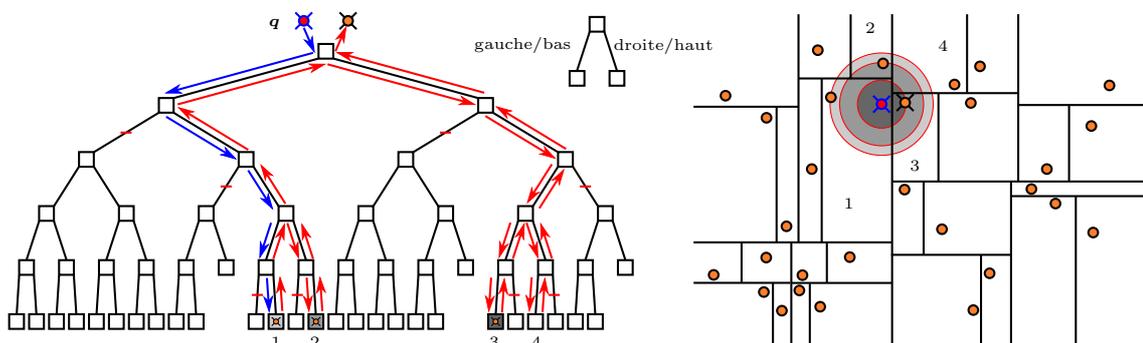
## C.2 Requête de plus proche voisin

La recherche du plus proche voisin d'un point requête  $q$  dans un arbre de subdivision peut s'effectuer en un nombre d'opérations dépendant de la profondeur de l'arbre, qui est en général bien inférieure à la taille du nuage de points. Cette recherche s'effectue en deux temps:

1. plongée dans l'arbre pour identifier la cellule la plus profonde contenant le point  $q$ . On recherche ensuite le point de cette cellule qui soit le plus proche de  $q$ . Ce point est potentiellement le point du nuage qui est effectivement le plus proche de  $q$ , mais cette hypothèse doit être vérifiée et corrigée si elle s'avère fausse (phase 2).
2. remontée dans l'arbre pour tester les branches non parcourues, et vérifier qu'elles ne peuvent pas contenir de point plus proche que le candidat courant. Dans le cas contraire, il est nécessaire d'explorer les branches (plongée dans la branche) pour mettre à jour le candidat.

Cette démarche permet de ne tester que les cellules dont les limites recoupent le voisinage optimal obtenu jusqu'alors, puisque seules ces cellules sont en mesure de fournir un point plus proche que le candidat optimal. Cet événement est en général très rare, de sorte que dans le cas moyen, seuls quelques branches sont effectivement explorées en plus de la branche initiale (phase de plongée).

La méthode peut être aisément étendue de manière à gérer le retour d'une liste des  $k$  points les plus proches de  $q$  au lieu d'un unique point.



**Figure C.4:** Exemple de recherche de plus proche voisin dans un KD-tree. Le processus plonge tout d'abord dans l'arbre (flèches bleues) et se positionne dans la cellule 1 contenant le point requête  $q$ . Le point de cette cellule définit un premier voisinage (gris clair), dans lequel peuvent se trouver d'autres points. En remontant (flèches rouge), on constate qu'une première branche peut fournir un point plus proche. La vérification nous conduit à la cellule 2 contenant effectivement un point plus proche, que l'on retient avec le nouveau voisinage correspondant (gris intermédiaire). Une nouvelle vérification après être remonté nous conduit à la cellule 3 contenant cette fois-ci le point le plus proche de  $q$ , puisque l'exploration de la cellule 4 n'est pas concluante.

## C.3 Voisinage d'une primitive

Les arbres de subdivision spatiale peuvent en outre être utilisés pour permettre la recherche rapide de points se trouvant à l'intérieur d'une primitive donnée, si l'on sait tester efficacement l'intersection entre la primitive requête et les volumes englobant les cellules de subdivision.

Cette recherche s'effectue simplement en éliminant, lors de la plongée dans l'arbre, toutes les cellules dont le volume englobant est disjoint de la primitive requête, puisque de telles cellules ne peuvent héberger de points à l'intérieur de la primitive. A la fin de ce processus de plongée, on obtient ainsi une liste de feuilles contenant potentiellement des points à l'intérieur de la primitive. Il est alors nécessaire de tester la totalité de ces points pour ne retenir que ceux qui satisfont la requête.

Le voisinage d'une primitive peut ainsi être obtenu en la faisant enfler de la taille du voisinage souhaité, et cherchant les points à l'intérieur de cette primitive grossie. Parmi ces points, il est ensuite nécessaire de filtrer ceux qui se trouvent trop éloignés de la surface.

# Bibliography

- [AA03] Anders Adamson and Marc Alexa. Approximating and intersecting surfaces from points. In *Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 230–239. Eurographics Association, 2003. 54
- [ABCO<sup>+</sup>01] Marc Alexa, Johannes Behr, Daniel Cohen-Or, Shachar Fleishman, David Levin, and Claudio T Silva. Point Set Surfaces. *Proceedings of the conference on Visualization*, pages 21–28, 2001. 30, 38
- [ACK01] Nina Amenta, Sunghee Choi, and R.K. Kolluri. The power crust. In *Proceedings of the sixth ACM symposium on Solid modeling and applications*, pages 249–266. ACM, 2001. 54
- [AF06] Marco Attene and Bianca Falcidieno. Hierarchical mesh segmentation based on fitting primitives. *The Visual Computer*, 22(3):181–193, February 2006. 43
- [Ahn08] Sung Joon Ahn. Geometric Fitting of Parametric Curves and Surfaces. *Journal of Information Processing Systems*, 2008. 36
- [All06] Rémi Allègre. *Contributions à l'introduction de flexibilité dans la reconstruction et l'édition de modèles 3D*. PhD thesis, Université Claude Bernard Lyon 1, 2006. 53, 54
- [AMCO08] Dror Aiger, N.J. Mitra, and D. Cohen-Or. 4-points congruent sets for robust pairwise surface registration. In *ACM Transactions on Graphics (TOG)*, volume 27, page 85. ACM, 2008. 29
- [AS00] Marco Attene and Michela Spagnuolo. Automatic Surface Reconstruction from Point Sets in Space. *Computer Graphics Forum*, 19(3):457–465, September 2000. 54
- [AS05] I Al-Subaihi. Algebraic fitting of quadric surfaces to data. *Communications in Applied Analysis*, 2005. 34, 76
- [B12] Roseline Bénière. *Reconstruction d'un modèle B-Rep à partir d'un maillage 3D*. PhD thesis, 2012. 42, 56
- [Bau72] Bruce G. Baumgart. Winged edge polyhedron representation. October 1972. 15

- [BC00] Jean-Daniel Boissonnat and Frédéric Cazals. Smooth surface reconstruction via natural neighbour interpolation of distance functions. *of the sixteenth annual symposium on*, 2000. 55
- [BDZ09] C Benedek, X Descombes, and J Zerubia. Building extraction and change detection in multitemporal aerial and satellite images in a joint stochastic approach. Technical Report October, 2009. 111
- [Ben75] Jon Louis Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517, September 1975. 193
- [Ben00] R Benlamri. Range image segmentation of scenes with occluded curved objects. *Pattern Recognition Letters*, 21(12):1051–1060, November 2000. 41, 44
- [BF06] Christian Beder and W. Forstner. Direct solutions for computing cylinders from minimal sets of 3D points. *Lecture Notes in Computer Science*, 3951:135, 2006. 46
- [BJ88] P.J. Besl and R.C. Jain. Segmentation through variable-order surface fitting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(2):167–192, March 1988. 42, 43
- [BLGTK07] H Boulaassal, T Landes, P Grussenmeyer, and F Tarsha-Kurdi. Automatic Segmentation of Building Façades using Terrestrial Laser Data. *The International Archives of Photogrammetry and Remote Sensing*, XXXVI:65–70, 2007. 43, 48
- [BM92] P.J. Besl and N.D. McKay. A method for registration of 3-D shapes. *IEEE Transactions on pattern analysis and machine intelligence*, 14(2):239–256, 1992. 29
- [BM99] F Bernardini and J Mittleman. The Ball-pivoting algorithm for surface reconstruction. *Visualization and*, 1999. 54
- [BMV01] Pal Benko, Ralph R Martin, and Tamas Varady. Algorithms for Reverse Engineering Boundary Representation Models. *Computer Aided Design*, 33:839–851, 2001. 44, 45, 55
- [Bos10] Frédéric Bosché. Automated Recognition of 3D CAD Model Objects and Calculation of As-built Dimensions for Dimensional Compliance Control in Construction. *Elsevier Journal of Advanced Engineering Informatics*, 2010. 58, 69
- [BR05] F. Bretar and M. Roux. Hybrid Image Segmentation using LIDAR 3D Planar Primitives. In *ISPRS Laserscanning*, volume 6, pages 12–14, Enschede, 2005. 48
- [Bro98] S. Brooks. Markov chain Monte Carlo method and its application. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 47(1):69–100, 1998. 106
- [BT93] Dimitris Bertsimas and John Tsitsiklis. Simulated annealing. *Statistical Science*, 8(1):10–15, 1993. 104, 108, 114

- [BWK02] Mario Botsch, Andreas Wiratanaya, and Leif Kobbelt. Efficient High Quality Rendering of Point Sampled Geometry. In *Proceedings of the 13th Eurographics workshop on Rendering*, pages 53–64, 2002. 193
- [Cai87] R. Cairolì. *Algèbre linéaire (tomes I & II)*. Presses polytechniques romandes, 1987. 35
- [CC78] E. Catmull and J. Clark. Recursively generated B-spline surfaces on arbitrary topological meshes. *Computer-Aided Design*, 10(6):350–355, November 1978. 21
- [CG95] S. Chib and E. Greenberg. Understanding the metropolis-hastings algorithm. *The American Statistician*, 49(4):327–335, 1995. 106, 143
- [CG01] Thomas Chaperon and François Goulette. Extracting cylinders in full 3D data using a random sampling method and the Gaussian image. *Vision, Modeling and Visualization*, 2001. 45, 51
- [CG04] Frédéric Cazals and Joachim Giesen. Delaunay Triangulation Based Surface Reconstruction: Ideas and Algorithms. Technical Report November, 2004. 54
- [Cha00] Raphaëlle Chaine. *segmentation d'ensembles non organisés de points 3D d'une surface : propagation anisotrope d'étiquettes basée sur les graphes*. PhD thesis, Université Claude Bernard Lyon 1, 2000. 39, 42
- [Cha02] Thomas Chaperon. *Segmentation de nuages de points pour la modélisation automatique d'environnements industriels numérisés*. PhD thesis, Ecole des mines de Paris, 2002. 36, 37, 56, 57, 58, 61, 98, 123, 149
- [Cha03] R. Chaine. A geometric convection approach of 3-D reconstruction. In *Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 218–229. Eurographics Association, 2003. 54
- [CL09] Chia-Ming Cheng and Shang-Hong Lai. A consensus sampling technique for fast and robust model fitting. *Pattern Recognition*, 42(7):1318–1329, July 2009. 48
- [CLP10] A.L. Chauve, Patrick Labatut, and J.P. Pons. Robust piecewise-planar 3D reconstruction and completion from large-scale unstructured point data. *Computer Vision and Pattern Recognition, 2010*, 2010. 57
- [CM91] Yang Chen and Gérard Medioni. Object modelling by registration of multiple range images. In *Proceedings of IEEE International Conference on Robotics and Automation*, 1991. 29
- [CS07] C Chen and I Stamos. Range Image Segmentation for Modeling and Object Detection in Urban Scenes. *The 6th International Conference on 3-D Digital Imaging and Modeling*, 2007. 39, 42, 43, 45
- [CSD04] D. Cohen-Steiner and F. Da. A greedy Delaunay-based surface reconstruction algorithm. Technical Report 1, 2004. 54

- [DD04] Florent Duguet and George Drettakis. Flexible Point-Based Rendering on Mobile Devices. *IEEE Computer Graphics and Applications*, 24(4):57–63, 2004. [193](#)
- [Des04] Xavier Descombes. *Méthodes stochastiques en analyse d'image: des champs de Markov aux processus ponctuels marqués*. PhD thesis, 2004. [111](#)
- [DH72] R.O. Duda and P.E. Hart. Use of the Hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 15(1):11–15, 1972. [49](#)
- [DJ94] MP Dubuisson and A K Jain. A Modified Hausdorff Distance for Object Matching. *Pattern Recognition*, 1:566–568, 1994. [82](#)
- [DLP08] Rodrigo De Toledo, Bruno Lévy, and Jean-Claude Paul. Reverse Engineering For Industrial-Environment CAD Models. *Tools and Methods for Competitive Engineering*, 2008. [34](#), [58](#)
- [DMDV11] Jérôme Demantké, Clément Mallet, Nicolas David, and Bruno Vallet. Dimensionality based scale selection in 3d lidar point clouds. *ISPRS Workshop on Laser Scanning*, 2011. [30](#)
- [DMZ08] Xavier Descombes, Robert Minlos, and Elena Zhizhina. Object Extraction Using a Stochastic Birth-and-Death Dynamics in Continuum. *Journal of Mathematical Imaging and Vision*, 33(3):347–359, October 2008. [111](#), [113](#), [125](#), [139](#)
- [DTB06] J.R. Diebel, Sebastian Thrun, and M. Brünig. A Bayesian method for probable surface reconstruction and decimation. *ACM Transactions on Graphics (TOG)*, 25(1):39–59, 2006. [57](#), [59](#), [72](#)
- [DZ02] X Descombes and J Zerubia. Marked point process in image analysis. *Signal Processing Magazine, IEEE*, 2002. [111](#)
- [Ede03] Herbert Edelsbrunner. Surface reconstruction by wrapping finite sets in space. *Discrete and Computational Geometry*, 25:379–404, 2003. [53](#)
- [FB87] Martin A Fischler and Robert C Bolles. Random Sample Consensus : a Paradigm for Model Fitting with applications to Image Analysis and Automated Cartography. *Readings in computer vision: issues, problems, principles, and paradigms*, pages 726–740, 1987. [46](#), [48](#), [75](#)
- [FCOS05] Shachar Fleishman, D. Cohen-Or, and C.T. Silva. Robust moving least-squares fitting with sharp features. In *ACM Transactions on Graphics*, volume 24, pages 544–552. ACM, 2005. [55](#)
- [FEF97] Andrew W Fitzgibbon, David W Eggert, and Robert B Fisher. High-level CAD Model Acquisition From Range Image. *Computer-Aided Design*, 1997. [34](#)
- [Fis04] R.B. Fisher. Applying knowledge to reverse engineering problems. *Computer-Aided Design*, 36(6):501–510, 2004. [55](#), [57](#), [58](#)

- [FMA<sup>+</sup>10] Alfredo Ferreira, Simone Marini, Marco Attene, M.J. Fonseca, Michela Spagnuolo, J.A. Jorge, and Bianca Falcidieno. Thesaurus-based 3d object retrieval with part-in-whole matching. *International Journal of Computer Vision*, 89(2):327–347, June 2010. 52
- [FO09] Takahiko Furuya and Ryutarou Ohbuchi. Dense sampling and fast encoding for 3d model retrieval using bag-of-visual features. In *Proceeding of the ACM International Conference on Image and Video Retrieval*, page 26. ACM, 2009. 52
- [FS06] T Funkhouser and P Shilane. Partial matching of 3D shapes with priority-driven search. *Proceedings of the fourth Eurographics symposium on Geometric Processing*, 2006. 52
- [GCO06] Ran Gal and Daniel Cohen-Or. Salient geometric features for partial shape matching and similarity. *ACM Transactions on Graphics*, 25(1):130–150, January 2006. 52
- [GG04] Natasha Gelfand and Leonidas J Guibas. Shape Segmentation Using Local Slippage Analysis. *Eurographics Symposium on Geometry Processing*, 2004. 42, 43
- [GH09] Peter J. Green and David I. Hastie. Reversible Jump MCMC, July 2009. 108
- [GM06] Daniel Girardeau-Montaut. *Détection de changement sur des données géométriques tridimensionnelles*. PhD thesis, Ecole nationale supérieure des télécommunications, Paris, 2006. 37, 67, 96, 171, 193
- [Gor07] Ben Gorte. Planar Feature Extraction in Terrestrial Laser Scans using Gradient Based Range Image Segmentation. *The International Archives of Photogrammetry and Remote Sensing*, XXXVI:173–177, 2007. 43
- [Gre95] Peter J. Green. Reversible Jump Markov Chain Monte Carlo Computation and Bayesian Model Determination. *Biometrika*, 82(4):711, December 1995. 108
- [GS85] Leonidas Guibas and Jorge Stolfi. Primitives for the manipulation of general subdivisions and the computation of Voronoi. *ACM Transactions on Graphics*, 4(2):74–123, April 1985. 16
- [GSH<sup>+</sup>07] Ran Gal, Ariel Shamir, Tal Hassner, Mark Pauly, and Daniel Cohen-Or. Surface Reconstruction using Local Shape Priors. *Proceedings of the fifth Eurographics symposium on Geometry processing*, pages 253–262, 2007. 57
- [HDD<sup>+</sup>92] Hugues Hoppe, Tony DeRose, Tom Duchamp, John McDonald, and Werner Stuetzle. Surface Reconstruction From Unorganized Points. *Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, 1992. 30
- [HGLT11] Jean-François Hullo, Pierre Grussenmeyer, Tania Landes, and Guillaume Thibault. Georeferencing of tls data for industrial indoor complex scenes: beyond current solutions. In *ISPRS Laserscanning 2011*, 2011. 29

- [HJBJ<sup>+</sup>96] Adam Hoover, Gillian Jean-Baptiste, Xiaoyi Jiang, Patrick J Flynn, Horst Bunke, Dmitry B Goldgof, Kevin Bowyer, David W Eggert, Andrew Fitzgibbon, and Robert B Fisher. An Experimental Comparison Of Range Image Segmentation Algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18:673–689, 1996. [39](#)
- [Hor87] B.K.P. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4(4):629–642, 1987. [28](#)
- [Hou62] P.V.C. Hough. Method and means for recognizing complex patterns. U.S. Patent 3.069.654, December 1962. [49](#)
- [JH99] Andrew Eddie Johnson and M Hebert. Using spin images for efficient object recognition. *IEEE transactions on pattern analysis and machine intelligence*, 1999. [52](#), [81](#)
- [JWB<sup>+</sup>06] P Jenke, M Wand, M Bokeloh, A Schilling, and W. Straß er. Bayesian point cloud reconstruction. *Computer Graphics Forum*, 25(3):379–388, 2006. [57](#), [59](#), [72](#)
- [KBK<sup>+</sup>04] S.W. Kwon, F. Bosche, C. Kim, C.T. Haas, and K.A. Liapi. Fitting range data to primitives for rapid local 3D modeling using sparse range point clouds. *Automation in Construction*, 13(1):67–81, 2004. [58](#)
- [Ker09] Barna Keresztes. *Segmentation De Zones D'intérêt Par Processus Objets : Application A La Détection Des Réseaux De Failles Sismiques*. PhD thesis, 2009. [111](#)
- [Kho07] Kourosh Khoshelham. Extending Generalized Hough Transform to Detect 3D Objects in Laser Range Data. *The International Archives of Photogrammetry and Remote Sensing*, XXXVI:206–210, 2007. [51](#)
- [KJD<sup>+</sup>09] Maria Kulikova, Ian Jermyn, Xavier Descombes, Josiane Zerubia, and Elena Zhizhina. A marked point process model with strong prior shape information for the extraction of multiple , arbitrarily-shaped objects. *Fifth International Conference on Signal-Image Technology and Internet-Based Systems (SITIS)*, 2009. [111](#)
- [Lev03] D Levin. Mesh-independent surface interpolation. *Geometric modeling for scientific visualization*, 2003. [37](#)
- [LGD10] Florent Lafarge, Georgy Gimel'farb, and Xavier Descombes. Geometric feature extraction by a multimarked point process. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1597–609, September 2010. [112](#), [125](#)
- [LGM<sup>+</sup>04] F C Langbein, C H Gao, B I Mills, A D Marshall, and R R Martin. Topological and Geometric Beautification of Reverse Engineered GeometricModels. *Proceedings of the 9th ACM Symposium on Solid Modeling and Applications*, pages 255–260, 2004. [55](#), [58](#)
- [LKBH10] Florent Lafarge, Renaud Keriven, M. Brédif, and V.H. Hiep. Hybrid multi-view reconstruction by jump-diffusion. In *Computer Vision and Pattern Recognition, 2010*, pages 350–357. IEEE, 2010. [11](#), [56](#), [57](#), [114](#)

- [Loo87] Charles Teorell Loop. *Smooth Subdivision Surfaces Based on Triangles*. PhD thesis, 1987. 21
- [LWC<sup>+</sup>11] Yangyan Li, Xiaokun Wu, Yiorgos Chrysathou, Andrei Sharf, D. Cohen-Or, and N.J. Mitra. GlobFit: consistently fitting primitives by discovering global relations. *ACM Transactions on Graphics*, 30(4):52, 2011. 56, 57, 58
- [Man88] M. Mantyla. *Introduction to Solid Modeling*. January 1988.
- [MAVdF05] Boris Mederos, Nina Amenta, Luiz Velho, and L.H. de Figueiredo. Surface reconstruction from noisy point clouds. In *Proceedings of the third Eurographics symposium on Geometry processing*, page 53. Eurographics Association, 2005. 54
- [MGP06] Niloy J Mitra, Leonidas J Guibas, and Mark Pauly. Partial and Approximate Symmetry Detection for 3D Geometry. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 2006. 52
- [MLB<sup>+</sup>09] C Mallet, F Lafarge, F Bretar, M Roux, U Soergel, and C Heipke. A stochastic approach for modelling airborne lidar waveforms. In *ISPRS Laserscanning 2009*, pages 201 – 206, 2009. 26
- [MLM01] David Marshall, Gabor Lukacs, and Ralph Martin. Robust Segmentation of Primitives from Range Data in the Presence of Geometric Degeneracy. *IEEE PAMI*, 23:304–314, 2001. 36, 43
- [MLMM01] B I Mills, F C Langbein, A D Marshall, and R R Martin. Approximate Symmetry Detection For Reverse Engineering. *Proceedings of Sixth ACM Symposium on Solid Modelling and Applications*, pages 241–248, 2001. 52
- [MN04] Niloy J Mitra and An Nguyen. Estimating Surface Normals in Noisy Point Cloud Data. *Special Issue of International Journal of Computational Geometry and Applications*, 14:261–276, 2004. 30
- [MVD03] B. Mederos, L. Velho, and L.H. De Figueiredo. Moving least squares multiresolution surface approximation. In *Computer Graphics and Image Processing, 2003. SIBGRAPI 2003. XVI Brazilian Symposium on*, pages 19–26. IEEE, 2003. 54
- [NSZCO10] Liangliang Nan, A Sharf, H Zhang, and D Cohen-Or. Smartboxes for interactive urban reconstruction. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 29(4):Article 93, 2010. 48, 52, 58
- [OBA<sup>+</sup>03] Yutaka Ohtake, Alexander Belyaev, Marc Alexa, Greg Turk, and Hans-Peter Seidel. Multi-Level Partition of Unity Implicits. *Proceedings of ACM SIGGRAPH 2003*, 2003. 55
- [OBKI04] Jens Overby, Lars Bodum, Erik Kjems, and PM Iisoe. Automatic 3D building reconstruction from airborne laser scanning and cadastral data using Hough transform. In *ISPRS Congress*,, 2004. 51
- [OBS03] Yutaka Ohtake, Alexander Belyaev, and H.P. Seidel. A multi-scale approach to 3D scattered data interpolation with compactly supported basis functions. In *Shape Modeling International, 2003*, pages 153–161. IEEE, 2003. 54, 55

- [OBS06] Yutaka Ohtake, Alexander Belyaev, and H.P. Seidel. A composite approach to meshing scattered data. *Graphical Models*, 68(3):255–267, 2006. 55
- [ODJ03] Mathias Ortner, Xavier Descombes, and J. Building extraction from digital elevation models. Technical report, 2003. 106, 111
- [Ort04] Mathias Ortner. *Processus ponctuels marqués pour l'extraction automatique de caricatures de bâtiments à partir de modèles numériques d'élévation*. PhD thesis, 2004. 111
- [OTDS04] Kazuo Oda, Tadashi Takano, Takeshi Doihara, and Ryosuke Shibasaki. Automatic Building Extraction and 3-D City Modeling from Lidar Data Based Hough Transformation. In *IAPRS*, pages 3–6, 2004. 51
- [PB06] Norbert Pfeifer and Christian Briese. Laser Scanning, Principles and Applications. Technical report, 2006. 25
- [PKKG03] Mark Pauly, Richard Keiser, Leif P Kobbelt, and Markus Gross. Shape Modeling with Point-Sampled Geometry. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 2003. 54
- [PMG<sup>+</sup>05] Mark Pauly, N.J. Mitra, Joachim Giesen, Markus Gross, and L.J. Guibas. Example-based 3D scan completion. *Proceedings of the third Eurographics symposium on Geometry processing*, page 23, 2005. 33
- [PP03] M Peternell and H Pottmann. Hough transform and Laguerre geometry for the recognition and reconstruction of special 3D shapes. *Discrete Mathematics and Geometry*, pages 1–31, 2003. 51
- [PTVF92] William H Press, Saul A Teukolsky, Williauw T Vetterling, and Brian P Flannery. *Numerical Recipes in C : The Art of Scientific Computing*. Cambridge University Press, second edi edition, 1992. 36
- [Rab06] Tahir Rabbani. *Automatic Reconstruction of Industrial Installations Using Images and Point Clouds*. PhD thesis, University of Engineering and Technology, Lahore, Pakistan, 2006. 51
- [RDvV07] Tahir Rabbani, Sander Dijkman, Frank van Den Heuvel, and George Vosselman. An integrated Approach for Modelling and Registration of Point Clouds. *ISPRS Journal of Photogrammetry & Remote Sensing*, 61:355–370, 2007. 29
- [Ric73] A Ricci. A Constructive Geometry for Computer Graphics. *The Computer Journal*, 16(2):157–160, 1973. 18
- [RV05] Tahir Rabbani and F. Van Den Heuvel. Efficient hough transform for automatic detection of cylinders in point clouds. *ISPRS WG III/3, III/4*, 3:60–65, 2005. 51, 57, 58
- [RvV06] T Rabbani, F. van Den Heuvel, and G. Vosselmann. Segmentation of point clouds using smoothness constraint. *International archives of photogrammetry, remote sensing and spatial information sciences*, 36(5):248–253, 2006. 42, 43

- [SD01] A.D. Sappa and Michel Devy. Fast range image segmentation by an edge detection strategy. In *Proceedings of the Third International Conference on 3-D Digital Imaging and Modeling*, pages 292–299. IEEE, 2001. 44
- [SDK09] Ruwen Schnabel, Patrick Degener, and Reinhard Klein. Completion and Reconstruction with Primitive Shapes. *Computer Graphics Forum (Proc. of Eurographics)*, 28(2):503–512, 2009. 33
- [SF05] CE Scheidegger and Shachar Fleishman. Triangulating point set surfaces with bounded error. *Proceedings of the third Symposium on Geometry Processing*, pages 63–72, 2005. 54
- [SLMT09] S Soudarissanane, R Lindenbergh, M Menenti, and P Teunissen. Incidence angle influence on the quality of Terrestrial Laser Scanning points. In *ISPRS Laserscanning 2009*, pages 183 – 188, 2009. 26
- [SLS<sup>+</sup>06] A. Sharf, T. Lewiner, A. Shamir, L. Kobbelt, and D. Cohen-Or. Competing fronts for coarse-to-fine surface reconstruction. In *Computer Graphics Forum*, volume 25, pages 389–398. Wiley Online Library, 2006. 55
- [SP10] Jörg Schmittwilken and Lutz Plümer. Model-based reconstruction and classification of facade parts in 3d point clouds. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 38(1):269–274, 2010. 57
- [SPOK95] V.V. Savchenko, A.A. Pasko, O.G. Okunev, and T.L. Kunii. Function representation of solids reconstructed from scattered surface points and contours. In *Computer Graphics Forum*, volume 14, pages 181–188. Wiley Online Library, 1995. 55
- [SRML09] Xianfang Sun, P.L. Rosin, R.R. Martin, and F.C. Langbein. Noise analysis and synthesis for 3D laser depth scanners. *Graphical Models*, 71(2):34–48, 2009. 31
- [SSS08] S Shalom, L Shapira, and A Shamir. Part analogies in sets of objects. In *Eurographics Workshop on 3D Object Retrieval*, pages 1–8, 2008. 52
- [Ste00] Matthew Stephens. Bayesian analysis of mixture models with an unknown number of components—an alternative to reversible jump methods. *The Annals of Statistics*, 28(1):40–74, 2000. 110
- [SWK07] Ruwen Schnabel, Roland Wahl, and Reinhard Klein. Efficient RANSAC for Point-Cloud Shape Detection. *Computer Graphics Forum*, 26(2):214–226, 2007. 43, 47, 48, 49, 57, 58, 59, 75, 139, 140, 143, 147, 148
- [SWWK08] Ruwen Schnabel, Raoul Wessel, Roland Wahl, and Reinhard Klein. Shape recognition in 3d point-clouds. Technical report, Institut für Informatik II, Universität Bonn, 2008. 55
- [Tau91] G. Taubin. Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(11):1115–1138, 1991. 43

- [TKLG07] F Tarsha-Kurdi, T Landes, and P Grussenmeyer. Hough Transform and Extended Ransac Algorithms for Automatic Detection of 3D Building Roof Planes From Lidar Data. *The International Archives of Photogrammetry and Remote Sensing*, XXXVI:407–412, 2007. 43, 48, 51
- [TO02] Greg Turk and James F. O’Brien. Modelling with implicit surfaces that interpolate. *ACM Transactions on Graphics*, 21(4):1–12, 2002. 54, 55
- [TZ00] P H S Torr and A Zisserman. MLESAC: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, 2000. 75
- [USF08] T Ullrich, V Settgast, and D W Fellner. Robust Shape Fitting and Semantic Enrichment. *Journal on Computing and Cultural Heritage*, 2008. 76
- [VD01] George Vosselman and Sander Dijkman. 3D building model reconstruction from point clouds and ground plans. *International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences*, 34(3/W4):37–44, 2001. 51
- [VMC97] Tamas Varady, Ralph R Martin, and Jordan Cox. Reverse Engineering of Geometric Models - An Introduction. *Computer-Aided Design*, 29:255–268, 1997. 45
- [Wal04] B. Walsh. Markov chain monte carlo and gibbs sampling. Technical Report April, 2004. 106
- [WbFAR99] Naoufel Werghi, bob Fisher, Anthony Ashbrook, and Craig Robertson. Faithful Recovering of Quadric Surfaces from 3D Range Data by Global Fitting. *International Journal Of Shape Modeling*, 1999. 34
- [WK10] R Wessel and R Klein. Learning the Compositional Structure of Man-Made Objects for 3D Shape Retrieval. 2010. 52, 58
- [WKWL02] H Woo, E Kang, Semyung Wang, and Kwan H Lee. A new Segmentation Method for Points Cloud Data. *International Journal of Machine Tools & Manufacture*, 42:167–178, 2002. 42, 44, 45
- [WPH<sup>+</sup>04] Tim Weyrich, Mark Pauly, Simon Heinzle, Richard Keiser, Sascha Scandella, and Markus Gross. Post-processing of Scanned 3D Surface Data. *Symposium on Point-Based Graphics*, pages 85–94, 2004. 38, 44
- [YY08] Liu Yu and Xiong Youlun. Automatic segmentation of unorganized noisy point clouds based on the Gaussian map. *Computer Aided Design*, 40(5):576–594, 2008. 42, 43, 44
- [ZOF01] H.K. Zhao, Stanley Osher, and Ronald Fedkiw. Fast surface reconstruction using the level set method. In *Variational and Level Set Methods in Computer Vision, 2001. Proceedings. IEEE Workshop on*, pages 194–201. IEEE, 2001. 55