



HAL
open science

Vers une polyédrisation des objets discrets bruités 3D

Laurent Provot

► **To cite this version:**

Laurent Provot. Vers une polyédrisation des objets discrets bruités 3D. Autre [cs.OH]. Université Henri Poincaré - Nancy 1, 2009. Français. NNT : 2009NAN10119 . tel-01748301

HAL Id: tel-01748301

<https://hal.univ-lorraine.fr/tel-01748301>

Submitted on 29 Mar 2018

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

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



AVERTISSEMENT

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire élargie.

Il est soumis à la propriété intellectuelle de l'auteur. Ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document.

D'autre part, toute contrefaçon, plagiat, reproduction illicite encourt une poursuite pénale.

Contact : ddoc-theses-contact@univ-lorraine.fr

LIENS

Code de la Propriété Intellectuelle. articles L 122. 4

Code de la Propriété Intellectuelle. articles L 335.2- L 335.10

http://www.cfcopies.com/V2/leg/leg_droi.php

<http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm>

Vers une polyédrisation des objets discrets bruités 3D

THÈSE

présentée et soutenue publiquement le 2 décembre 2009

pour l'obtention du

Doctorat de l'université Henri Poincaré – Nancy 1
(spécialité Informatique)

par

Laurent PROVOT

Composition du jury

<i>Rapporteurs :</i>	Achille BRAQUELAIRE, Professeur des Universités Jean-Marc CHASSERY, Directeur de Recherche CNRS	LaBRI, Bordeaux GIPSA-lab, Grenoble
<i>Examineurs :</i>	Grégoire MALANDAIN, Directeur de Recherche INRIA Adam CICHON, Professeur des Universités	Sophia-Antipolis Loria, Nancy
<i>Directrice de thèse :</i>	Isabelle DEBLED-RENNESSON, Maître de conférences (HDR)	Loria, Nancy

Table des matières

Introduction générale	1
1 Notions de base en géométrie discrète	5
1.1 Espaces discrets	5
1.2 Voisinages	6
1.3 Objets discrets	6
1.4 Courbes et surfaces discrètes	8
1.5 Structures de données pour l'espace discret	10
1.6 Discrétisation	13
1.7 Primitives discrètes linéaires	14
1.7.1 Droite discrète arithmétique	14
1.7.2 Plan discret arithmétique	15
1.8 Algorithmes de reconnaissance de primitives discrètes linéaires	16
1.8.1 Approche arithmétique	16
1.8.2 Reconnaissance par dualité	17
1.8.3 Approche géométrique	19
1.9 Synthèse	22
2 Un panorama des méthodes de polyédration en géométrie discrète	23
2.1 Introduction	23
2.2 Méthodes produisant un maillage triangulaire du bord de l'objet	24
2.2.1 Méthodes se basant sur des configurations locales	24
2.2.2 Méthodes basées sur un « Diagramme de Voronoï discret »	26
2.3 Méthodes réversibles se basant sur des primitives discrètes	32
2.3.1 Polyédration en polyèdre discret standard	33
2.3.2 Méthodes exploitant la préimage	35
2.3.3 Reconstruction analytique inversible	42
2.4 Synthèse sur la polyédration en géométrie discrète	45
3 Primitives discrètes floues	47
3.1 Introduction	47
3.2 Segments flous de droites discrètes	48
3.3 Morceaux flous de plans discrets	51

3.4	Reconnaissance de morceaux flous de plans discrets	53
3.4.1	Approche par programmation linéaire	53
3.4.2	Une approche géométrique	59
3.4.3	Adaptation de méthodes existantes	63
3.5	Résultats expérimentaux	64
3.5.1	Morceaux de plans discrets	65
3.5.2	Morceaux de sphères discrètes	67
3.5.3	Morceaux de surfaces discrètes	70
3.5.4	Synthèse et discussion	72
3.6	Conclusion	73
4	Estimateurs de caractéristiques géométriques	75
4.1	Introduction	75
4.2	Estimateurs de vecteur normal	77
4.2.1	Approches par tranches	77
4.2.2	Approche multi-directionnelles	83
4.2.3	Résultats	88
4.3	Estimateur de forme	93
4.3.1	Notion de courbure en géométrie différentielle	96
4.3.2	Estimation de la courbure sur des surfaces discrètes	97
4.3.3	Approche par tranches	101
4.3.4	Approche par patches discrets d'épaisseur ν	102
4.4	Conclusion	107
5	Segmentation d'objets discrets bruités et applications	109
5.1	Introduction	109
5.2	Segmentation du bord d'un objet discret bruité	110
5.2.1	Choix de la graine	110
5.2.2	Algorithme de reconnaissance	111
5.2.3	Critères de bonne formation	111
5.2.4	Développement des MFPD $_{\nu}$	111
5.3	Lissage	117
5.4	Vers la polyédrisation d'objets discrets bruités 3D	121
5.4.1	Lissage d'objets discrets bruités non convexes	121
5.4.2	Idées pour des perspectives de polyédrisation	123
5.5	Conclusion	124
	Conclusion et perspectives	127
1	Contributions	127
2	Perspectives	128
	Annexes	131

A Perturbation d'objets discrets	131
A.1 Perturbations aléatoires	131
A.2 Modèle de Kanungo	132
Bibliographie	135

Introduction générale

Il est un fait difficilement contestable de nos jours : l'information numérique occupe une place prépondérante dans notre société. Avec l'essor de l'informatique, le développement des supports numériques et les performances de calcul toujours plus impressionnantes repoussent inmanquablement les limites de l'envisageable. Et les conjectures de Moore nous promettent encore un bel avenir de ce côté.

L'imagerie numérique, en tant que sous-ensemble de l'information numérique, n'échappe évidemment pas à cette mouvance. Son usage s'est multiplié dans notre quotidien ces dernières années, et les fruits de cette technologie nous sont de plus en plus facilement accessibles. L'exemple le plus parlant est très probablement celui des photographies, prises sur notre appareil numérique ultra compact, mais pourtant toujours plus précis, et exploitées dans notre logiciel de retouche d'images, en appliquant des traitements des plus étonnants. Si cette dernière utilisation relève plutôt du cadre des loisirs, il est des domaines pour lesquels les techniques d'imagerie sont devenues indispensables. L'analyse d'images en est un. L'étude automatique des images satellites a, par exemple, permis la démocratisation de la cartographie numérique. Mais les applications ne s'arrêtent pas à l'espace 2D. En médecine, notamment, l'imagerie tridimensionnelle apporte un moyen d'examiner les structures internes du corps humain, et de détecter des pathologies avant même d'en observer les signes extérieurs, et ce, bien souvent, en totale innocuité.

On comprend alors rapidement, au vu de ces enjeux, que l'exploitation de ce type d'images ne peut se faire que dans un cadre rigoureux permettant de capter et de contrôler les propriétés de ces dernières et de leur support. L'une des théorisations possibles est appelée *géométrie discrète* et c'est dans cette dernière que s'inscrit cette thèse : géométrie, car étudiant des formes et leur propriétés dans l'espace ; discrète, de par la nature non continue des données manipulées. Une autre possibilité aurait été de s'appuyer sur une géométrie bien connue et déjà bien développée : la géométrie euclidienne. Néanmoins, les notions de *nombres réels* et de *continuité* sous-jacentes à cette dernière mènent à des théorèmes qui ne sont plus vrais lorsqu'on considère un espace discret. Un exemple bien connu est celui de l'intersection entre deux droites. Alors qu'un théorème de géométrie euclidienne stipule que deux droites non parallèles du plan sont concourantes en un point unique, dans un espace discret cette affirmation est quelque peu chamboulée, comme l'illustre la figure 1. Un des objectifs de la géométrie discrète est donc de définir un cadre formel pour l'étude d'images discrètes, en essayant malgré tout de conserver une analogie, dans les concepts développés, avec la géométrie du continu.

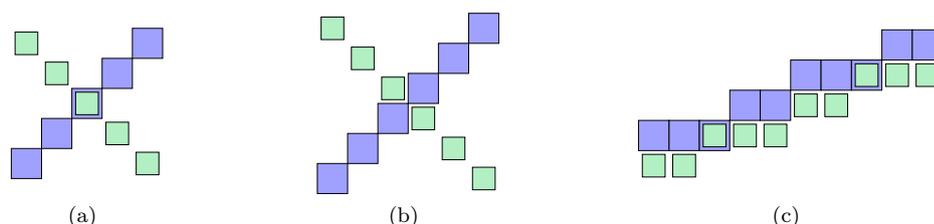


Figure 1 – Différents type d'intersection entre des droites discrètes : (a) l'intersection de deux droites sécantes est composée d'un point unique (comme dans le continu) ; (b) l'intersection de deux droites sécantes est vide ; (c) deux droites parallèles ayant une intersection non vide dans le discret.

De nombreuses études ont déjà été menées au sein de la géométrie discrète, pour en définir l'espace de travail ainsi que les objets fondamentaux manipulés et en saisir leurs propriétés. Des algorithmes de reconnaissance pour ces primitives discrètes ont été développés et utilisés dans des problèmes comme la reconnaissance de formes, l'extraction de caractéristiques géométriques et bien d'autres encore. Néanmoins, la majorité des études ont été effectuées en se reposant sur la régularité des structures fondamentales de l'espace discret, souvent issues de définitions arithmétiques, et ces critères de régularité sont généralement essentiels aux différents algorithmes développés. Or, en pratique, les objets manipulés sont très souvent bruités par les méthodes d'acquisition (scanners, IRM, ...) qui suppriment ce caractère régulier des objets. Les techniques jusqu'alors développées sont susceptibles de ne plus fournir de résultats satisfaisants. Il semble donc intéressant de poser de nouvelles définitions, destinées à apporter plus de flexibilité dans les traitements, afin de proposer des algorithmes capables de fournir des résultats satisfaisants aussi bien sur des objets réguliers que non réguliers.

En conservant une approche arithmétique, l'étude de primitives discrètes pouvant apporter une telle souplesse a débuté en 2003, avec l'introduction, par I. Debled-Rennesson et ses coauteurs [DRRRD03], de la notion de segment flou de droite discrète. Motivés par les résultats non satisfaisants obtenus lors de la polygonalisation de courbes discrètes bruitées (montrant une sur-segmentation dans certaines zones paraissant pourtant intuitivement plates, voir figure 3.1 page 48 par exemple), les auteurs ont proposé une approche reposant sur une décomposition en segments discrets d'épaisseur variable. Un paramètre ν , définissant l'épaisseur maximale souhaitée, permet de contrôler la tolérance aux perturbations de cette primitive. Des résultats encourageants ayant été obtenus en 2D, cette thèse fait suite à ces travaux en proposant une généralisation de cette approche, dans le cadre de la décomposition d'objets discrets tridimensionnels en morceaux de plans discrets. Notre objectif est d'apporter plus de souplesse à la phase de segmentation afin de décomposer les zones plates d'objets discrets en un nombre réduit de grands morceaux de plans.

Le problème de la polyédricisation d'objets discrets a été choisi comme fil conducteur de cette thèse et les chapitres de ce manuscrit s'enchaînent de manière à refléter la démarche adoptée pour résoudre cette problématique. Ainsi, le chapitre 1 est dédié à la présentation des éléments fondamentaux de la géométrie discrète. De manière générale, nous y présentons le cadre dans lequel s'inscrit ce travail, et nous y rappelons plus précisément les notions attenantes aux primitives discrètes linéaires et à leur reconnaissance.

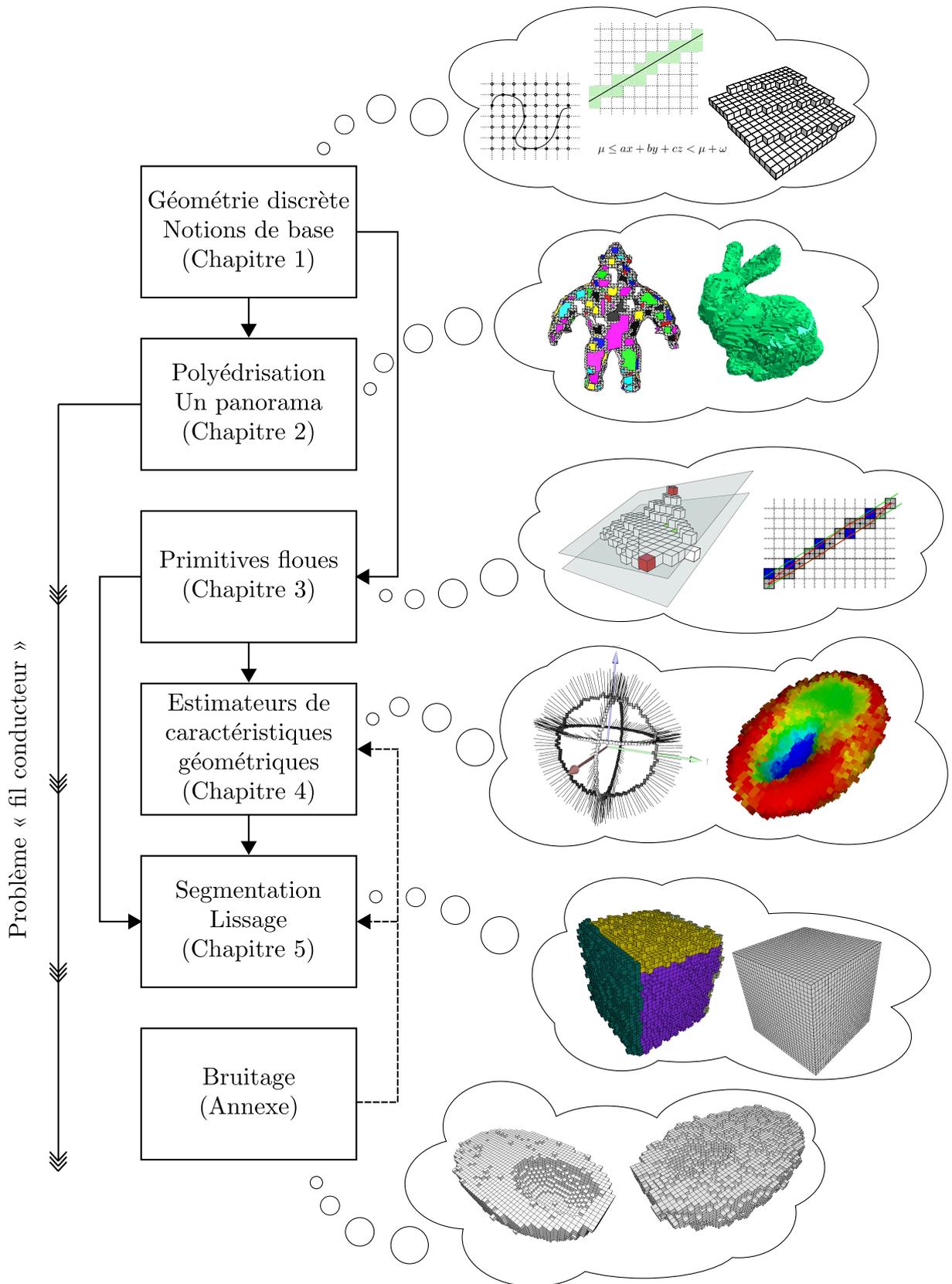
Le chapitre 2 propose un panorama des méthodes de polyédricisation en géométrie discrète. Sans toutefois être exhaustif, nous présentons de manière assez détaillée les démarches employées dans les travaux antérieurs aux nôtres sur cette problématique. L'analyse de ces travaux sera l'occasion de fixer plus précisément les objectifs de cette thèse et d'introduire les étapes nécessaires pour y arriver.

Les principales contributions de notre travail sont présentées à partir du chapitre 3, avec l'introduction des morceaux flous de plans discrets d'épaisseur ν . Cette nouvelle primitive discrète est le cœur des travaux effectués dans cette thèse. Après l'avoir définie, nous présentons différents algorithmes de reconnaissance, et effectuons une étude expérimentale pour analyser le comportement en pratique de l'un de ces algorithmes incrémentaux.

L'utilisation des morceaux flous de plans discrets est rendue effective au chapitre 4, avec pour objectif l'extraction de caractéristiques géométriques au bord d'objets discrets possiblement bruités. Nous proposons notamment des estimateurs de vecteurs normaux, que nous comparons à certains déjà existant dans le domaine, ainsi qu'un estimateur de forme permettant de distinguer les parties plates, bombées ou creuses du bord d'un objet discret.

Guidé par ces estimateurs, le chapitre 5 décrit un processus de segmentation du bord d'objets discrets en morceaux flous de plans discrets. Une première application pour le lissage d'objets bruités est ensuite exposée. Nous ne proposons malheureusement pas de solution finale au problème de la polyédricisation d'objets discrets, néanmoins nous montrons quelques résultats préliminaires et discutons les problèmes rencontrés en suggérant quelques pistes de travail.

Étant donné que les tests effectués dans nos études utilisent des objets bruités, nous discutons à l'annexe A de méthodes permettant de synthétiser des objets discrets bruités à partir d'objets initiaux généralement réguliers. Le diagramme suivant résume les connections entre les chapitres de cette thèse.



Chapitre 1

Notions de base en géométrie discrète

Sommaire

1.1	Espaces discrets	5
1.2	Voisinages	6
1.3	Objets discrets	6
1.4	Courbes et surfaces discrètes	8
1.5	Structures de données pour l'espace discret	10
1.6	Discrétisation	13
1.7	Primitives discrètes linéaires	14
1.7.1	Droite discrète arithmétique	14
1.7.2	Plan discret arithmétique	15
1.8	Algorithmes de reconnaissance de primitives discrètes linéaires	16
1.8.1	Approche arithmétique	16
1.8.2	Reconnaissance par dualité	17
1.8.3	Approche géométrique	19
1.9	Synthèse	22

Tout comme deux étrangers qui tenteraient de communiquer sans utiliser un langage commun seraient inmanquablement voués à rester incompris l'un de l'autre, il semble peu sensé de rédiger un écrit scientifique conséquent relatant son travail, en l'occurrence cette thèse, sans en fixer les notions clés du domaine, ainsi que les symboles et notations utilisés ; c'est là le but de ce chapitre.

Comme annoncé en introduction, mon travail se place dans le domaine de la *géométrie discrète*, discipline qui, bien qu'étant encore assez jeune comparée à la géométrie euclidienne, possède une théorisation déjà bien conséquente. Je ne vais donc pas tenter d'être exhaustif dans la description des notions de base de cette discipline, mais je vais me restreindre à une présentation des concepts fondamentaux, nécessaires à la bonne compréhension de ce manuscrit.

D'excellents ouvrages sont disponibles pour une introduction plus complète au domaine, parmi lesquels [CMC07], dont s'inspire en partie ce chapitre, ou encore [KR04, CM91, Coe02b, Siv04].

1.1 Espaces discrets

Avant de présenter les objets manipulés, il convient d'introduire dans un premier temps l'espace dans lequel ces derniers sont définis. De manière générale nous appelons *espace discret* de dimension d un pavage de l'espace multidimensionnel \mathbb{R}^d et *point discret* le centre de gravité de chaque cellule de ce pavage. En 2D, par exemple, l'espace discret est un pavage du plan, et seul trois polygones réguliers sont candidats pour être des cellules de ce pavage : le triangle, le carré et l'hexagone (voir figure 1.1).

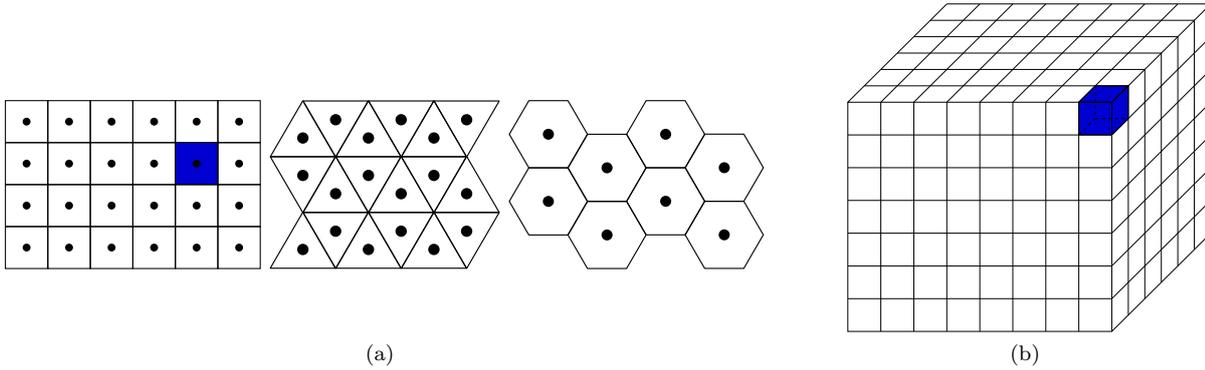


Figure 1.1 – Espaces discrets et points discrets en dimensions 2 et 3. (a) Différents pavages réguliers du plan, avec un pixel représenté en bleu. (b) Le pavage régulier de l'espace 3D, avec un voxel représenté en bleu.

Dans le cas 3D, le cube est le seul polyèdre régulier à paver l'espace. Bien que les pavages ne soient pas nécessairement réguliers, dans cette thèse nous nous limiterons à l'utilisation des pavages réguliers les plus classiques, à savoir les carrés, en 2D, ou les cubes, en 3D. Les points discrets associés à ces espaces sont alors les points de \mathbb{Z}^2 et \mathbb{Z}^3 . Les cellules de ces pavages peuvent être vues comme une représentation alternative des points discrets et sont alors communément appelées *pixels* en 2D (pour *picture elements*) et *voxels* en 3D (pour *volume elements*).

1.2 Voisinages

Les relations d'adjacence entre les points discrets peuvent se formaliser à travers la notion de *voisinage*. Elles expriment les déplacements élémentaires d'un point à un autre dans l'espace discret. Ainsi, pour le cas 2D, nous pouvons définir le *4-voisinage* comme étant la relation d'adjacence par arêtes entre deux pixels de l'espace et le *8-voisinage* comme étant la relation d'adjacence par arêtes et sommets.

Ces notions de voisinages peuvent aussi être caractérisées à partir des distances entre les points. Ainsi, en nous basant sur la distance de Manhattan $d_1(p, q) = |x_p - x_q| + |y_p - y_q|$ et la distance de l'échiquier $d_\infty(p, q) = \max(|x_p - x_q|, |y_p - y_q|)$ entre deux points $p(x_p, y_p)$ et $q(x_q, y_q)$ de \mathbb{Z}^2 , nous pouvons dire que p et q sont *4-voisins* si $d_1(p, q) = 1$, et *8-voisins* si $d_\infty(p, q) = 1$.

En dimension 3, nous pouvons de la même manière introduire les notions de *6-voisinage*, *18-voisinage* ou *26-voisinage* en considérant les adjacences par faces, arêtes et sommets, et les caractériser par les extensions naturelles en 3D des distances d_1 et d_∞ . Le tableau 1.1 résume les différentes possibilités. Nous parlerons dans la suite de manière générale de l' α -adjacence, pour $\alpha \in \{4, 8, 6, 18, 26\}$.

Il existe une dernière écriture, unifiée et aisément généralisable aux dimensions supérieures, pour exprimer les relations d'adjacence entre les points discrets. Deux points de \mathbb{Z}^d sont *r-voisins* si chaque coordonnée diffère au plus de 1 et qu'au moins r coordonnées sont égales. Dans ce formalisme, les 4- et 8-voisinages introduits précédemment se notent respectivement 1- et 0-voisinages. Et en 3D les 6-, 8- et 26-voisinages sont notés 2-, 1- et 0-voisinages respectivement. Dans la suite de ce manuscrit nous utiliserons principalement l' α -adjacence, plutôt que la r -adjacence que nous venons de présenter, car elle reste la notation la plus courante.

1.3 Objets discrets

L'espace discret étant défini et les relations d'adjacences entre les points y étant formalisées, nous décrivons à présent les objets que nous manipulerons par la suite. Les définitions qui suivent sont énoncées pour une α -adjacence donnée, $\alpha \in \{4, 8, 6, 18, 26\}$, dans les espaces \mathbb{Z}^2 et \mathbb{Z}^3 , bien que généralisables dans \mathbb{Z}^d avec la notion de r -adjacence.

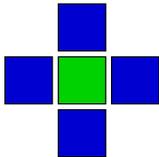
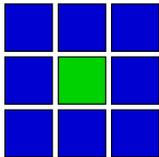
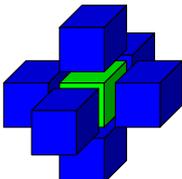
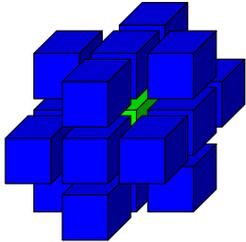
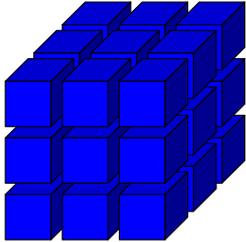
Configuration dans l'espace	Désignation	Caractérisation par distances
	4-voisinage	$d_1(p, q) = 1$
	8-voisinage	$d_\infty(p, q) = 1$
	6-voisinage	$d_1(p, q) = 1$
	18-voisinage	$d_1(p, q) \leq 2$ et $d_\infty(p, q) = 1$
	26-voisinage	$d_\infty(p, q) = 1$

Tableau 1.1 – Caractérisation des voisinages dans les espaces discrets \mathbb{Z}^2 et \mathbb{Z}^3 . Le point p est représenté en vert, et le point q représente l'un des points bleus.

Définition 1.1 (chemin α -connexe) Soit un ensemble X de points discrets et une relation de α -adjacence. Un *chemin α -connexe* (dans X) joignant deux points A et B de X est une séquence $\pi = (A_0, \dots, A_n)$ de points de X telle que $A_0 = A$, $A_n = B$ et A_i est α -voisin de A_{i-1} pour tout $i = 1, \dots, n$.

De cette notion de chemin α -connexe découle la définition d'un objet discret comme composante connexe de points discrets au sens de l' α -adjacence.

Définition 1.2 (objet discret α -connexe) Soit un ensemble X de points discrets et une relation de α -adjacence. X est un objet discret α -connexe si pour tout couple de points A et B de X , il existe un chemin α -connexe dans X joignant A et B .

Nous pouvons quelque peu contraindre la notion très générale d'un chemin α -connexe afin de définir la notion de *courbe discrète*.

Définition 1.3 (courbe α -connexe) Soit π un chemin α -connexe. π est une courbe α -connexe si, pour tous les éléments $\{A_i\}_{0 \leq i \leq n}$ de π , A_i a exactement deux points α -voisins dans π , sauf A_0 et A_n qui n'en ont qu'un (A_0 et A_n sont alors les extrémités de la courbe).

Lorsque *tous* les points de π ont exactement deux α -voisins dans π on parlera de courbe *fermée* α -connexe.

Si la définition des courbes découle assez aisément de celle d'un chemin α -connexe, ce n'est pas pour autant que dans le domaine discret ces dernières ont les propriétés naturelles attendues. C'est ce dont nous discutons brièvement dans la section suivante.

1.4 Courbes et surfaces discrètes

La définition de courbe discrète α -connexe présentée ci-dessus soulève en effet quelques problèmes topologiques. Il est généralement souhaitable que la notion de courbe fermée α -connexe dans \mathbb{Z}^2 nous permette de définir un intérieur et un extérieur à cette courbe. Cette propriété est associée en géométrie différentielle au théorème de Jordan, qui stipule qu'une courbe simple et fermée dans un sous-ensemble connexe de \mathbb{R}^2 sépare l'espace en deux composantes connexes disjointes. Or, dans l'espace discret issu d'un pavage par carrés, cette propriété n'est pas directement transposable et amène, sans considérations supplémentaires, à un paradoxe de connexité, comme le montre la figure 1.2. Si nous considérons une même relation d' α -adjacence pour toute l'image (objet et fond) alors il se peut qu'une courbe fermée 8-connexe ne sépare pas le fond en deux composantes 8-connexes, ou encore qu'une courbe fermée 4-connexe sépare le fond en plus de deux composantes 4-connexes.

Ce paradoxe est généralement levé en considérant des connexités différentes entre la courbe et le fond, soit la 4-connexité pour la courbe et la 8-connexité pour le fond, soit la 8-connexité pour la courbe et la 4-connexité pour le fond. Néanmoins cela ne résout pas tous les problèmes puisque une difficulté supplémentaire peut se poser lorsque nous considérons plusieurs courbes qui s'intersectent (comme illustré à la figure 1.3). Si nous considérons la 8-connexité pour les courbes et la 4-connexité pour le fond, alors les courbes se croisent, mais pourtant ne se séparent pas mutuellement. Nous retrouvons bien évidemment les mêmes difficultés en dimension supérieure.

Ces problèmes ont un impact direct lorsqu'il s'agit de définir une notion de surface d'un objet discret, puisque si nous désirons transposer dans le discret cette notion, il semble souhaitable qu'une telle entité possède la propriété de séparation, au sens de Jordan, permettant de dissocier d'un côté la composante décrivant l'intérieur de l'objet et de l'autre son complémentaire¹, le fond. Différents types d'approche ont été considérés pour essayer de définir une notion de surface correcte.

La première, dite *homogène*, donne une vision unifiée des surfaces et des objets, en les considérant tous deux comme un assemblage de points discrets de l'espace. Ainsi, dans [USH82], une approche de la notion de surface discrète est suggérée à travers la définition du bord d'un objet discret :

1. Pour un ensemble X de points d'un espace discret E , on appelle *complémentaire* de X dans E , et on note \bar{X} , l'ensemble des points de E qui n'appartiennent pas à X . En notation ensembliste $\bar{X} = E \setminus X$.

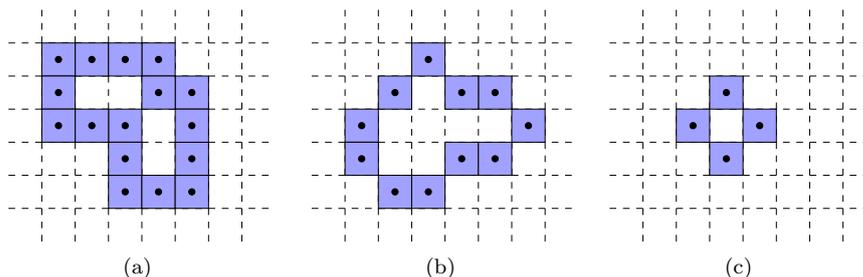


Figure 1.2 – Cas de figures pouvant induire un paradoxe de connexité invalidant le théorème de Jordan dans l’espace discret : (a) une courbe 4-connexe sépare le fond en 3 composantes 4-connexes ; (b) une courbe 8-connexe qui ne sépare pas le fond en deux composantes 8-connexes ; et (c) un ensemble de 4 pixels qui n’est pas une courbe 4-connexe mais qui pourtant sépare le fond en deux composantes 4-connexes.

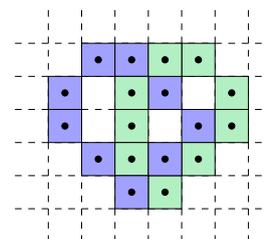


Figure 1.3 – En considérant la 8-connexité pour les courbes bleue et verte et la 4-connexité pour le fond, ces deux courbes s’intersectent sans pourtant se séparer.

Définition 1.4 (en dimension n) Un point discret de l’espace appartient au bord d’un objet discret si un de ses voisins, au sens de la connexité considérée sur le complémentaire de l’objet, n’appartient pas à l’objet discret.

Cette définition souffre des problèmes décrits plus haut, induits par les paradoxes de connexité. C’est donc en général plus une définition de frontière que de surface discrète. Il faudra donc utiliser des connexités spécifiques pour obtenir la séparation. Une autre définition de surface discrète, possédant la propriété de séparation, a été proposée dans [Mal97] et reliée à celle proposée dans [MR81].

Définition 1.5 (en dimension 3) Le sous-ensemble de voxels $S \subset \mathbb{Z}^3$ est une surface discrète si et seulement si S sépare \mathbb{Z}^3 en deux composantes 6-connexes, et tout voxel de S est 6-adjacent à chaque composante de \bar{S} .

Malgré tout, comme le soulève L. Perrotin dans [Per94], certains inconvénients persistent avec cette représentation homogène :

- la surface d’un objet diffère de la surface de son complémentaire ;
- les deux surfaces, interne et externe, que l’on aimerait obtenir pour un objet avec une cavité, peuvent être confondues si l’objet n’est pas assez épais.

C’est pourquoi une autre approche, qualifiée d’*hétérogène* a été envisagée. Elle considère pour cela une *décomposition cellulaire* de l’espace discret (cf. figure 1.4). Dans cette représentation, chaque cellule du pavage de l’espace discret est décomposée en cellules de dimensions inférieures. En dimension 2, les sommets de chaque polygone du pavage sont des cellules de dimension 0, appelés *pointels*, les arêtes des cellules de dimension 1, appelées *lignels*, et l’intérieur du polygone (ouvert) est la cellule de dimension 2 représentant le pixel lui-même. Nous retrouvons, de manière analogue, les pointels et les lignels qui sont les sommets et les arêtes du cube, les faces du cube, de dimension 2, sont appelées *surfels* et l’intérieur du cube est la cellule de dimension 3 représentant le voxel. Ces notions sont bien sûr généralisables en dimension quelconque.

Dans un espace discret de dimension d , cette représentation fournit ainsi une granularité plus fine permettant de considérer, outre les chemins de cellules de dimension d , des chemins de cellules de dimensions inférieures à d , par exemple de chaînes de surfels ou de lignels dans \mathbb{Z}^3 .

La définition de surface discrète suivante utilise cette représentation cellulaire :

Définition 1.6 (en dimension 3) Si nous considérons un surfel s comme la face commune d’une paire $\{v, v'\}$ de voxels 6-voisins, ce que nous notons $s = \{v, v'\}$, alors on appelle surface d’un objet discret 3D O , l’ensemble des surfels $s = \{v, v'\}$ tels que $v \in O$ et $v' \notin O$.

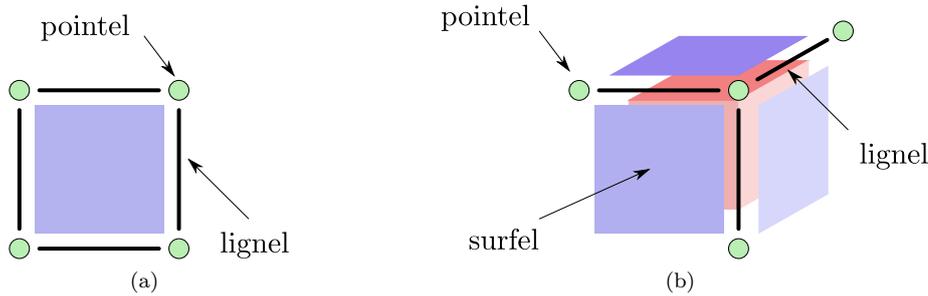


Figure 1.4 – Décomposition (a) d'un pixel et (b) d'un voxel en cellules de dimensions inférieures.

On peut généraliser cette définition en dimension n [Her92, Her98], et montrer que cette surface est séparante, au sens de Jordan, lorsqu'on emploie les bonnes relations d'adjacence.

Il est à noter que si ce formalisme permet d'introduire de bonnes notions topologiques, il reste principalement un outil de travail intermédiaire pour les images, car la représentation directe des cellules de dimensions inférieures au point n'est pas toujours possible dans le support de l'image. Le contour en lignels d'un objet 2D dans une photo a un sens (il représente la notion de surface dans ce formalisme) néanmoins, si nous désirons colorier ce contour et afficher le résultat sur un écran LCD classique, par exemple, il faudra choisir de colorier les pixels adjacents aux lignels de ce contour ; les lignels n'étant pas des éléments d'un écran LCD.

1.5 Structures de données pour l'espace discret

Nous voyons que le choix d'une définition plutôt qu'une autre peut donc aussi induire des structures de données spécifiques aux traitements à effectuer. De ce côté, une représentation matricielle des données (tableau de points) est probablement la structure la plus simple pour représenter une image. Néanmoins, cette structure peut ne pas être suffisamment élaborée, ou efficace, pour représenter une image en considérant sa décomposition cellulaire. De nombreux travaux ont été menés sur ces problématique, par exemple (en étant loin d'être exhaustif) [Lie94, Fra95, Bur02, Lac03, Kov01, Kov08].

L'approche par graphe est une possibilité, et elle se prête bien à la description de surfaces, comme développée dans les travaux de Jean Françon [Fra95]. De plus, elle permet une séparation de la topologie et de la géométrie, qui peut être intéressante dans certaines situations. La notion de surface développée dans ces travaux est celle des variétés combinatoires de dimension 2.

Soit G un graphe fini et F un ensemble de cycles élémentaires de G , c'est-à-dire des permutations circulaires $(v_0, v_1, \dots, v_{k-1})$, $k > 2$, à l'orientation près, de sommets de G telle que, pour tout i , v_i est adjacent à v_{i+1} (les indices étant considérés modulo k) et $v_i \neq v_j$ si $i \neq j$. De tels cycles sont appelés des *faces*. Deux faces sont adjacentes si elles partagent une arête. Une *ombrelle* centrée en un sommet v de G est une permutation circulaire, à l'orientation près, de faces de F , $(f_0, f_1, \dots, f_{k-1})$, $k > 1$, toutes partageant le sommet v , et telles que f_i et f_{i+1} soient adjacentes pour tout i (les indices étant considérés modulo k). Si f_0 et f_{k-1} ne sont pas adjacentes alors (f_0, \dots, f_{k-1}) forment un *éventail* de faces.

La paire $V = (G, F)$ est appelée une *variété combinatoire de dimension deux sans bord* si et seulement si les conditions suivantes sont vérifiées :

- chaque arête de G est incidente à exactement deux faces de F ;
- chaque sommet de G est le centre d'exactly une ombrelle de F .

V est une *variété combinatoire de dimension deux avec bord* si et seulement si les conditions suivantes sont vérifiées :

- chaque arête de G est incidente à exactement une ou deux faces de F , le bord de V est alors formé des arêtes incidentes à une seule face et des sommets incidents à ces arêtes ;

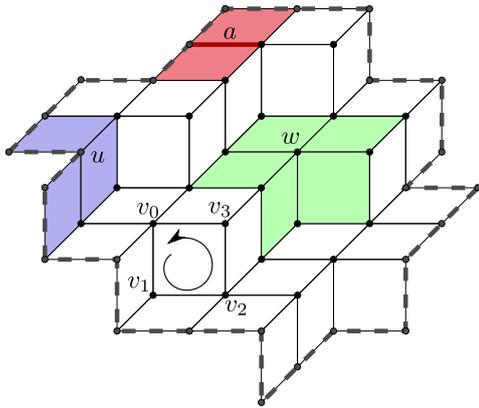


Figure 1.5 – Un graphe représentant une 2-variété combinatoire avec bord, et plus précisément un disque topologique. Le bord est indiqué en gris foncé et représenté par le chemin en pointillés. (v_0, v_1, v_2, v_3) est un cycle représentant une face. Les deux faces rouges sont adjacentes car elles partagent l'arête a . Les faces bleues forment un éventail centré en u et les faces vertes composent une ombrelle centrée en w .

- chaque sommet de G n'appartenant pas au bord de V est le centre d'exactlyement une ombrelle de F et chaque sommet du bord de V est le centre d'exactlyement un éventail de F .

La variété combinatoire de dimension 2 (ou 2-variété combinatoire) V est dite *connexe* si et seulement si G est connexe.

Nous pouvons remarquer qu'en faisant les associations, pointel \leftrightarrow sommet de G , lignel \leftrightarrow arête de G et surfel \leftrightarrow face de F , ce formalisme permet de représenter des surfaces selon l'approche hétérogène. Nous rappelons ici la notion de disque topologique puisque nous nous en servons plus tard.

Définition 1.7 Un *disque topologique* est une variété combinatoire de dimension 2 avec bord, connexe, dont le bord est composé d'une unique composante connexe.

Cette propriété sera utilisée dans les algorithmes de segmentations de plusieurs méthodes pour garantir qu'une face reconstruite aura bien un bord cohérent.

Les différentes notions présentées jusqu'ici sont illustrées dans la figure 1.5 et l'ensemble des ombrelles possibles d'une 2-variété combinatoire sans bord sont rappelées dans la figure A.2 en annexe. Mais cette représentation n'est pas le seul à avoir été étudiée.

Le modèle des cartes combinatoires de dimension n , ou leur généralisation, les n -G-cartes, est aussi un modèle très utilisé en géométrie discrète. C'est un modèle topologique combinatoire qui permet de représenter et de manipuler des objets subdivisés. Nous ne présentons que très brièvement la notion de n -G-carte car nous ne l'utilisons pas directement dans cette thèse, néanmoins nous décrivons des travaux qui l'utilisent. Une description plus complète peut être trouvée dans [Lie91, Lie94]. Le principal avantage des cartes généralisées par rapport aux cartes combinatoires est le fait qu'elles sont homogènes en toute dimension. Leur définition est la suivante :

Définition 1.8 Une carte généralisée de dimension n , (ou n -G-carte) est une algèbre $G = (B, \alpha_0, \dots, \alpha_n)$ tel que :

- B est un ensemble fini de brins ;
- $\alpha_0, \dots, \alpha_n$ sont des involutions² sur B ;
- Pour tout i, j tels que $0 \leq i < i + 2 \leq j \leq n$, $\alpha_i \circ \alpha_j$ est une involution.

Une n -G-carte est composée d'un ensemble d'entités élémentaires appelées *brins* et d'opérations α_i permettant de réaliser des « coutures » de dimension i , créant ainsi des arêtes, des surfaces et des volumes. La figure 1.6 montre la décomposition topologique d'une maison à l'aide d'une 2-G-carte.

Les cartes généralisées sont très pratiques lorsqu'il s'agit de fusionner ou de découper des régions. Elles sont par conséquent grandement utilisées dans des processus de segmentations d'images.

2. Une involution f sur un ensemble fini E est une bijection de E dans E telle que $f = f^{-1}$.

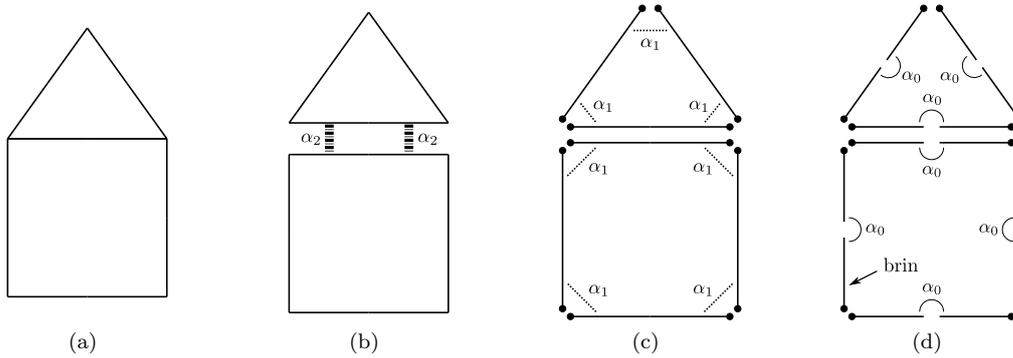


Figure 1.6 – Décomposition topologique d’une maison en 2-G-carte. (a) La topologie d’une maison simple avec un toit et (b) sa décomposition en faces topologiques cousues par des involutions α_2 . (c) La décomposition des faces en arêtes reliées par des involutions α_1 . Et pour finir, (d) la décomposition des arêtes en brins élémentaires cousus par des involutions α_0 .

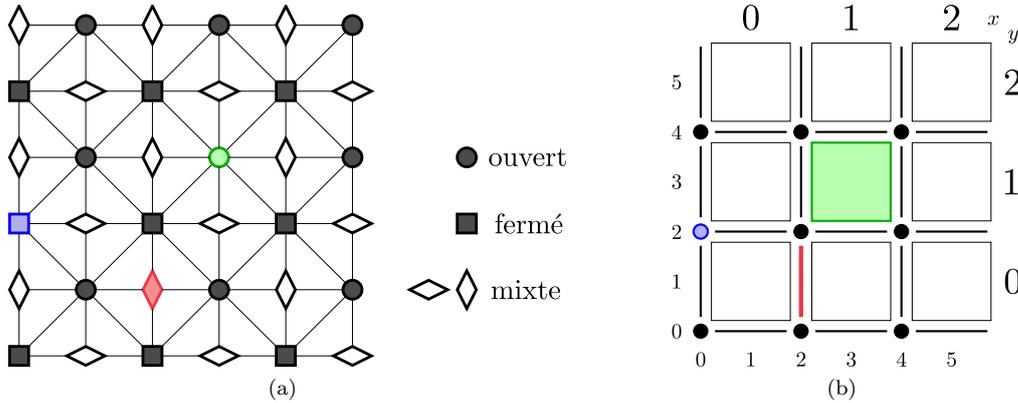


Figure 1.7 – L’isomorphisme entre (a) le plan de Khalimsky (définissant une topologie sur \mathbb{Z}^2) et (b) la décomposition cellulaire de la grille 2D. Les pointels correspondent aux fermés, les lignels aux ensembles mixtes et les pixels aux ouverts. En gros, les coordonnées des pixels dans \mathbb{Z}^2 , et en petit, les coordonnées des éléments interpixels de la décomposition cellulaire.

Pour terminer, nous pouvons rappeler une autre manière de coder les éléments d’une décomposition cellulaire pour des grilles régulières. Elle est due à une interprétation géométrique de la topologie de Khalimsky [KKM90] et nous la retrouvons dans les travaux de V. Kovalevsky [Kov89, Kov08]. En considérant une nouvelle grille, issue d’un changement d’échelle de facteur 2 de la grille sur \mathbb{Z}^2 , nous pouvons coder chaque élément de la décomposition cellulaire de la manière suivante : un point dont les deux coordonnées sont paires désigne un pointel de la décomposition (une 0-cellule) ; un point dont une coordonnée est paire et l’autre impaire représente un lignel (une 1-cellule) ; et un point dont les deux coordonnées sont paires code un pixel (une 2-cellule). La topologie est donc codée avec la parité des coordonnées dans la nouvelle grille, et une division entière par 2 suffit à retrouver les coordonnées originales d’un point. La figure 1.7 illustre ce codage.

Cette représentation s’étend à toutes les dimensions et J.O. Lachaud a d’ailleurs proposé un codage efficace de cette structure dans [Lac03]. Ses travaux ont donné lieu à la bibliothèque [Ima], un framework générique pour écrire des algorithmes de géométrie discrète en dimension quelconque. Nous l’avons d’ailleurs utilisé (tardivement) dans nos travaux pour coder des estimateurs de vecteurs normaux présentés à la section 4.2.1.1.

Les objets discrets généralement étudiés à travers ces représentations peuvent avoir différentes origines :

1. soit ils proviennent d'une transposition dans l'espace discret d'un objet continu ;
2. soit ils sont le résultat d'un procédé d'acquisition ;
3. soit ils sont directement issus de définitions mathématiques ou algorithmiques.

Dans le premier cas, la transposition s'effectue à travers une transformation, appelé *processus de discrétisation*, que nous présentons dans la section suivante. Le second cas regroupe les objets décrits par un nuage de points, provenant généralement de capteurs (appareils photos, scanners, IRM, ...) ou créés « à la main », et qui n'ont, a priori, pas de signification géométrique formelle. On désire souvent répondre à ce problème en effectuant une opération de *segmentation* visant à décomposer l'objet en éléments géométriques de base. Ces derniers, qui font partie du cas 3, sont généralement décrits analytiquement et détectés à partir d'algorithmes de *reconnaissance*. Nous présentons les primitives géométriques les plus simples, les droites et les plans, à la section 1.7 et quelques algorithmes de reconnaissance pour ces dernières à la section 1.8.

1.6 Discrétisation

Lorsqu'il s'agit de transposer un objet du continu vers le discret, plusieurs techniques sont possibles, selon qu'elles s'appliquent sur des objets possédant un intérieur et un extérieur (dont le bord possède la propriété de Jordan), ou bien sur des courbes, ou encore, pour les plus polyvalentes, sur n'importe quel objet.

Pour des objets disposant d'une notion d'intérieur et d'extérieur, un processus très simple consiste à considérer comme discrétisation d'un objet continu F , l'ensemble des points à coordonnées entières contenu dans F . Ce processus, dû à Gauss, est appelé *Object Boundary Quantization* (OBQ).

Pour les courbes non associées à des notions d'intérieur/extérieur, nous pouvons appliquer un autre schéma de discrétisation, appelé *Grid Intersect Quantization* (GIQ), qui consiste à approximer la courbe par l'ensemble des points « les plus proches » selon un certain critère. Ainsi, si nous considérons le maillage dual au pavage de l'espace discret (c'est-à-dire le graphe dont les sommets sont les centres des cellules et les arêtes représentent les adjacences entre cellules), à chaque fois que la courbe coupe une arête du maillage, seul le point discret correspondant au sommet de l'arête le plus proche de la courbe, au sens de la distance euclidienne, fait partie de la discrétisation de celle-ci. Dans le cas où l'intersection est équidistante des deux points, les deux sont conservés, au risque de faire apparaître une *bulle* dans la discrétisation.

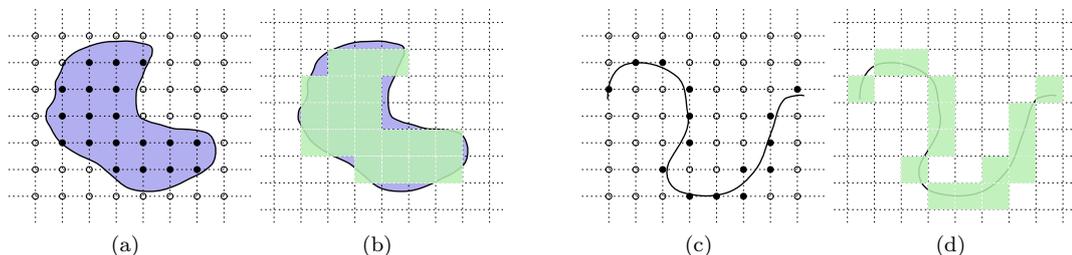


Figure 1.8 – Un objet continu, en bleu, et sa discrétisation OBQ représentée (a) par les points noirs du maillage dual du pavage formant l'espace discret et (b) par les pixels, en vert, du pavage. Une courbe continue avec sa discrétisation GIQ (c) sur le maillage et (d) sur le pavage.

Dans le cas spécifique des droites et des plans, la discrétisation GIQ peut être vue comme un processus d'arrondi à l'entier le plus proche. Ainsi, pour une droite $D : y = \alpha x + \beta$ du premier octant (pente positive

inférieure à 1), les points de la discrétisation sont donnés par ³ :

$$\{(x_i, y_i) : y_i = \lfloor \alpha x_i + \beta \rfloor, x_i \in \mathbb{Z}\}$$

Par abus de langage, on appelle discrétisation OBQ de D l'ensemble de points entiers :

$$\{(x_i, y_i) : y_i = \lfloor \alpha x_i + \beta \rfloor, x_i \in \mathbb{Z}\}$$

et l'ensemble

$$\{(x_i, y_i) : y_i = \lceil \alpha x_i + \beta \rceil, x_i \in \mathbb{Z}\}$$

est appelé discrétisation BBQ (*Background Boundary Quantization*) de D .

Enfin, un schéma de discrétisation appelé *supercouverture*, noté $\mathbb{S}(\cdot)$ définit la discrétisation d'un objet continu F comme étant l'ensemble des cellules du pavage de l'espace discret intersectées par F . Ce procédé possède de bonnes propriétés et se généralise facilement à toutes les dimensions, mais pour les objets linéaires il possède le même défaut que GIQ, à savoir la création de bulles quand l'objet passe par des sommets du pavage. La discrétisation d'une droite n'est alors pas une *courbe* 4-connexe, comme on l'aimerait, mais un *objet* 4-connexe. Le *modèle standard*, introduit dans [And02], répond à ce problème, fournissant un schéma de discrétisation supercouverture sans bulles pour les objets linéaires.

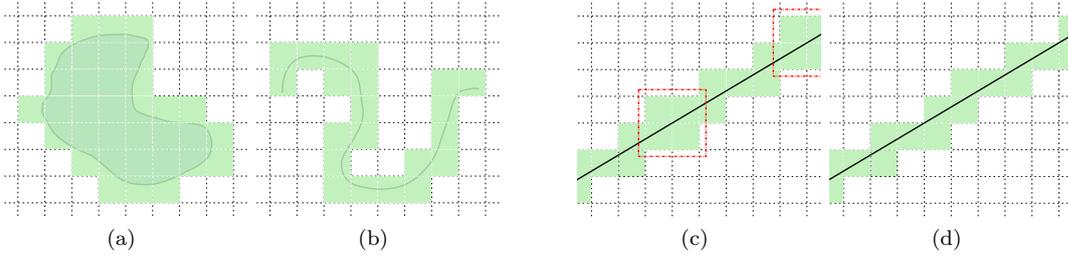


Figure 1.9 – La discrétisation supercouverture (a) d'un objet continu, (b) d'une courbe continue et (c) d'une partie de droite continue. Dans ce dernier cas, le problème des bulles, encadrées en rouge, peut être résolu en employant une discrétisation suivant le modèle standard (d).

1.7 Primitives discrètes linéaires

Dans le but de construire une géométrie de l'espace discret, il convient d'en fixer les objets géométriques de base ; les plus simples étant probablement les primitives linéaires que sont les droites et les plans. Nous présentons ici ces objets à travers leurs définitions arithmétiques.

1.7.1 Droite discrète arithmétique

La définition arithmétique d'une droite discrète a été introduite par J.-P. Reveillès dans [Rev91] :

Définition 1.9 Une *droite discrète arithmétique* de vecteur directeur (b, a) , de borne inférieure μ et d'épaisseur arithmétique ω (avec a, b, μ, ω entiers, $\omega > 0$ et a et b premiers entre eux) est l'ensemble des points entiers (x, y) vérifiant la double inégalité diophantienne

$$\mu \leq ax - by < \mu + \omega$$

Nous la notons $\mathcal{D}(a, b, \mu, \omega)$, ou tout simplement \mathcal{D} s'il n'y a pas d'ambiguïté possible. Le paramètre ω permet de contrôler la connexité de la droite, comme illustré à la figure 1.10, et en particulier :

3. $\lfloor x \rfloor$ désigne l'arrondi de x à l'entier le plus proche et $\lfloor x \rfloor$ (resp. $\lceil x \rceil$) correspond à l'entier inférieur (resp. supérieur) le plus proche de x .

- si $\omega < \max(|a|, |b|)$, \mathcal{D} n'est pas connexe ;
- si $\omega = \max(|a|, |b|)$, \mathcal{D} est une courbe 8-connexe appelée *droite naïve* ;
- si $\omega = |a| + |b|$, \mathcal{D} est une courbe 4-connexe appelée *droite standard* ;
- si $\omega > |a| + |b|$, \mathcal{D} est dite *épaisse*.

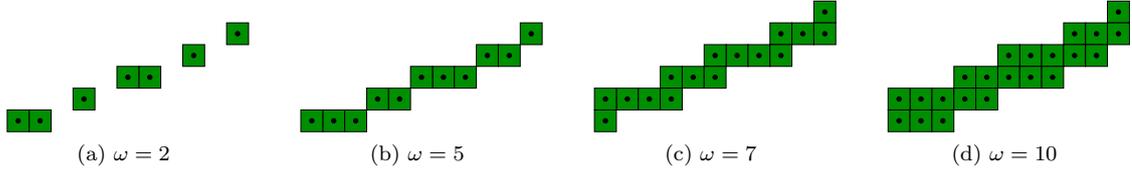


Figure 1.10 – Une droite discrète arithmétique de vecteur directeur $(5, 2)$ et de borne inférieure 0, suivant différentes épaisseurs ω .

On appelle *reste* au point $M(x_M, y_M)$ de la droite discrète $\mathcal{D}(a, b, \mu, \omega)$ la valeur entière $r(M) = ax_M - by_M - \mu$. Les deux droites euclidiennes d'équations $ax - by - \mu = 0$ et $ax - by - \mu = \omega - 1$, appelées respectivement *droite d'appui inférieure* et *droite d'appui supérieure*, sont deux droites parallèles qui forment une bande fermée contenant tous les points de \mathcal{D} . Les points de \mathcal{D} appartenant à la droite d'appui inférieure (resp. supérieure) sont appelés *points d'appui inférieurs* (resp. *supérieurs*).

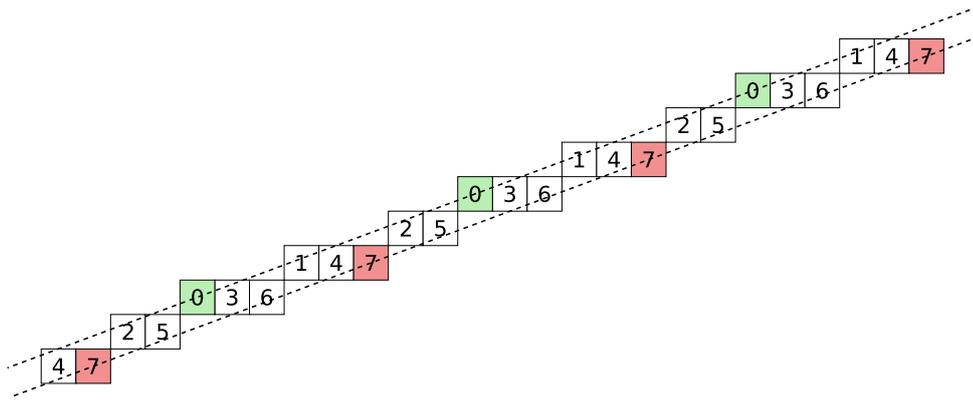


Figure 1.11 – Un segment de la droite discrète naïve $\mathcal{D}(3, 8, -4, 8)$. La valeur du reste en chaque point est représentée dans chaque pixel. La droite d'appui inférieure (resp. supérieure) passant par les points d'appui inférieurs en vert (resp. supérieurs en rouge) est représentée en pointillés.

Ces notions, illustrées sur la figure 1.11, sont d'un grand intérêt puisqu'elles sont la base du premier algorithme de reconnaissance de droites discrètes arithmétiques. Mais nous y reviendrons après avoir introduit la définition des plans discrets arithmétiques.

1.7.2 Plan discret arithmétique

La définition des plans discrets arithmétiques est en fait une généralisation de celle des droites discrètes arithmétiques [Rev91, And93] :

Définition 1.10 Un *plan discret arithmétique* de vecteur normal (a, b, c) , de paramètre de translation μ et d'épaisseur arithmétique ω (avec a, b, c, μ, ω entiers, $\omega > 0$ et a, b, c premiers entre eux) est l'ensemble des points entiers (x, y, z) vérifiant

$$\mu \leq ax + by + cz < \mu + \omega$$

Nous le notons $\mathcal{P}(a, b, c, \mu, \omega)$, et pouvons définir, de manière analogue à celle des droites, la notion de reste au point $M(x_M, y_M, z_M)$ comme étant la valeur entière $r(M) = ax_M + by_M + cz_M - \mu$, ainsi

que les notions de points d'appui et de plans d'appui, où ces derniers sont cette fois-ci les plans euclidiens d'équations $ax + by + cz - \mu = 0$ et $ax + by + cz - \mu = \omega - 1$.

Si nous regardons la relation entre la connexité des plans et le paramètre ω (figure 1.12) nous retrouvons des cas similaires aux droites discrètes, avec :

- les plans discrets *naïfs*, quand $\omega = \max(|a|, |b|, |c|)$;
- les plans discrets *standard*, quand $\omega = |a| + |b| + |c|$;
- les plans discrets *épais* quand $\omega > |a| + |b| + |c|$.

Lorsque $\omega < \max(|a|, |b|, |c|)$ l'introduction d'une dimension supplémentaire par rapport aux droites discrètes fait apparaître des nouveaux cas de figures (cf. figure 1.12b) où le plan discret, bien que restant formé d'une seule composante connexe, contient des « trous ». Sans toutefois le développer ici, la description des cas de figures possibles peut être formalisée à travers la notion de *tunnel* [AAS97].

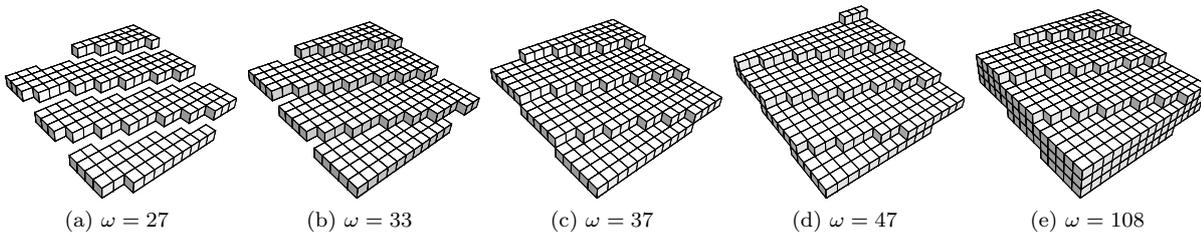


Figure 1.12 – Une plan discret arithmétique de vecteur normal $(3, 7, 37)$ et de borne inférieure 0, suivant différentes épaisseurs ω .

Si la définition et la caractérisation des ces primitives linéaires est une étape importante dans la construction d'une théorie discrète de la géométrie, lorsque nous désirons appliquer en pratique les concepts développés, il est nécessaire de pouvoir identifier si un ensemble de points peut être décrit par une de ces primitives. Nous avons donc besoin d'algorithmes de reconnaissance permettant de répondre à cette problématique. Nous présentons dans la section suivante ceux qui ont servi de base à nos travaux, ou qui les ont inspirés.

1.8 Algorithmes de reconnaissance de primitives discrètes linéaires

Les algorithmes de reconnaissance de droites et plans discrets arithmétiques que nous exposons peuvent se classer dans différentes catégories, selon qu'ils utilisent directement l'arithmétique sous-jacente aux primitives discrètes, ou qu'ils passent par un espace dual et se ramènent à un problème de programmation linéaire ou encore qu'ils soient issus de la géométrie algorithmique. Un algorithme aura un intérêt supplémentaire s'il est incrémental, car il sera dans ce cas un candidat de choix pour des applications utilisant des approches par croissance de régions, comme la segmentation ou la polygonalisation d'objets, par exemple.

1.8.1 Approche arithmétique

Le premier algorithme incrémental de reconnaissance de segments de droites discrète naïves a été proposé dans [DR95, DRR95] et exploite l'arithmétique des droites discrètes. Soit S un segment de $\mathcal{D}(a, b, \mu)$, avec $0 \leq a \leq b$, et M un point ajouté à S . Le théorème ci-dessous permet de dire si $S' = S \cup \{M\}$ est ou non un segment de droite discrète naïve et, si c'est le cas, il fournit les caractéristiques a, b, μ de la droite correspondant à S' .

Théorème 1.1 *Soit $S(M_0, M_1)$ un segment de $\mathcal{D}(a, b, \mu)$, avec $0 \leq a \leq b$. Le point $M(x_M, y_M)$ (avec $M = M_1 + (1, 0)$ ou $M = M_1 + (1, 1)$) est ajouté à S . U_F (resp. L_F) est le point d'appui supérieur (resp. inférieur) de S dont la coordonnée en x est minimale et U_L (resp. L_L) est le point d'appui supérieur*

(resp. inférieur) de S dont la coordonnée en x est maximale. On considère alors la valeur du reste r au point M en fonction de \mathcal{D} . Alors :

- (i) si $0 \leq r(M) < \omega$, $M \in \mathcal{D}(a, b, \mu)$ et $S \cup \{M\}$ est un segment de \mathcal{D} ;
- (ii) si $r(M) = -1$, $S \cup \{M\}$ est un segment de la droite naïve dont la pente est donnée par le vecteur $\overrightarrow{U_F M}$;
- (iii) si $r(M) = \omega$, $S \cup \{M\}$ est un segment de la droite naïve dont la pente est donnée par le vecteur $\overrightarrow{L_F M}$;
- (iv) si $r(M) < -1$ ou $r(M) > \omega$, $S \cup \{M\}$ n'est pas un segment de droite discrète naïve.

L'élaboration de l'algorithme incrémental et linéaire de reconnaissance de droites discrètes naïves [DR95] découle directement de ce théorème. Les points sont ajoutés suivant les abscisses croissantes de x et l'algorithme effectue simplement une mise à jour des caractéristiques de \mathcal{D} en fonction du cas rencontré.

Un algorithme de reconnaissance de morceaux de plans discrets naïfs, incrémental et linéaire, a aussi été proposé dans [DR95]. Ce sont cette fois-ci les polygones des points d'appui qui sont utilisés pour mettre à jour les caractéristiques du morceau de plan à reconnaître, en fonction des valeurs des restes des points ajoutés. Cet algorithme ne fonctionne néanmoins que sur des morceaux rectangulaires de plans discrets, car il exploite un parcours des points par balayage, en suivant un ordre lexicographique sur leurs coordonnées.

1.8.2 Reconnaissance par dualité

Une autre approche permettant de définir des algorithmes efficaces de reconnaissance de primitives discrètes linéaires consiste à utiliser un *espace dual*, appelé aussi *espace des paramètres*.

1.8.2.1 Notion de préimage

La transformation classique considérée en géométrie est celle qui à une droite d'équation $y = \alpha x + \beta$ de l'espace euclidien associe un point de coordonnées (α, β) dans l'espace des paramètres et vice-versa, à une droite d'équation $\beta = -x\alpha + y$ dans le dual est associée un point (x, y) dans l'espace euclidien. Ainsi, des alignements de points dans l'espace euclidien correspondent à des droites concourantes dans l'espace dual.

La notion de préimage, introduite dans [DS84], utilise cet espace dual pour caractériser des droites discrètes.

Définition 1.11 Étant donné un procédé de discrétisation Q , on appelle **préimage** d'un ensemble X de pixels, le domaine, dans l'espace des paramètres, représentant l'ensemble des droites réelles dont la discrétisation selon Q contient X .

Dans le premier octant, considérons par exemple le modèle de discrétisation OBQ. D'après les définitions de la section 1.6, nous savons que si nous considérons un segment X , formé d'un sous-ensemble de pixels $\{(x, y)\}$ d'une droite discrète naïve alors, il existe un ensemble de couples $(\alpha, \beta) \in [0, 1] \times [0, 1[$ tels que X est contenu dans la discrétisation OBQ de la droite réelle $y = \alpha x + \beta$.

La préimage de X est donc l'ensemble

$$\{(\alpha, \beta) \in [0, 1] \times [0, 1[\mid \forall (x, y) \in X, 0 \leq \alpha x + \beta - y < 1\}$$

Sa décomposition nous donne que chaque pixel $(x, y) \in X$ engendre dans l'espace dual des paramètres (α, β) une bande B définie par :

$$B(x, y) = \begin{cases} \alpha x + \beta - y \geq 0 \\ \alpha x + \beta - y < 1 \end{cases}$$

Présentée sous cette forme, la préimage est un domaine convexe, généré par intersection de contraintes linéaires. Par conséquent, savoir si un ensemble de pixels est un sous-ensemble d'une droite discrète se ramène à un problème de programmation linéaire. De nombreux algorithmes existent pour résoudre ce

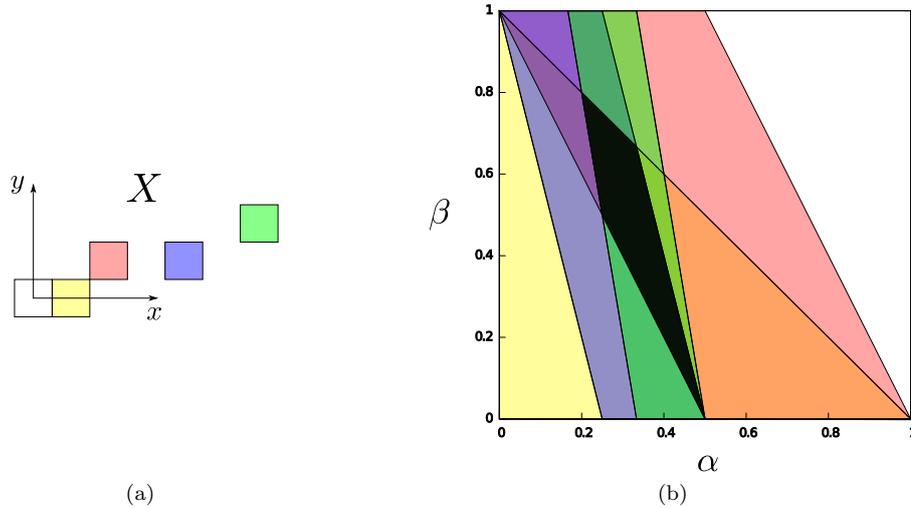


Figure 1.13 – (a) Un ensemble X de pixels du plan et (b) la préimage de X (le domaine en noir) pour le modèle de discrétisation OBQ. Les bandes générées par chaque pixel sont représentées de la même couleur que le pixel.

problème. Si nous désirons uniquement savoir si X est un sous-ensemble d'une droite discrète, alors il s'agit seulement de vérifier que le système admet une solution ; c'est un problème de décision qui se calcule en temps $\mathcal{O}(n)$ pour un système de n inéquations linéaires.

S'il s'agit d'identifier les paramètres des droites discrètes contenant X , alors il faut calculer le polytope des solutions. Des algorithmes de calcul d'enveloppe convexe peuvent être employés et fournissent des solutions efficaces en temps $\mathcal{O}(n \log n)$.

Ces approches offrent des techniques générales pour déterminer si un ensemble de pixels est un sous-ensemble d'une droite discrète, néanmoins, s'il s'agit de reconnaître des segments de droites, l'ajout de contraintes de connexité entre les pixels permet d'élaborer des algorithmes plus efficaces. Par exemple, en considérant un ensemble de pixels 8-connexe (cas des droites naïves), la préimage peut être décrite plus précisément [DS84, McI85] :

Propriété 1.1 Soit S un ensemble des $N + 1$ pixels 8-connexes dont l'abscisse minimale vaut x_0 . Alors la préimage $\mathcal{P}(S)$ de S a les propriétés suivantes :

1. $\mathcal{P}(S)$ est un polygone convexe possédant au plus 4 sommets ;
2. les abscisses de ses sommets sont des termes successifs de la suite de Farey⁴ d'ordre $\max x_0, N - x_0$. De plus, si l'abscisse d'un sommet est $\frac{p}{q}$, alors son ordonnée est un multiple de $\frac{1}{q}$;
3. si ce polygone a quatre sommets, alors deux sommets ont la même abscisse.

En utilisant ces propriétés les auteurs proposent des algorithmes de reconnaissance optimaux en temps. Cette préimage s'étend à la dimension supérieure et est à la base de nombreux autres algorithmes de reconnaissance de droites et plans discrets [Vit99, VC00, Coe02b].

1.8.2.2 Préimage généralisée

Dans [Dex06, DA09], les auteurs définissent une forme généralisée de préimage pour une reconnaissance d'hyperplans discrets en dimension quelconque. Cette dernière est particulièrement bien adaptée pour la reconnaissance d'hyperplans selon un modèle de discrétisation supercouverture, mais elle est assez générale pour être employée avec d'autres modèles de discrétisation. Cette *préimage généralisée* est définie, pour

4. Une suite de Farey d'ordre $n \in \mathbb{N}^*$ est la suite des fractions irréductibles de l'intervalle $[0, 1]$, rangées par ordre croissant, dont les dénominateurs sont inférieurs à n .

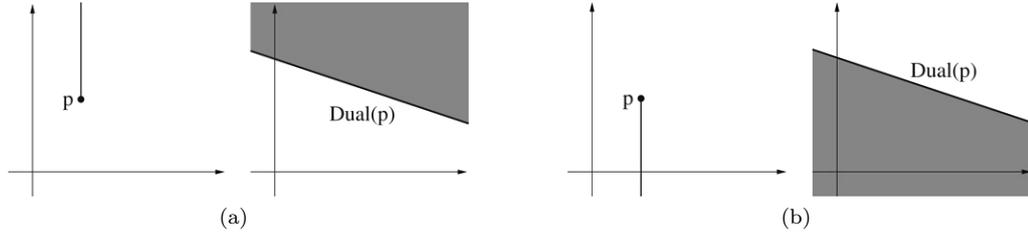


Figure 1.14 – L'extrusion d'un point (une demi-droite) et son dual un demi-espace en dimension 2 : (a) l'extrusion positive et (b) l'extrusion négative (source [DA09]).

l'espace euclidien \mathcal{E}_n de dimension n , en considérant un espace des paramètres $\mathcal{P}_n \subset \mathbb{R}^n$ et la dualité basée sur les transformations $D_{\mathcal{E}} : \mathcal{E}_n \rightarrow \mathcal{P}_n$ et $D_{\mathcal{P}} : \mathcal{P}_n \rightarrow \mathcal{E}_n$:

$$D_{\mathcal{E}}(x_1, \dots, x_n) = \left\{ (y_1, \dots, y_n) \in \mathcal{P}_n \mid y_n = - \sum_{i=1}^{n-1} x_i y_i + x_n \right\}$$

$$D_{\mathcal{P}}(y_1, \dots, y_n) = \left\{ (x_1, \dots, x_n) \in \mathcal{E}_n \mid x_n = \sum_{i=1}^{n-1} y_i x_i + y_n \right\}$$

Ainsi, chaque point dans \mathcal{E}_n (resp. \mathcal{P}_n) se transforme par $D_{\mathcal{E}}$ (resp. $D_{\mathcal{P}}$) en un hyperplan dans \mathcal{P}_n (resp. \mathcal{E}_n). Pour un ensemble X de polytopes de \mathcal{E}_n , la préimage généralisée est le domaine dans \mathcal{P}_n représentant l'ensemble des hyperplans coupant X . Dans [DA09], elle est définie de la façon suivante :

Définition 1.12 Soit $X = (X_1, \dots, X_k)$ un ensemble de k polytopes, la **préimage généralisée** \mathbb{G}_p de X est l'ensemble

$$\mathbb{G}_p(X) = \bigcap_{i=1}^k Dual(X_i)$$

Les auteurs proposent dans [DA09] un ensemble assez conséquent de définitions et de propriétés pour formaliser la construction du dual d'un polytope. Nous n'allons pas les reprendre ici. Nous proposons simplement les figures 1.14 et 1.15 synthétisant les résultats en 2D, ainsi que la notion principale sous-jacente, *l'extrusion d'un point* :

Définition 1.13 Soit $p(x_1, \dots, x_n)$ un point de \mathbb{R}^n . L'*extrusion positive* de p dans \mathbb{R}^n est définie par

$$p^+ = \{p'(x'_1, \dots, x'_n) \in \mathbb{R}^n \mid \forall i \in \llbracket 1, n-1 \rrbracket, x'_i = x_i \text{ et } x'_n \geq x_n\}$$

De même, l'*extrusion négative* de p est définie par

$$p^- = \{p'(x'_1, \dots, x'_n) \in \mathbb{R}^n \mid \forall i \in \llbracket 1, n-1 \rrbracket, x'_i = x_i \text{ et } x'_n \leq x_n\}$$

L'algorithme incrémental de reconnaissance d'hyperplans discrets basé sur cette préimage généralisée consiste simplement, lors de l'ajout d'un hypervoxel v à un morceau de plan P en cours de reconnaissance, à calculer l'intersection entre $G_p(P)$ et $Dual(v)$. Tant qu'elle n'est pas vide, le morceau P est un hyperplan discret. Nous pouvons remarquer que cet algorithme ne dépend pas de la connexité entre les hypervoxels. La figure 1.16 illustre cette procédure sur un segment discret.

1.8.3 Approche géométrique

Pour finir, un algorithme de reconnaissance de morceaux de plans discrets naïfs original été proposé [Gér03], et exploite les propriétés d'un ensemble particulier, l'ensemble des cordes. Les différentes notions relatives à cet ensemble ont été développées dans le mémoire de thèse de Yan Gérard [Gér99].

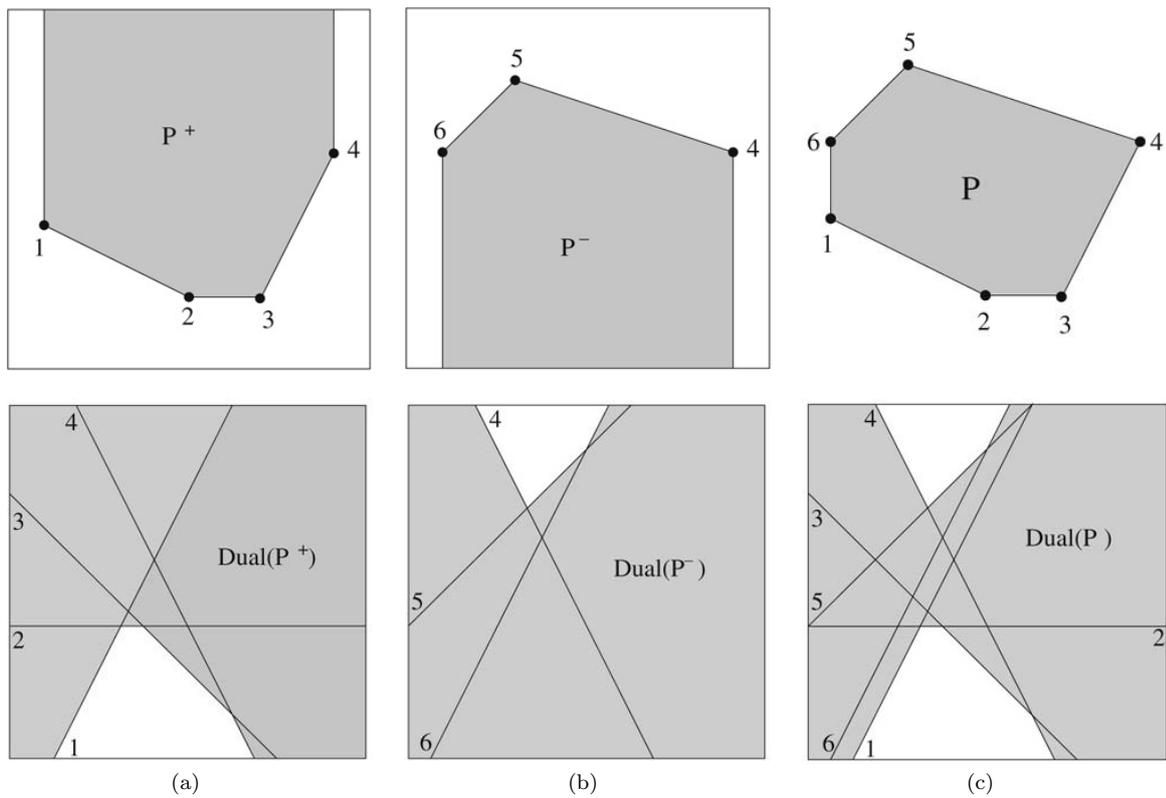


Figure 1.15 – (a) L'extrusion positive d'un polytope (en haut) et son dual (en bas), (b) l'extrusion négative d'un polytope et son dual et (c) le dual d'un polytope : $Dual(P) = Dual(P^+) \cap Dual(P^-)$ (source [DA09]).

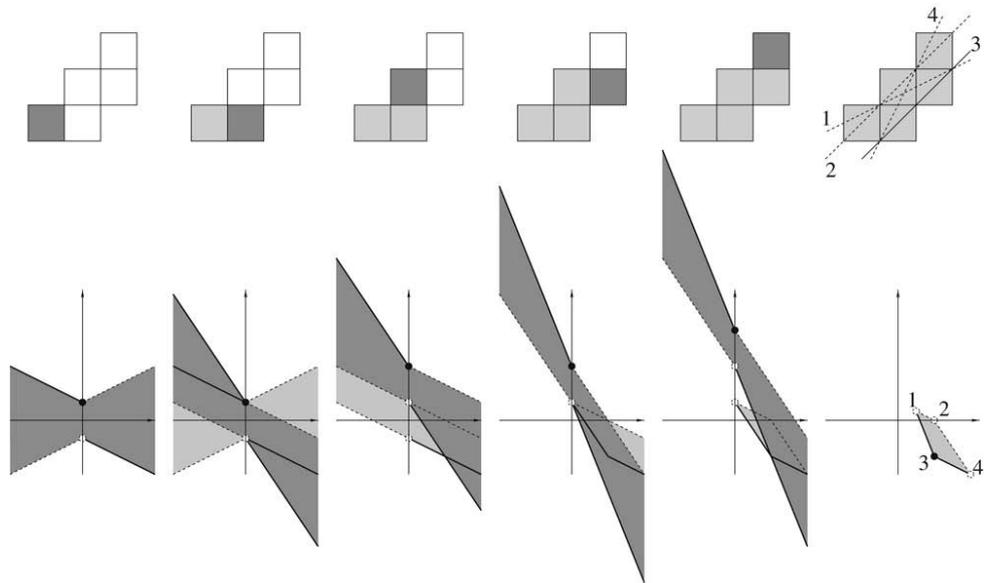


Figure 1.16 – L'évolution de la préimage généralisée (en bas) en fonction de l'ajout d'un pixel gris foncé à un segment discret en cours de reconnaissance (source [DA09]).

Définition 1.14 L'ensemble des cordes de $S \subset \mathbb{R}^n$ est l'ensemble des différences entre les points de S , noté $chord(S)$:

$$chord(S) = S + (-S) = \{m' - m \mid m, m' \in S\}$$

Si le cardinal S est égal à n alors celui de $chord(S)$ est au plus de $n(n - 1) + 1$.

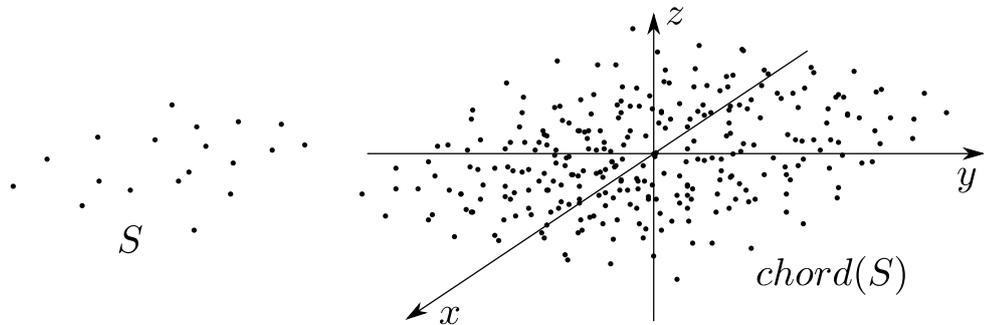


Figure 1.17 – Un ensemble fini S de 19 points et $chord(S)$ (source [GDRZ05]).

Si nous notons $conv(S)$ l'enveloppe convexe de S , alors nous avons la relation suivante

$$\forall S \subset \mathbb{R}^n, conv(chord(S)) = chord(conv(S))$$

Nous pouvons alors définir l'épaisseur géométrique de S dans une direction :

Définition 1.15 L'épaisseur géométrique d'un ensemble fini $S \subset \mathbb{R}^n$ dans la direction Oz est la coordonnée en z du point du demi-axe Oz (avec $z \geq 0$) appartenant à la surface de $conv(chord(S))$. Elle sera notée $\tau_{Oz}(S)$.

Le théorème suivant est à la base de l'algorithme de reconnaissance de morceaux de plans discrets naïfs. Nous supposons dans la suite que $c = \max(|a|, |b|, |c|)$.

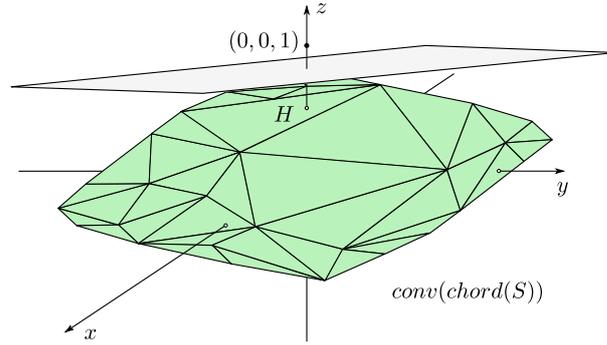


Figure 1.18 – L’enveloppe convexe de $chord(S)$: l’épaisseur de S dans la direction Oz est la coordonnée en z du point H (source [GDRZ05]).

Théorème 1.2 ([Gér03]) Soit $S \subset \mathbb{Z}^3$ un ensemble fini, alors il existe $(a, b, c, \mu) \in \mathbb{Z}^4$ vérifiant $\forall (x, y, z) \in S, \mu \leq ax + by + cz < \mu + c$ si et seulement si $\tau_{Oz}(S) < 1$.

L’algorithme de reconnaissance de plans discrets naïfs basé sur ce théorème consiste à rechercher sur $conv(chord(S))$ la ou les facette(s) coupée(s) par le demi-axe Oz positif. Si la hauteur (c.-à-d. la composante z du point d’intersection de la facette avec Oz) est supérieur à 1, alors conformément au théorème 1.2 S n’est pas un plan discret naïf, sinon il l’est.

Dans [GDRZ05], les auteurs proposent un algorithme de reconnaissance de plans discrets naïfs qui, en plus de décider si un ensemble de points de \mathbb{Z}^3 est un morceau de plan discret naïf, permet de retrouver les caractéristiques d’un tel plan. De plus, si ces caractéristiques (a, b, c, μ) sont telles que $c = \left\lceil \frac{1}{1 - \tau_{Oz}(S)} \right\rceil$, alors la minimalité de c est assurée. L’algorithme utilise une approche subtile permettant d’éviter de calculer explicitement l’enveloppe convexe de $chord(S)$. À partir d’un premier triangle T , formé de trois points de $chord(S)$, dont la projection dans Oxy contient l’origine, l’algorithme va rechercher deux points de S , m_{min} et m_{max} dont l’un minimise $\vec{n} \cdot m_{min}$ et l’autre maximise $\vec{n} \cdot m_{max}$, où \vec{n} représente le vecteur normal à T de composante $z > 0$. Dans ce cas, $M = m_{max} - m_{min}$ est un point de $chord(S)$ qui maximise la forme linéaire $(\vec{n} \cdot)$ sur $chord(S)$ et deux cas sont envisageables :

1. M est sur T , donc T est une face de $conv(chord(S))$. Si la hauteur de T est inférieure à 1, alors S est un morceau d’un plan naïf $\mathcal{P}(a, b, c, \mu, c)$ où $(a, b, c) = \vec{n}$ et μ est donnée par $\vec{n} \cdot m_{min}$;
2. sinon les sommets de T définissent avec M un tétraèdre dont l’une des faces est, par construction, coupée par Oz et plus haute que T . Si cette face a une hauteur supérieure ou égale à 1, alors S n’appartient pas à un plan discret naïf, sinon elle est choisie comme nouveau triangle T .

Une version incrémentale de cette algorithme a aussi été proposée. Même si la cardinalité plutôt élevée de $chord(S)$ par rapport à celle de S fait que la complexité en temps au pire cas de l’algorithme est en $\mathcal{O}(n^7)$, où n représente le nombre de points de l’ensemble S , le comportement de l’algorithme semble plutôt linéaire en pratique.

1.9 Synthèse

Nous avons brièvement introduit dans ce chapitre, des notions de base en géométrie discrète. Après avoir présenté les espaces discrets dans lesquels nous travaillons, nous avons décrit les objets fondamentaux manipulés. Nous nous sommes un peu plus attardés sur les primitives linéaires que sont les droites et les plans discrets, et nous avons présenté différentes techniques de reconnaissance pour de telles entités géométriques.

Dans le chapitre suivant, nous allons voir comment ces structures sont utilisées dans le cadre du problème qui a guidé les étapes de cette thèse, à savoir, celui de la polyédrisation d’objets discrets 3D.

Chapitre 2

Un panorama des méthodes de polyédrisation en géométrie discrète

Sommaire

2.1	Introduction	23
2.2	Méthodes produisant un maillage triangulaire du bord de l'objet	24
2.2.1	Méthodes se basant sur des configurations locales	24
2.2.2	Méthodes basées sur un « Diagramme de Voronoï discret »	26
2.3	Méthodes réversibles se basant sur des primitives discrètes	32
2.3.1	Polyédrisation en polyèdre discret standard	33
2.3.2	Méthodes exploitant la préimage	35
2.3.3	Reconstruction analytique inversible	42
2.4	Synthèse sur la polyédrisation en géométrie discrète	45

2.1 Introduction

De manière générale, on entend par « polyédrisation » le passage d'une représentation d'un objet sous forme de nuage de points à une représentation sous forme de structure polyédrique, c'est-à-dire un ensemble structuré de sommets, d'arêtes et de faces.

Dans le domaine de la géométrie algorithmique on parlera plus souvent de *génération de maillage* (mesh generation). Le maillage étant en fait un terme plus générique pour désigner une structure qui représente la frontière (maillage surfacique) d'un objet, où bien l'objet dans sa totalité (maillage volumique), comme une décomposition en primitives géométriques simples telles que des triangles, des polygones ou des tétraèdres.

Du côté de la géométrie discrète et de l'analyse d'images on parlera plutôt de *reconstruction polyédrique réversible* ou d'*approximation polyédrique* en fonction des méthodes utilisées.

La polyédrisation est une étape importante du traitement d'un objet discret, généralement destinée à en simplifier ou en alléger la représentation. Mais ceci dépend principalement des applications envisagées. Si pour des opérations ensemblistes (union, intersection, différence, ...) une représentation volumique des objets sous forme de voxels semble raisonnable (voire même préférable), dans un contexte de visualisation des données, où la rapidité d'affichage et la qualité (beauté) du rendu sont primordiaux, il apparaît claire qu'une représentation de l'objet sous forme voxel n'est plus vraiment adaptée, au vu de la quantité de données à afficher. Dans ce cas, la polyédrisation permettra par exemple de remplacer un groupe de voxels coplanaires par une simple face polygonale, beaucoup plus adaptée aux environnements de développement d'applications de visualisation (spécification OpenGL par exemple). La polyédrisation est ainsi *un pont*

entre l'espace discret et l'espace continu. Passer d'une représentation voxel à une structure polyédrique donne la possibilité d'utiliser des algorithmes développés pour ces dernières afin de traiter des données qui à la base n'étaient qu'un nuage de points.

Cette opération est d'ailleurs primordiale pour certaines applications, comme le modèleur géométrique à base topologique d'objets discrets (SpaMod) développé au laboratoire de Poitiers [DA05]. Ce dernier permettant de passer de manière bijective d'une représentation discrète à une représentation continue d'un objet, l'opération de polyédrisation est bien un des fondements du logiciel. Elle devra dans ce cas être réversible pour pouvoir passer d'une représentation à l'autre.

Mais différents critères peuvent caractériser une polyédrisation :

- le type des éléments géométriques qui la composent : triangles, quadrilatères ou polygones quelconques, éléments homogènes ou hétérogènes, petits ou grands, etc...
- le nombre de ces éléments
- le respect de contraintes topologiques ou géométriques

ou encore des critères plus subjectifs tels que la qualité visuelle du résultat, notamment dans un contexte de visualisation des données.

Dans l'état de l'art qui suit, nous allons nous concentrer sur les méthodes de polyédrisation « du point de vue de la géométrie discrète », c'est-à-dire que nous n'aborderons quasiment pas les techniques issues de la géométrie algorithmique (telles que les triangulations ou tétraédralisations de Delaunay ou bien les Alpha-Shapes). On se focalisera sur les techniques particulières développées pour traiter des données plongées dans un espace discret.

De nombreux travaux ont été entrepris sur la reconstruction polyédrique, mais ceux décrits ci-après peuvent se classer dans principalement deux catégories :

1. les techniques produisant une triangulation du bord de l'objet discret ;
2. et les techniques réversibles basées sur une décomposition en primitives discrètes linéaires.

Nous décrivons dans les sections suivantes de ce chapitre les différentes méthodes sus-évoquées.

2.2 Méthodes produisant un maillage triangulaire du bord de l'objet

2.2.1 Méthodes se basant sur des configurations locales

Les techniques décrites dans cette section s'appuient sur une étude des configurations des points dans un voisinage fixé pour déduire un ensemble de polygones approximant la surface de l'objet dans le voisinage considéré.

Dans cette catégorie, la méthode la plus répandue est probablement celle du Marching Cubes, développée par W.E. Lorensen et H.E. Cline [LC87]. Cette dernière permet de calculer une triangulation d'une isosurface⁵ d'un objet discret représenté sous forme d'image raster 3D.

Pour cela la méthode utilise une approche « divide-and-conquer » pour déterminer comment la surface intersecte un cube *virtuel* (le marching cube) formé de huit points de l'image, dont quatre sont voisins dans une tranche de l'objet et quatre sont voisins dans une tranche voisine, comme illustré sur la figure 2.1. Une fois ce cube traité, on passe au suivant.

Dans sa version originelle, l'algorithme a été développé pour traiter des images issues du domaine médical. Les valeurs des voxels de l'image représentent donc un échantillonnage d'une certaine fonction continue, et une valeur σ , fournie par l'utilisateur, déterminera l'isosurface de cette fonction dont on veut calculer une triangulation. Ainsi, un sommet du marching cube est initialisé à *un* si la valeur du voxel représenté par ce sommet est supérieure (ou égale) à σ ; le sommet est ainsi considéré comme étant à

5. On peut considérer une isosurface comme l'analogie en 3D d'une courbe de niveau. C'est en fait le lieu des points de l'espace pour lesquels une certaine fonction (par exemple la pression, la température, la vitesse, la densité, etc...) est constante. L'isosurface de valeur c d'une fonction f est donc l'ensemble $\{(x, y, z) \in R^3 \mid f(x, y, z) = c\}$.

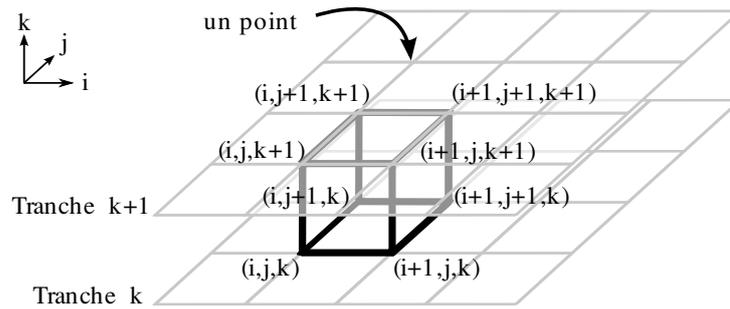


Figure 2.1 – Composition d'un Marching Cube.

l'intérieur (ou sur) la surface en construction. Les sommets dont la valeur est inférieure à σ sont eux initialisés à *zéro* et sont considérés comme étant à l'extérieur de la surface. On est ainsi ramené à calculer la triangulation d'une image binaire.

Dans ces conditions, l'algorithme du Marching Cubes considère que la surface de l'objet intersecte les arêtes du cube virtuel pour lesquelles une extrémité est à *un* et l'autre à *zéro*. La topologie de la surface à l'intérieur de ce cube est alors déterminée et une interpolation linéaire des valeurs des voxels permet de déterminer les coordonnées de l'intersection avec l'arête.

Étant donné qu'un cube possède huit sommets, et qu'un sommet peut valoir soit zéro soit un, seulement $2^8 = 256$ configurations différentes peuvent apparaître dans l'algorithme du Marching Cubes. Néanmoins, grâce aux symétries du cube et à la complémentarité des configurations où les sommets zéros et uns sont interchangeables, on peut ramener ce nombre à seulement 15 configurations différentes, toutes représentées sur la figure 2.2

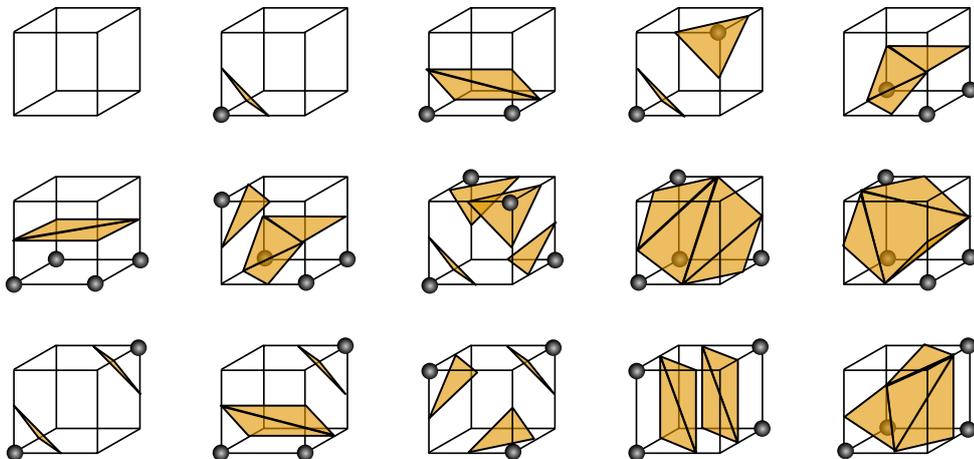


Figure 2.2 – Les 15 configurations de l'algorithme du Marching Cubes.

Les différentes configurations (256) sont précalculées et stockées dans une table. Chaque configuration est indexée par un octet dont chaque bit représente l'état d'un sommet du cube virtuel de la configuration. Lors du parcours de l'image, la triangulation locale associée au marching cube courant est alors obtenue très rapidement grâce à cette table. Au final, l'ensemble des triangles obtenus par chaque cube virtuel représente la triangulation de l'isosurface recherchée.

Les avantages de cet algorithme de reconstruction polyédrique sont indéniablement sa simplicité et sa rapidité. Néanmoins le nombre de triangles obtenus est très élevé et ces derniers sont très petits. De plus rien n'assure que la triangulation obtenue est une 2-variété combinatoire, notamment rien ne garantit que la triangulation soit fermée ou sans trous.

On peut donc raisonnablement penser qu'utilisé tel quel, sur des données bruitées, l'algorithme du

Marching Cubes ne produirait pas des résultats satisfaisants. Néanmoins des travaux ont été menés, d'une part pour certifier que la triangulation obtenue est une 2-variété combinatoire et conserve certaines caractéristiques topologiques voulues [Lac96, LM00], et d'autre part pour simplifier la triangulation en contrôlant l'approximation effectuée [KT96]. On peut donc imaginer utiliser cette polyédrisation comme base de travail pour des algorithmes de plus haut niveau.

2.2.2 Méthodes basées sur un « Diagramme de Voronoï discret »

Dans cette partie nous présentons deux méthodes de reconstruction polyédrique qui s'appuient sur un *diagramme de Voronoï discret*. Ce diagramme est obtenu à partir de *graines* placées à la surface de l'objet en fonction de la courbure de cette dernière. Les méthodes semblent similaires, mais la façon d'obtenir le diagramme de Voronoï et son utilisation pour extraire une triangulation de la surface ne sont en fait pas les mêmes.

2.2.2.1 Approche par squelettisation

Dans [Bur02, BM03], Jasmine Burguet propose une méthode de polyédrisation de la surface en surfels d'un objet discret. Le diagramme de Voronoï est ici obtenu en utilisant un algorithme de squelettisation.

Dans un premier temps, des graines sont placées à la surface de l'objet en fonction de la courbure moyenne de cette dernière. L'idée est de placer plus de graines aux endroits où la surface a une courbure importante, de façon à ce que la surface polyédrique résultante soit proche de la surface discrète initiale. Les étapes du placement des graines à la surface Σ de l'objet sont en fait les suivantes :

Pour tout surfel $s \in \Sigma$ on calcule la courbure moyenne de la surface en s grâce à un algorithme proposé par A. Lenoir [Len97]. Ensuite un surfel $x \in \Sigma$ est choisi au hasard – ça sera la première graine – et on itère le processus suivant :

1. on détermine pour x une largeur maximale $l(x)$ qui dépend de la courbure moyenne en x ;
2. on effectue un parcours en largeur du graphe d'adjacence de Σ à partir de x , jusqu'à ce que la largeur limite $l(x)$ ait été atteinte, en marquant les surfels parcourus ;
3. s'il reste des surfels non marqués sur Σ , on choisit une nouvelle graine parmi les voisins non marqués des surfels déjà visités et on recommence à partir de l'étape 1 ; sinon on arrête le processus, toutes les graines ont été placées.

La figure 2.3 présente quelques résultats de ce procédé sur différents objets discrets.

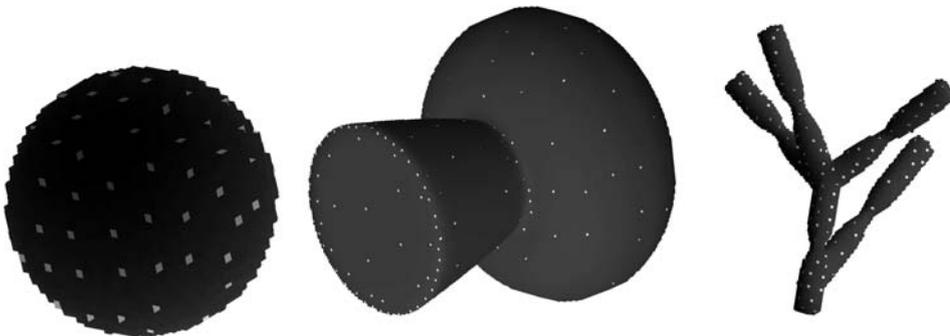


Figure 2.3 – Placement des graines à la surface d'un objet discret (source [BM03]).

L'étape suivante de la méthode de polyédrisation est une squelettisation de la surface Σ à laquelle l'ensemble des graines a été ôté. Si l'on se réfère à la figure 2.3, ceci signifie que l'on va effectuer une squelettisation (amincissement) de la surface grise, relativement aux points blancs considérés comme extérieurs à la surface.

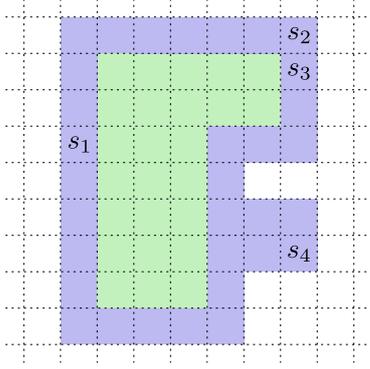


Figure 2.4 – La notion de P_e -simplicité. Considérons l'ensemble X de surfels verts et bleus avec la e -adjacence. L'ensemble $P \subset X$ est quant à lui formé des surfels bleus uniquement et l'ensemble $R = X \setminus P$ des surfels verts. Le surfel s_1 est P_e -simple. Les surfels s_2 et s_4 ne sont pas P_e -simple car il ne sont pas e -adjacents à un surfel de R . Le surfel s_3 n'est pas P_e -simple car la propriété 2 n'est pas respectée pour son e -voisin s_2 .

La squelettisation est effectuée en deux étapes, l'une parallèle et l'autre séquentielle, qui reposent respectivement sur les notions de surfels P_n -simples et de surfels n -simples. Nous rappelons ici une caractérisation locale de ces notions.

Soit n une relation de surfel-adjacence dans $\{e, v\}$ et $\bar{n} = e$ si $n = v$ et $\bar{n} = v$ si $n = e$ (l' e -adjacence représentant l'adjacence entre surfels par arête et la v -adjacence l'adjacence entre surfels par sommet). Soit X un sous-ensemble de surfels de Σ . On dit qu'un surfel est n -intérieur dans X si son n -voisinage (c'est-à-dire l'ensemble des surfels qui lui sont n -adjacents) est inclus dans X . Un surfel est n -simple dans X s'il n'est pas n -intérieur et si ses n -voisins dans X forment un ensemble n -connexe.

On dit aussi qu'un sous-ensemble $Y \subset X$ est n -homotope à X si et seulement si Y peut être obtenu à partir de X par une suppression séquentielle de surfels n -simples.

Intuitivement la notion de surfel simple signifie que l'on peut retirer ce surfel de l'ensemble X sans en modifier la topologie. Ainsi, il est possible d'obtenir un squelette d'un ensemble de surfels par une suppression séquentielle des surfels simples. Néanmoins cette propriété ne permet pas de retirer plusieurs surfels simultanément, le squelette est donc dépendant de l'ordre de suppression des surfels simples. Pour répondre à ce problème la notion de surfels P_n -simples a été introduite.

Pour tout sous-ensemble Y de Σ , notons $A_n(x, Y)$ les surfels y de Y qui sont v -voisins de x et que l'on peut relier à x par un n -chemin dans l'union de x et de son n -voisinage. Soit un ensemble $P \subset X$ et $x \in P$, alors le surfel x est P_n -simple dans X si et seulement si :

1. x est n -simple dans X ;
2. tout n -voisin de x dans P est n -adjacent à un élément de $A_n(x, X \setminus P)$;
3. tout \bar{n} -voisin de x dans P est \bar{n} -adjacent à un élément de $A_{\bar{n}}(x, \Sigma \setminus X)$.

Cette notion de P_n -simplicité amène une propriété intéressante : si $Y \subset X \subset \Sigma$ alors l'ensemble Y est homotope à l'ensemble X si et seulement si l'ensemble $X \setminus Y$ est un ensemble de surfels P_n -simples dans X . En pratique, ceci signifie que l'ordre dans lequel les surfels sont retirés de X pour obtenir Y n'a pas d'influence.

Le procédé de squelettisation proposé dans [Bur02] exploite cette propriété. Étant donné un ensemble $X \subset \Sigma$ seuls les surfels qui sont \bar{n} -adjacents à $\Sigma \setminus X$ peuvent être supprimés de X à chaque étape. Ainsi, pour tester la P_n -simplicité des surfels dans X , l'ensemble P sera l'ensemble des surfels dans X qui sont \bar{n} -adjacents à $\Sigma \setminus X$. C'est-à-dire qu'on va retirer tous les surfels P_n -simples du « contour » de X . On réitère ce procédé tant qu'il reste des surfels P_n -simples.

Une fois cette suppression en parallèle effectuée, il peut toutefois subsister des surfels n -simples. On effectue alors un amincissement en testant séquentiellement les surfels restants et en supprimant ceux qui sont n -simples.

Pour obtenir le squelette qui représente un diagramme de Voronoï topologique de la surface Σ on applique le procédé présenté ci-dessus en prenant comme ensemble X initial l'ensemble $\Sigma \setminus G$, où G

représente l'ensemble des graines choisies en première étape de l'algorithme. On peut voir en blanc sur la figure 2.5 les résultats sur les objets de la figure 2.3.



Figure 2.5 – Squelettisations de la surface d'objets, relativement aux graines choisies (en blanc), qui représentent des diagrammes de Voronoï discrets topologiques (source [BM03]).

La dernière étape de l'algorithme est de construire une triangulation à partir des graines et du diagramme de Voronoï topologique. Pour cela remarquons que le diagramme de Voronoï topologique n'est constitué que de deux sortes de surfels : les surfels des branches, c'est-à-dire ceux qui ont deux voisins dans le squelette, et les surfels des intersections, c'est-à-dire ceux qui ont strictement plus que deux voisins. Chaque surfel intersection est ainsi relié à un autre surfel intersection par le biais d'un chemin C de surfels branches qui sépare deux germes g_1 et g_2 . Si l'on note I_1 et I_2 le barycentre des composantes connexes de surfels intersections auxquelles appartiennent les deux extrémités de C , alors C va engendrer les deux triangles $I_1g_1I_2$ et $I_1g_2I_2$ dans la triangulation finale. Ainsi en parcourant toutes les branches du diagramme de Voronoï topologique on reconstruit une triangulation qui approxime la surface Σ . La figure 2.6 montre quelques résultats.

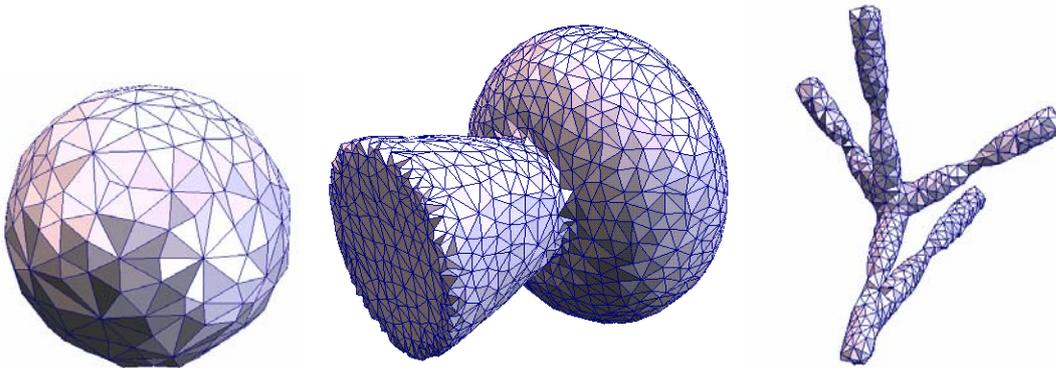


Figure 2.6 – Triangulations obtenues à partir des surfaces discrètes en surfels (source [BM03]).

Nous avons au final une méthode qui fournit une approximation polyédrique sous forme de triangulation et, même s'il reste élevé, le nombre de triangles est inférieur à celui de la technique des Marching Cubes. De plus, en choisissant de guider certaines étapes avec la courbure moyenne de la surface (placement des graines g et choix du paramètre de largeur $l(g)$), la surface polyédrique reste proche de la surface discrète initiale. Malgré tout, si le but est de retrouver de grandes faces, comme les faces naturelles d'objets polyédriques tels que le cube, cette technique ne semble alors pas adaptée.

2.2.2.2 Approche par dualité Diagramme de Voronoï / Triangulation de Delaunay

L'approche décrite ci-après tire son originalité dans le fait qu'elle transpose des concepts de la théorie des diagrammes de Voronoï et triangulations de Delaunay – des classiques en géométrie algorithmique – dans le monde de la géométrie discrète et en particulier des images discrètes 3D. L'utilisation de ces notions permet de garder un bon contrôle sur la triangulation finale obtenue, notamment au niveau de la topologie, et se prête ainsi bien à la polyédrisation d'objets discrets issus du domaine médical.

Dans [Bol07], Dobrina Boltcheva propose plusieurs méthodes permettant de construire une triangulation représentant une approximation polyédrique d'un objet discret. Les approches sont assez similaires et diffèrent principalement sur la notion de frontière considérée pour la polyédrisation de l'objet. L'idée de base consiste à sélectionner un sous-ensemble d'éléments de la frontière discrète comme étant les *nœuds* de la triangulation, de calculer une *approximation discrète du diagramme de Voronoï* de ces nœuds et d'extraire le maillage surfacique par *dualité*, tout en s'assurant que le modèle polygonal de la surface est une 2-variété topologique.

Une première méthode, appelée *Delaunay Discret sur les Voxels* (DDV), va permettre de construire une approximation polyédrique d'un objet discret à partir de sa frontière définie comme un sous-ensemble de voxels, en l'occurrence le bord interne. L'approche se décompose en plusieurs phases, résumées dans la figure 2.7, que nous allons détailler ci-après.

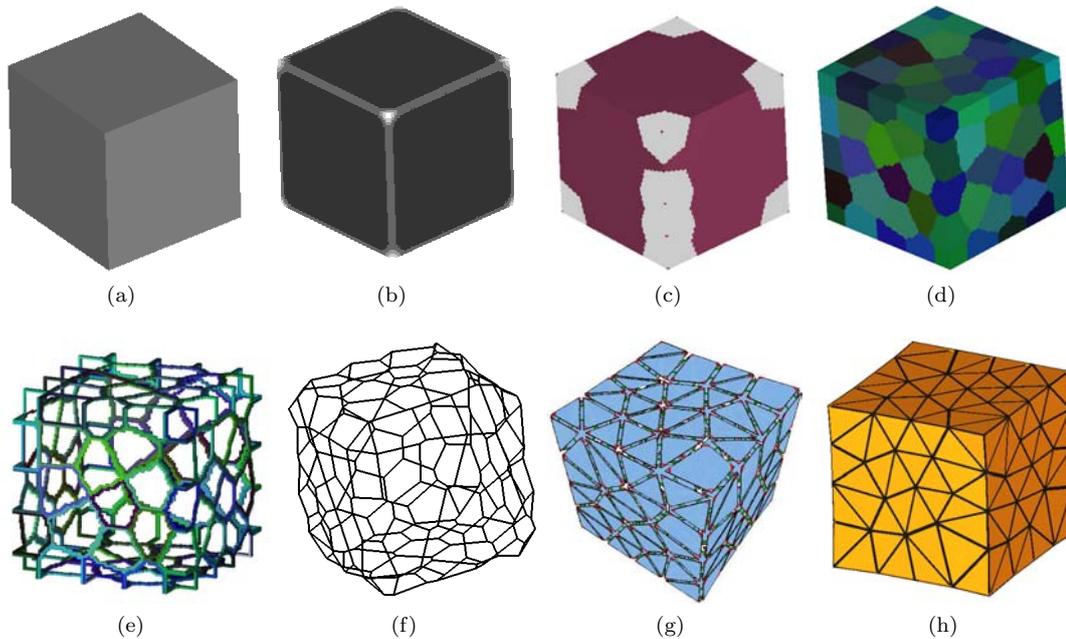


Figure 2.7 – Synthèse des différentes étapes de la méthode Delaunay Discret. (a) La frontière d'un objet discret (b) L'approximation de la courbure (c) Sélection partielle des graines (d) Le diagramme de Voronoï discret (e) Le SKIZ discret (f) Le graphe de Voronoï euclidien (g) La surface topologique duale (h) Le maillage triangulaire final (source [Bol07]).

Dans un premier temps, quelques pré-traitements (opérations de morphologie mathématique, algorithme d'Aktouf [ABP02]) vont être appliqués sur l'objet discret afin d'éliminer certaines cavités et tunnels qui apparaissent à cause des imprécisions des processus d'acquisition ou des erreurs de segmentation.

Ensuite une image binaire de la frontière de l'objet va être calculée. Comme DDV est une méthode de reconstruction travaillant sur le bord interne de l'objet discret, un simple filtrage de l'image avec des

masques de voisinage adaptés aux connexités choisies pour l'objet suffit. Nous présentons ici la méthode pour un objet 6-connexe avec un fond 26-connexe, mais cela reste valable pour n'importe laquelle des paires de Jordan. Ainsi, les voxels de la frontière de l'objet discret seront les voxels de l'objet ayant un voisin du fond dans leur 26-voisinage. C'est-à-dire que l'on extrait le 6-bord interne de l'objet. Il est à noter que cette notion de frontière ne définit pas forcément une surface pour tout objet discret. En particulier, aux endroits très fins, cette frontière ne représente pas forcément une surface séparante au sens de Jordan. Néanmoins, l'utilisation des nombres topologique⁶ permet en pratique de décider si ce bord définit bien une surface ou pas. Les résultats expérimentaux de [Bol07] tendent ainsi à montrer que si le bord interne ne contient pas de points de courbes⁶ alors celui-ci définit bien une surface discrète 6-connexe séparante.

L'étape suivante de la méthode va consister à choisir un ensemble de nœuds de la frontière qui vont représenter les nœuds du futur maillage. L'idée est d'obtenir un ensemble de graines distribué de manière quasi-uniforme afin d'obtenir un maillage final assez régulier. De plus, la courbure moyenne aux points de la frontière va être utilisée pour influencer le placement des graines, dans le but de donner une priorité plus importante aux zones plus courbées.

Cette fois-ci, le calcul de la courbure moyenne s'effectue grâce à une formule proposée par J.W. Bullard et co. [BGCF95] :

$$H(p) \approx \frac{4v}{\pi r^4} - \frac{8}{3r}$$

où H représente la courbure moyenne en un point p de la frontière, r le rayon d'une sphère centrée au point p et v le nombre de voxels de l'objet contenus dans cette sphère. Une fois la courbure calculée en chaque point de la frontière, on effectue les opérations suivantes :

1. on choisit comme graine le point non marqué ayant la courbure moyenne la plus élevée ;
2. on effectue une croissance de région autour de cette nouvelle graine selon une approximation de la distance géodésique par un masque de chanfrein $\langle 16, 21, 28 \rangle$ et on marque les points visités ;
3. on stoppe la croissance dès que tous les points visités sont à une distance de la graine supérieure à un certain *rayon de résolution* fixé ;
4. s'il reste des points non marqués on recommence à l'étape 1 ; sinon on arrête le processus, toutes les graines ont été choisies.

À partir de l'image des graines calculée précédemment, une approximation discrète des régions de Voronoï de la frontière va être déterminée. Pour cela, une croissance simultanée de régions va être effectuée à partir des graines selon une approximation de la distance géodésique par un masque de chanfrein $\langle 16, 21, 28 \rangle$. Cette fois-ci chaque point de la frontière va être marqué du même label que celui de la graine dont il est la plus proche (relativement à la distance géodésique). On obtient ainsi un partitionnement de la frontière en régions telles que la région associée à une graine contient tous les points plus proches de cette graine que de toute autre graine (cf. figure 2.7(d)) – ce qui correspond bien à la notion de diagramme de Voronoï, ici dans l'espace discret.

Ce diagramme de Voronoï discret permet d'obtenir une structure discrète appelée *SKIZ* (*SKELeton by Influence Zones*). Ce SKIZ est en fait formé de tous les points qui ont un voisin dont le label est différent du leur. C'est en quelque sorte le « bord » du diagramme de Voronoï discret précédent (cf. figure 2.8(a)).

Un *sommet primaire* de cardinalité $n (> 3)$ du SKIZ est un voxel dont le 6-voisinage (le voxel inclus) contient des voxels de n labels différents. Et une arête du SKIZ est définie comme un ensemble de voxels dont chaque voxel a un 6-voisinage qui contient des voxels d'exactly deux labels différents (cf. figure 2.8(b)).

L'étape suivante de l'algorithme d'approximation polyédrique va être de transformer le SKIZ en un graphe de Voronoï adapté à une transformation par dualité. Dans cette optique, le graphe de Voronoï devrait respecter certaines contraintes énoncées dans la propriété suivante :

6. Le lecteur intéressé par ces notions pourra consulter [MBA93] pour plus d'informations.

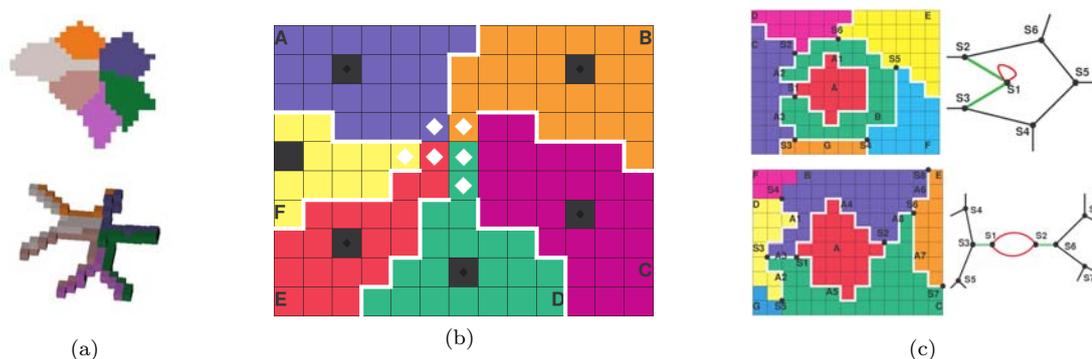


Figure 2.8 – (a) Intersection de régions de Voronoï et le SKIZ correspondant. (b) Projection dans le plan de six régions de Voronoï développées autour des graines en noir. Les sommets primaires du SKIZ sont représentés en losanges blancs. Les arêtes du SKIZ sont les voxels adjacents par arêtes aux lignes blanches. (c) Deux configurations de régions de Voronoï induisant des anomalies topologiques dans le graphe de Voronoï (source [Bol07]).

Propriété 2.1 *Le modèle polygonal dual d'un graphe de Voronoï $GV = (V, E)$ est une 2-variété combinatoire connexe sans bord, si et seulement si :*

1. GV est connexe ;
2. tout sommet $v \in V$ de cardinalité n , est de degré n ;
3. GV est un graphe simple (sans multi-arêtes).

On définit ainsi un sommet du graphe de Voronoï comme étant une composante 6-connexe maximale de sommets primaires dans le SKIZ. Si l'on définit une arête du graphe de Voronoï comme étant une arête du SKIZ, des anomalies topologiques peuvent apparaître dans le graphe de Voronoï (cf. figure 2.8(c)). En effet des régions du diagramme « encerclant » d'autres régions ou bien des régions partageant plus d'une frontière commune, par exemple, induisent des boucles ou des arêtes multiples dans le graphe. La condition 3 de la propriété 2.1 est donc contredite. La solution adoptée ici est de définir comme arêtes du graphe de Voronoï les arêtes du SKIZ qui n'induisent pas d'anomalies topologiques. Les autres ne définissent pas des arêtes du graphe de Voronoï mais permettent de construire une liste de sommets critiques du graphe qui impliquera des corrections à faire dans le maillage dual. En résumé, on construit un graphe de Voronoï dont le dual ne sera pas forcément une 2-variété combinatoire connexe sans bord, mais on modifiera si nécessaire le maillage dual pour corriger les problèmes locaux.

À ce stade de la méthode, le graphe de Voronoï n'est peut-être pas connexe. Il peut exister des sommets dits *invalides*, c'est-à-dire de degré inférieur à leur cardinal. En pratique, il a été décidé que si le nombre de tels sommets est inférieur à 0.1%, le maillage contiendra un petit nombre de trous et des corrections pourront être envisagées. Dans le cas contraire, l'algorithme s'arrête et il vaudra mieux recommencer avec un rayon de résolution plus petit.

Si le graphe ne contient aucun sommet invalide (ou très peu), le maillage dual de ce dernier (représenté sous forme d'une 2-G-carte) va être construit de la façon suivante : chaque sommet valide du graphe de Voronoï génère une face polygonale dont les sommets sont les graines de même label que les labels des sommets primaires composant le sommet du graphe de Voronoï (cf. figure 2.9(a), les brins des sommets sont cousus par α_0 et α_1). Une arête du graphe de Voronoï définit des coutures par α_2 entre les brins des faces issues des sommets de l'arête (cf. figure 2.9(b)).

Si le graphe de Voronoï ne contient aucun sommet invalide, alors la 2-G-carte obtenue est une 2-variété sans bord et une simple triangulation des faces polygonales de cette dernière permet d'obtenir la triangulation désirée, et termine ainsi l'algorithme (figures 2.7(g) et (h)).

Si au contraire le graphe contient des sommets invalides, la 2-G-carte a des bords, et une étape de correction s'avère nécessaire. Cette dernière consiste à ajouter les *couvercles* définis par les trous et à

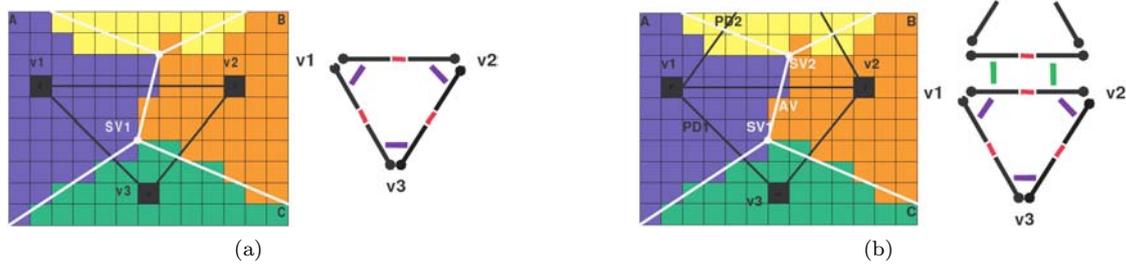


Figure 2.9 – (a) Dual d'un sommet valide du graphe. (b) Dual d'une arête du graphe (source [Bol07]).

triangler ces derniers. Pour un sommet critique isolé du bord, la triangulation est faite en ajoutant une ombrelle de triangles centrée sur ce sommet. Pour un ensemble de sommets critiques isolés du bord, une ombrelle de triangles centrée sur l'isobarycentre des sommets est ajoutée.

La méthode DDV présentée ci-dessus donne des triangulations proches des triangulations de Delaunay, et satisfait aux critères de validité requis par la plupart des algorithmes de visualisation, simulation, etc... Néanmoins, elle n'est pas adaptée pour traiter des objets fins (comme les veines ou les os). Ceci est dû à la définition de frontière comme sous-ensemble de voxels. C'est pourquoi une autre méthode, *Delaunay Discret sur les pointels* (DDP), a été proposée. La frontière de l'objet est cette fois-ci considérée comme un 2-complexe cellulaire. Le principe et les différentes étapes de la méthode DDP sont les mêmes que pour DDV, en utilisant les définitions de voisinages adaptées au modèle des complexes cellulaires sous-jacents. Cette approche a l'avantage d'avoir une notion de surface séparante bien définie, même au niveau des endroits fins de l'objet. En contrepartie, les structures de données manipulées sont un peu plus complexes.

Les résultats obtenus grâce à ces méthodes de polyédrisation sont visuellement proches des résultats obtenus avec l'approche précédente par squelettisation, avec des petits triangles proches de la surface initiale, mais ici de bonnes propriétés topologiques sont assurées. Si ces méthodes se prêtent bien à la polyédrisation d'objets issus du domaine médical, lorsqu'il s'agit de polyèdres avec des grandes faces plates, il semble plus judicieux, si nous désirons retrouver de grands segments correspondant à ces faces, de considérer des polyédrisations en morceaux de plans discrets. C'est ce que nous présentons dans la suite de ce chapitre.

2.3 Méthodes réversibles se basant sur des primitives discrètes

Au contraire des deux méthodes précédentes reposant sur un diagramme de Voronoï discret, qui fournissaient des approximations polyédriques du bord d'un objet discret, les méthodes présentées dans cette section sont des reconstructions exactes. En effet, étant donné un modèle de discrétisation, ces méthodes assurent que la discrétisation du polyèdre euclidien reconstruit est la même que l'objet discret de départ. C'est pourquoi on les qualifie de *polyédrisations réversibles*.

Les approches décrites ci-après, populaires dans la communauté française de géométrie discrète, peuvent s'abstraire en un procédé composé de deux étapes principales.

1. Une étape de *décomposition* de la surface de l'objet discret en un ensemble de morceaux de plans discrets, qu'on appelle communément une *segmentation*.
2. Une étape de *polygonalisation* visant à obtenir un ensemble de *facettes* polygonales à partir des morceaux de primitives discrètes obtenus à l'étape précédente.

Si le procédé se résume en quelques lignes, la description précédente laisse tout de même place à de nombreuses questions :

- Quelle est la surface de l'objet considérée ?

- Comment savoir si les points de la surface appartiennent à un morceau de plan discret ?
- Doit-on imposer des contraintes (taille, forme, connexité, nombre, etc...) sur les segments ?
- L'ordre de parcours des points de la surface pour développer les segments est-il important ?
- Quel représentant euclidien doit-on choisir quand il y a plusieurs possibilités ?
- Comment « recoudre » les faces entre elles ? etc...

C'est d'ailleurs sur les réponses apportées à ces différentes questions que se différencient les approches présentées dans la suite.

2.3.1 Polyédrisation en polyèdre discret standard

Une première méthode a été proposée par L. Papier et J. Françon dans [FP99, Pap99]. Cette méthode n'aboutit pas vraiment à la transformation en un polyèdre euclidien, dans le sens où le résultat n'est pas un ensemble de facettes planes polygonales. Les auteurs utilisent plutôt la notion de *polyèdre (discret) standard*. Un polyèdre standard est un ensemble de *faces standards*, chacune définie comme un ensemble de surfels formant un disque topologique et telle que les pointels incidents aux surfels de cette face appartiennent à un plan standard; un ensemble *d'arêtes standards*, c'est-à-dire un chemin de lignels appartenant au bord d'exactly deux faces standards; et enfin un ensemble de *sommets standards* qui sont les pointels incidents à au moins trois arêtes standards. La figure 2.10(b) montre un exemple d'un tel polyèdre.

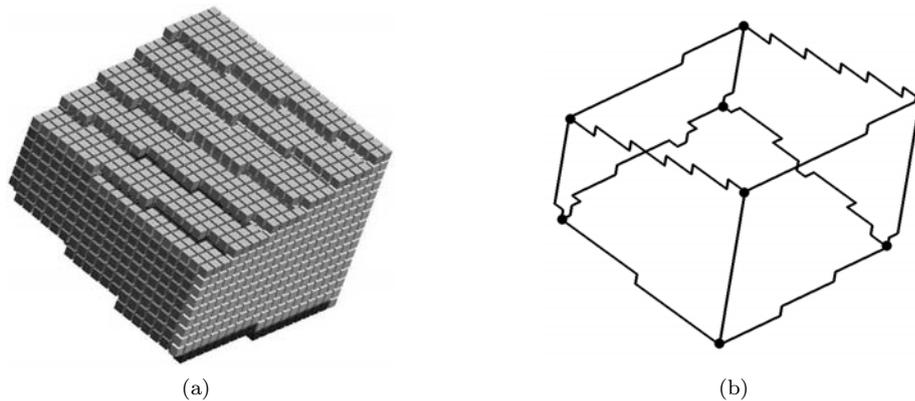


Figure 2.10 – (a) Un objet discret et, après rotation d'un quart de tour, (b) un polyèdre standard associé ayant les points noirs comme sommets standards (source [Pap99]).

La méthode de polyédrisation consiste à décomposer les surfels du bord d'un objet discret en faces standards pour former un polyèdre standard qui sera considéré comme une représentation polyédrique de l'objet discret. On travaille ici sur le bord en surfels de l'objet et il est supposé que celui-ci est une 2-variété combinatoire sans bord.

L'algorithme de facétisation consiste à faire « grossir » une face initiale en ajoutant à chaque fois un surfel. À chaque étape, les surfels de la face standard en cours de construction doivent former un disque topologique et les pointels incidents aux surfels doivent appartenir à un plan discret standard.

Le choix de la graine pour débiter une croissance de la face, s'effectue en priorité sur les ombrelles qui n'appartiennent à aucune face, et sinon en choisissant le premier surfel rencontré qui n'appartient à aucune face.

La croissance des faces s'effectue par un parcours en largeur du graphe des surfels du bord, à partir de la graine. Le surfel est rajouté à la face standard si et seulement si :

1. les pointels incidents aux surfels appartiennent à un plan standard. Pour vérifier cette contrainte les auteurs utilisent un algorithme de reconnaissance de morceaux de plans discrets standards [FST96], basé sur l'algorithme de Fourier-Motzkin [Fou26, Mot36] (un algorithme permettant de résoudre des systèmes d'inéquations linéaires);

2. la face en construction doit rester homéomorphe à un disque topologique (cf. figure 2.11).

Lorsqu'aucun surfel ne peut être ajouté à la face, on recommence avec une nouvelle graine tant qu'il existe des surfels n'appartenant à aucune face standard.

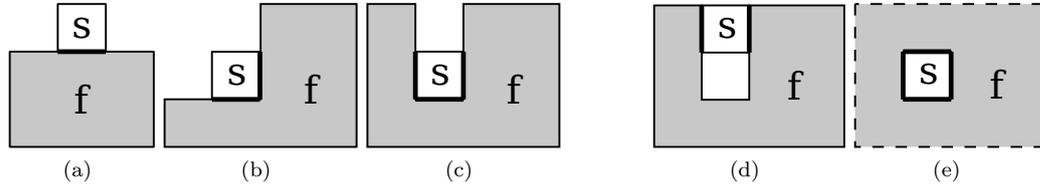


Figure 2.11 – Topologie des 5 configurations possibles lors de l'ajout d'un surfel s à la face courante f . Seules les configurations (a),(b) et (c) sont autorisées (source [Pap99]).

Les résultats obtenus avec un tel algorithme ne sont pas satisfaisants aux yeux des auteurs car la segmentation en faces standards obtenue ne respecte pas toujours les discontinuités de l'objet – telles que les arêtes naturelles des objets polyédriques (cf. figure 2.12(c)). Pour détecter ces arêtes naturelles, l'utilisation des vecteurs normaux aux surfels est préconisée. Ainsi, les vecteurs normaux d'ordre 0 (voir la section 4.2.2.1 page 83 pour plus de détails sur la notion d'ombrelle et de normale d'ordre n) vont être utilisés pour détecter :

1. les paliers (représentant la notion de zone « plate ») ;
2. les lignels arêtes (représentant les discontinuités).

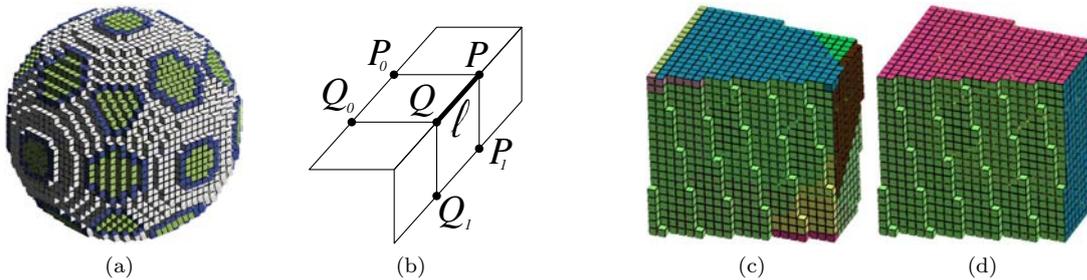


Figure 2.12 – (a) Les paliers d'une sphère de rayon 15. (b) Les notations pour décrire un lignel arête. Les résultats de la facétisation d'un hexaèdre avec (c) l'algorithme initial et (d) l'algorithme amélioré prenant en compte les paliers et les lignels arêtes (source [Pap99]).

Les paliers (figure 2.12(a)) sont des zones de l'objet où la normale d'ordre 0 est la même en tous les points de la zone. En se référant à la figure 2.12(b), un *lignel arête* (lignel ℓ , en gras sur la figure) est un lignel tel que les normales d'ordre 0 en P , P_0 et P_1 sont différentes et celles en P_0 et P_1 ne sont pas dans le même 48° d'espace, et de même pour les points Q_i . L'algorithme de facétisation est modifié pour choisir en priorité les graines sur des paliers et le parcours des surfels est modifié pour ne pas « franchir » les lignels arêtes. On peut voir le résultat sur la figure 2.12(d). On peut remarquer que pour ce polyèdre, la facétisation obtenue respecte les discontinuités de l'objet et on obtient 6 faces standards, ce qui correspond bien au résultat qu'on attendrait pour un tel objet.

Dans [Pap99], deux algorithmes de visualisation des polyèdres discrets sont proposés. Le premier est simplement une exploitation de la librairie OpenGL pour afficher le polyèdre standard en 3D. Une triangulation implicite est faite par cette librairie lors du rendu des faces standards (qui sont généralement non convexes et non coplanaires, au sens euclidien). Le résultat est montré à la figure 2.13(a) et (b). Le deuxième algorithme consiste à transformer le polyèdre standard en un *polyèdre semi-discret* en remplaçant les arêtes standards par des segments euclidiens joignant les deux extrémités de l'arête (on perd alors

la réversibilité). Encore une fois les sommets des facettes obtenues ne sont généralement pas coplanaires, mais ce polyèdre semi-discret a des facettes à bords plus réguliers, comme le montre la figure 2.13(c) et (d).

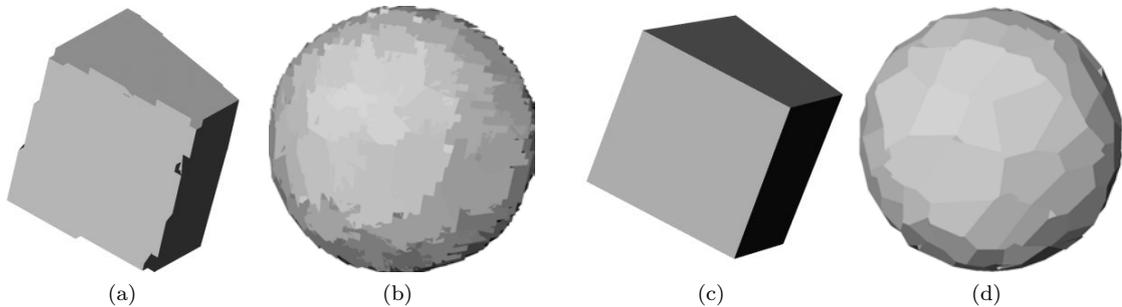


Figure 2.13 – Résultats des méthodes de visualisation sur (a), (b) de polyèdres discrets standard et (c), (d) leur version semi-discret (sources [Pap99, FP99]).

Ceci ne résout néanmoins pas le problème de la polyédrisation, si le but était d’obtenir une facette euclidienne par facette standard. De plus, l’utilisation de cette technique sur des objets bruités paraît difficilement envisageable, car la détection des paliers et des lignels arêtes repose sur les normales à l’ordre 0 et serait beaucoup trop sensible aux perturbations du bord de l’objet. Le sens de ces notions serait alors très limité.

2.3.2 Méthodes exploitant la préimage

Dans [Siv04] Isabelle Sivignon propose différents algorithmes de polyédrisation d’objets discrets 6-connexes en utilisant la notion de préimage présentée en 1.8.2.1 :

- l’un basé sur l’intersection de demi-espaces définis par des plans euclidiens ;
- un autre permettant d’aboutir à une « *reconstruction face par face réversible* » de la surface, mais qui n’est pas un polyèdre euclidien (une pseudo-polyédrisation donc) ;
- et un dernier par simplification d’un Marching Cubes.

Ces différents algorithmes font tous usage d’une segmentation de la surface de l’objet discret en morceaux de plans discrets. Ainsi, plusieurs méthodes de segmentation sont proposées dans [Siv04, SDC04a, SDC04b, CGS04]. Nous rappelons brièvement leur fonctionnement dans les paragraphes suivants, avant de présenter les trois méthodes de polyédrisation citées ci-dessus.

Segmentation en plans naïfs avec étiquetage des voxels : La première méthode de segmentation consiste à décomposer le bord d’un objet discret 6-connexe en un ensemble de morceaux de plans naïfs. Le but est d’étiqueter chaque voxel du bord interne de l’objet par un label qui représente le morceau de plan discret naïf auquel il appartient. De plus, une contrainte de connexité sur les morceaux de plans obtenus est ajoutée, à savoir : la projection des voxels d’un morceau de plan selon l’axe principal de ce plan doit être un ensemble de pixels 4-connexe. L’algorithme 2.1 résume la méthode.

L’idée est d’utiliser les tricubes comme brique de base de la segmentation et de la reconnaissance (en effet, on sait que tout plan discret naïf peut-être décomposé en tricubes) afin de mieux contrôler le développement des faces discrètes. Ainsi, on commence par choisir un voxel v_0 , germe d’une nouvelle face discrète f , parmi les voxels du bord non étiquetés et on initialise une liste de voxels à visiter $\text{vois}(f)$ avec v_0 . La normale élémentaire l de chaque surfel de v_0 appartenant à la surface de l’objet discret va définir une direction principale pour la face courante, et aussi déterminer quelle préimage utiliser pour l’algorithme de reconnaissance de plans naïfs. Pour chacune de ces directions, on va essayer d’étendre f en répétant les opérations suivantes tant que $\text{vois}(f)$ n’est pas vide : on récupère un voxel v de $\text{vois}(f)$ et on recherche un tricube tc centré en v dont les voxels appartiennent au bord de l’objet. S’il est trouvé, que les voxels de $f \cup tc$ appartiennent à un plan discret (vérifié grâce à l’algorithme incrémental de

Algorithme 2.1 : Algorithme de segmentation en plans naïfs : étiquetage des voxels**Entrées** : B , les voxels (non étiquetés) du bord interne d'un objet discret 6-connexe**Sorties** : Segmentation de B en plans discrets naïfs (étiquetage des voxels)**Variables** : f , la face discrète courante v , le voxel courant $\text{vois}(f)$, les voxels à traiter pour la face f

```

1  début
2  tant que tous les voxels de  $B$  ne sont pas étiquetés faire
3  |    $f \leftarrow$  un voxel germe  $v_0$  de  $B$ 
4  |    $\text{vois}(f) \leftarrow v_0$ 
5  |   pour chaque direction  $l$  telle que  $v_0 + l$  n'appartient pas à l'objet faire
6  |   |   tant que  $\text{vois}(f) \neq \emptyset$  faire
7  |   |   |    $v \leftarrow$  retirer un voxel de  $\text{vois}(f)$ 
8  |   |   |   Sauvegarder les paramètres de  $f$ 
9  |   |   |    $\text{tc} \leftarrow \text{RECHERCHE\_TRICUBE}(v, l)$  //  $\text{tc}$  est le tricube éventuellement trouvé
10 |   |   |   si  $\text{tc}$  existe et  $f \cup \text{tc}$  est un morceau de plan discret alors
11 |   |   |   |   Mettre à jour les paramètres de  $f$ 
12 |   |   |   |   pour chaque voxel  $v'$  de  $\text{tc}$  faire
13 |   |   |   |   |   si  $v'$  n'est pas étiqueté par un plan et  $v'$  a un voisin 4-connexe dans la
14 |   |   |   |   |   |   projection orthogonale de  $f$  selon  $l$  alors
15 |   |   |   |   |   |   |   Étiqueter  $v'$  comme appartenant à  $f$ 
16 |   |   |   |   |   |   |   Ajouter  $v'$  à  $\text{vois}(f)$ 
17 |   |   |   |   |   sinon
18 |   |   |   |   |   |   Restaurer les paramètres de  $f$ 
19 |   |   |   si  $(v_0$  n'est pas étiqueté) alors //  $v_0$  n'est centre d'aucun tricube
20 |   |   |   |   Étiqueter  $v_0$  à  $-1$ 
21 fin

```

Vittone [Vit99]) et que chaque voxel à un voisin 4-connexe dans la projection de f selon l , alors les voxels de ce tricube sont étiquetés comme appartenant à f et ajoutés à $\text{vois}(f)$. Si un voxel n'est pas le centre d'un tricube compatible avec les paramètres actuels de la face f alors il est étiqueté à -1 pour signifier qu'il a été visité, mais qu'il peut être ré-étiqueté plus tard comme appartenant à une autre face discrète.

Comme on peut s'en douter, plusieurs choses dans cet algorithme vont influencer la segmentation obtenue. Tout d'abord le choix de la graine v_0 , mais aussi le parcours du voisinage autour de v_0 . Dans ses travaux, I. Sivignon propose de prendre comme graine initiale un voxel choisi aléatoirement à la surface de l'objet, puis pour les graines suivantes le dernier voxel rencontré qui n'a pas pu être ajouté à la face discrète courante. Ensuite, au niveau du parcours du voisinage, plusieurs solutions sont envisagées. Elles sont illustrées à la figure 2.14. Comme on peut le voir, l'ordre dans lequel sont visités les voxels d'un tricube et l'ordre dans lequel les voxels sont retirés de $\text{vois}(f)$ modifie radicalement la forme des faces discrètes.

Une autre classe de parcours est aussi utilisée dans ses travaux [SDC04a]. Elle consiste à parcourir les voxels selon un système de pondération croissante. Chaque voxel d'un tricube est doté d'un poids (relatif à sa position dans le tricube) et, au cours du parcours, ces derniers sont insérés dans la liste (qui est en fait une file de priorité) $\text{vois}(f)$ en prenant en compte leurs poids et le poids du voxel centre du tricube (cf. figure 2.15). À chaque fois, le voxel suivant à visiter est le voxel de poids le plus faible dans $\text{vois}(f)$.

Les tests expérimentaux effectués dans [Siv04] tendent à montrer que les faces discrètes sont plus régulières lorsqu'on utilise un parcours en quelque sorte « isotrope » (comme le parcours pondéré précédent

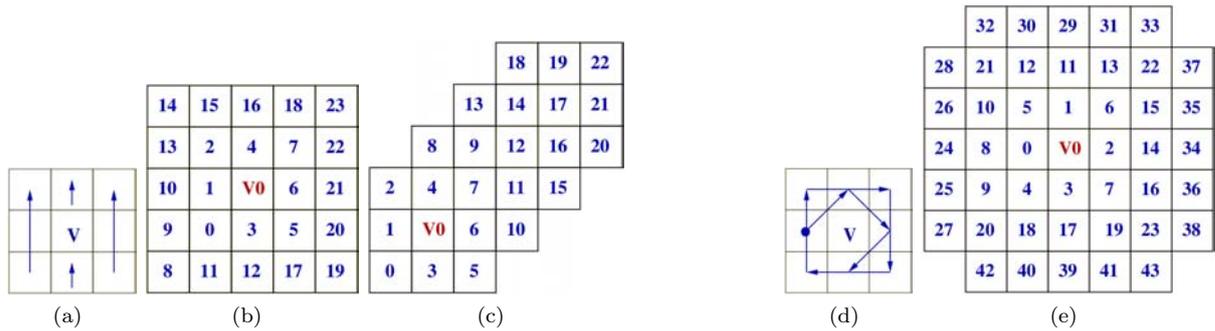


Figure 2.14 – Influence de l’ordre de parcours des voxels lors du développement des faces discrètes. Illustration dans la projection suivant la direction principale de la normale à la face. (a) Les voxels d’un tricube centré en V sont parcourus de gauche à droite dans le sens indiqué par les flèches. (b) Le premier voxel de la liste $\text{vois}(f)$ est choisi ; les numéros indiquent l’ordre dans lequel les voxels sont ajoutés. (c) Idem sauf que le dernier voxel de $\text{vois}(f)$ est choisi. (e) Croissance de la face lorsque les voxels sont ajoutés en suivant l’ordre de parcours du tricube indiqué en (d) et en choisissant le premier voxel dans $\text{vois}(f)$ (source [Siv04]).

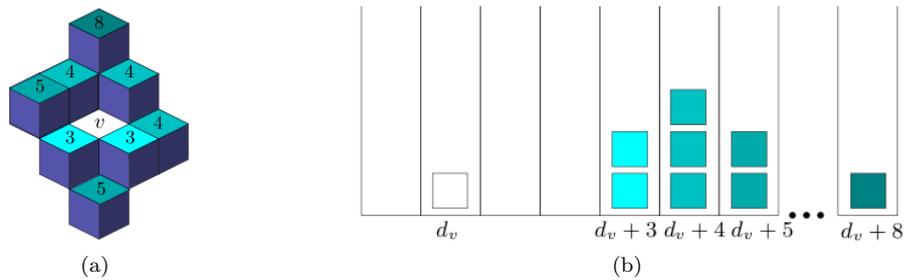


Figure 2.15 – Illustration de l’ajout des voxels d’un tricube centré en v dans une file de priorité : les poids de chaque voxel du tricube sont indiqués en (a) et l’insertion dans la file de priorité est représentée en (b) (source [Siv04]).

ou le parcours illustré à la figure 2.14(e)). Ces derniers semblent ainsi plus adaptés à une segmentation du bord de l’objet, dans un cadre de polyédrisation.

Segmentation en plans naïfs avec étiquetage des surfels : Pour traiter les surfaces d’objets discrets considérées comme un ensemble de surfels, l’algorithme précédent n’est plus utilisable tel quel. Néanmoins il peut être étendu pour étiqueter les surfels, comme nous allons le voir tout de suite. Le but de la segmentation avec étiquetage des surfels est d’arriver à une décomposition en morceaux de plans discrets tels que : si un surfel $s = \{v, v'\}$ (commun à deux voxels v et v' tel que v appartient à l’objet et v' n’appartient pas à l’objet) est étiqueté comme appartenant à un plan discret P , alors les plans euclidiens solutions pour P coupent le segment $[v, v']$. Ainsi, pour un sous-ensemble de surfels de même étiquette, il existe un plan coupant tous les intervalles définis par les deux voxels associés aux surfels.

En se rappelant que dans l’algorithme 2.1 l’étiquetage des voxels est fait suivant une direction de reconnaissance l (définie en fonction du voxel germe v_0), on voit que cela permet de reconnaître une face discrète composée de voxels v dont le surfel $\{v, v+l\}$ est à la surface de l’objet, et telle qu’il existe un plan euclidien coupant tous les segments $[v, v+l[$. C’est la notion de *reconnaissance directionnelle dans la direction l* introduite dans [Siv04].

Ainsi, pour arriver à un étiquetage des surfels, un algorithme en deux passes est proposé dans [SDC04b,

Siv04]. La première passe de l'algorithme va simplement consister à appliquer l'algorithme 2.1 en étiquetant les surfels $\{v, v'\}$ au lieu d'étiqueter les voxels v . On obtient alors un étiquetage comme celui illustré à la figure 2.16(a). Puis dans une deuxième passe, ce sont les *surfels intérieurs*⁷ qui sont étiquetés (cf. figures 2.16(b) et (c)).

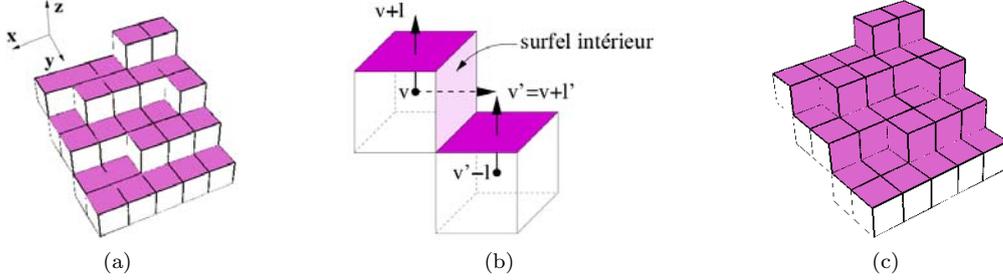


Figure 2.16 – Étiquetage des surfels de la surface d'un morceau de plan de paramètres $(1, 3, -5, 0)$: (a) après une première passe utilisant l'algorithme de reconnaissance directionnelle dans la direction \vec{z} ((b) la notion de surfel intérieur) et (c) après la deuxième passe étiquetant les surfels intérieurs (source [Siv04]).

Segmentation en plans standards avec étiquetage des pointels : La troisième méthode de segmentation proposée dans [Siv04] consiste à décomposer la surface d'un objet discret en morceaux de plans discrets standards avec un étiquetage de pointels. De plus, pour que les faces discrètes obtenues soient utilisables dans leur algorithme de polyédrisation, les auteurs introduisent une nouvelle contrainte, à savoir : chaque face discrète doit posséder un unique bord, et doit donc être homéomorphe à un disque topologique.

Pour cette méthode, I. Sivignon montre en fait qu'il existe une transformation permettant de passer d'une segmentation en plans naïfs avec étiquetage des surfels à une segmentation en plans standard avec étiquetage des pointels. Ainsi on peut directement obtenir la seconde segmentation à partir de la première grâce à la propriété suivante :

Propriété 2.2 ([Siv04]) Soit f un morceau de plan discret naïf reconnu à la surface d'un objet discret 6-connexe par l'algorithme de segmentation en plans naïfs avec étiquetage des surfels. Alors si le plan euclidien de paramètres (a, b, c, μ) est solution pour les voxels de f , le plan de paramètres (a, b, c, μ') est solution pour les pointels appartenant aux surfels de f dans la grille translatée du vecteur $t = (\frac{1}{2}, \frac{1}{2}, \frac{1}{2})$, avec :

$$\mu' = \tau - \frac{|a| + |b| + |c|}{2} - v.(a, b, c)$$

$$\text{où } \tau = \begin{cases} \mu & \text{si l'orientation est standard} \\ \mu + 1 & \text{si l'orientation est non standard} \end{cases} \quad \text{et} \quad v = -\frac{1}{2}[\text{sgn}(a), \text{sgn}(b), \text{sgn}(c)] + t$$

La propriété d'être homéomorphe à un disque topologique est maintenue incrémentalement au cours du développement des faces discrètes en s'assurant que l'ajout des voxels d'un tricube ne relie pas deux composantes connexes distinctes de la face courante.

Nous pouvons à présent introduire les différentes méthodes de polyédrisation proposées par I. Sivignon et ses coauteurs.

7. Soit f une face discrète reconnue dans la direction l et $s = \{v, v'\}$ un surfel de la surface de l'objet tel que $v' = v + l'$ avec $l' \neq l$. s est un *surfel intérieur* à la face f si et seulement si les deux surfels $\{v' - l, v'\}$ et $\{v, v + l\}$ appartiennent à f .

2.3.2.1 Intersection de demi-espaces

La méthode fondée sur l'intersection de demi-espaces s'appuie sur une segmentation en plans discrets avec l'une des méthodes présentées ci-dessus.

Pour chaque face discrète reconnue, un représentant euclidien (un plan euclidien dont la discrétisation contient les voxels/surfels/pointels de la face) est choisi parmi les différentes possibilités issues du polyèdre de l'espace dual construit au cours de la reconnaissance. L'intersection des demi-espaces créés par ces représentants forme un polyèdre convexe dont les faces représentent la surface polyédrique recherchée. Les figures 2.17 et 2.18 présentent quelques résultats. Il apparaît clairement qu'avec cette méthode les reconstructions obtenues ne seront représentatives que pour les objets *discrets convexes*.

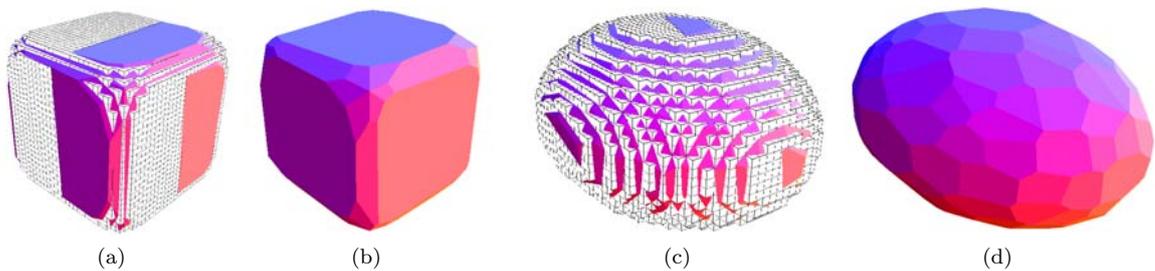


Figure 2.17 – Reconstruction de surfaces discrètes par intersection de demi-espaces. (a), (c) Superposition de l'objet discret et de la surface reconstruite ; et (b), (d) la surface seule (source [Siv04]).

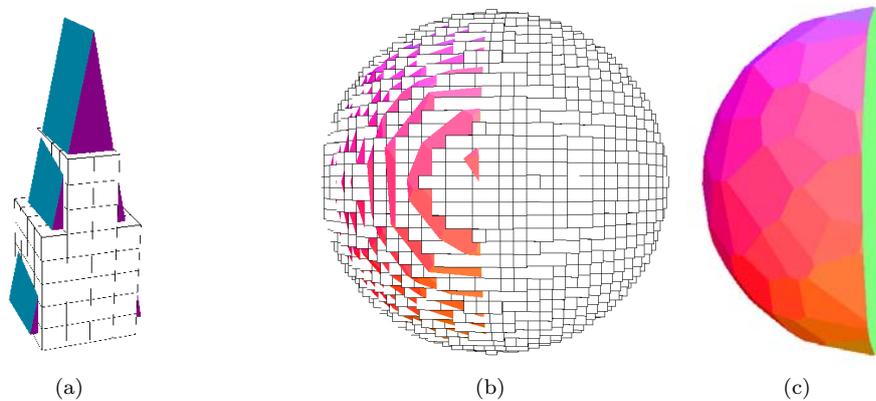


Figure 2.18 – Effets non désirés de la reconstruction : (a) l'intersection des représentants s'effectue loin de l'objet et (b), (c) une petite facette obtenue lors de la phase de reconnaissance restreint l'intersection (source [Siv04]).

On peut aussi voir sur la figure 2.18 que quelques problèmes, inhérents à cette méthode, peuvent apparaître dans la reconstruction. En effet, à des endroits fins ou « pointus » de l'objet par exemple, les représentants euclidiens peuvent se couper très loin du bord de l'objet discret. Ainsi la discrétisation de la surface polyédrique obtenue contiendra des points qui n'étaient pas dans l'objet discret initial. La méthode n'est donc pas réversible, ce qui la place dans la catégorie des méthodes *d'approximations polyédriques*. D'autre part de petites facettes obtenues lors de la phase de reconnaissance peuvent perturber la reconstruction, comme le montrent les figures 2.18(b) et (c). Une petite facette dont la normale a mal été estimée va restreindre l'intersection des demi-espaces et une partie de l'objet risque d'être perdue.

2.3.2.2 Reconstruction faces par faces

La reconstruction *faces par faces* est une reconstruction de la surface d'un objet discret sous forme d'un ensemble de facettes polygonales. Cette méthode peut être qualifiée de pseudo-polyédrisation car le résultat n'est en réalité pas une surface polygonale, mais un ensemble de facettes polygonales disjointes dont la discrétisation (ici la discrétisation *standard*) redonne l'objet discret initial; cette méthode est donc réversible.

La reconstruction *faces par faces* considère la surface en surfels d'un objet discret 6-connexe; on opère ainsi sur une surface discrète qui est une variété combinatoire de dimension 2.

Dans un premier temps, une segmentation en plans standards sur les pointels de cette surface est effectuée grâce à l'algorithme idoine. On obtient ainsi un ensemble de faces discrètes homéomorphes à des disques topologiques. D'une part, comme les faces sont des morceaux de plans discrets, on sait qu'il existe un plan euclidien P qui coupe chacun des voxels. D'autre part, comme la surface est une 2-variété combinatoire, chaque face a un unique bord. On peut donc calculer, pour chacune d'elle, une face polygonale dont la discrétisation standard est exactement la face discrète.

Pour cela un algorithme de polygonalisation réversible de courbes 3D planaires, présenté dans [SDC05], va être appliqué sur le bord des faces discrètes. Sans trop entrer dans les détails, l'algorithme fonctionne de la façon suivante : on considère les pointels du bord de la face discrète comme un ensemble ordonné de voxels $\{v_1, v_2, \dots, v_n\}$ formant une courbe discrète 6-connexe. Un point réel r_1 est choisi dans le premier voxel v_1 , débutant ainsi un premier segment discret s_1 , et les voxels suivants de la courbe sont ajoutés un à un tant que :

- s_1 est un segment de droite discrète;
- parmi les droites ayant une discrétisation qui contient s_1 , il existe au moins une droite contenue dans le plan euclidien P et qui part de r_1 .

Si l'une de ces deux conditions n'est plus satisfaite, alors on calcule r_2 , le point d'arrêt du segment s_1 , comme un point d'intersection entre le représentant euclidien choisi pour s_1 et le dernier voxel de s_1 . Cette extrémité devient le point de départ d'un nouveau segment réel et on recommence ce procédé jusqu'à ce que tous les points du bord de la face soient traités. Ces étapes sont résumées à la figure 2.19 et les résultats sur des objets discrets sont montrés à la figure 2.20.

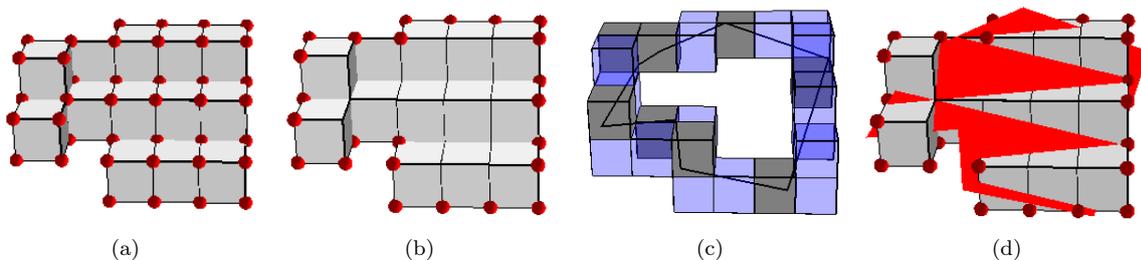


Figure 2.19 – Exemple de déroulement de l'algorithme de polygonalisation d'une face discrète : (a) une face discrète reconnue après segmentation, (b) les pointels du bord de cette face sont représentés en rouge, (c) polygonalisation du bord de la facette, chaque pointel est ici remplacé par le cube de côté 1 dont il est le centre et (d) la face polygonale résultante (source [Siv04]).

Comme on peut le voir on a bien un ensemble de facettes euclidiennes et le résultat visuel de la polygonalisation est proche de l'objet discret initial. On appréhende d'ailleurs très bien la forme de l'objet. Néanmoins ces facettes polygonales restent « décousues » et nous n'avons donc pas une surface polyédrique au final.

2.3.2.3 Simplification d'un Marching Cubes

La méthode présentée ici est une méthode de simplification d'une polyédrisation obtenue par la technique des Marching Cubes à l'aide d'une segmentation en plans discrets.

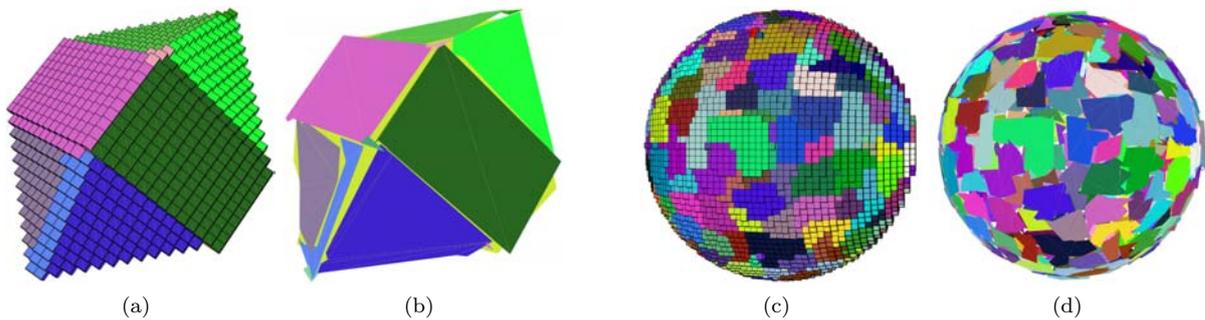


Figure 2.20 – (b), (d) Reconstruction *faces par faces* des objets discrets (a), (c) (source [Siv04]).

On considère un objet discret 6-connexé binaire (composé de voxels *objets* à 1 et de voxels *fonds* à 0). La première étape consiste à effectuer une polyédrisation de cet objet avec l’algorithme du Marching Cubes (décrit en 2.2.1) en utilisant un seuil de 0.5 pour l’isosurface. On sait alors ([Lac96, LM00]) que la surface obtenue est une variété combinatoire de dimension 2, c’est-à-dire fermée, orientée et sans auto-intersection, et que les sommets de la surface triangulée se trouvent exactement au milieu des arêtes de chaque cellule (lorsqu’une extrémité de l’arête est dans l’objet et l’autre non). Alors, si chaque point discret est représenté par un voxel, les sommets de la surface triangulée coïncident avec les centres des surfels de la surface en surfels de l’objet discret. De plus, la propriété suivante est vérifiée :

Propriété 2.3 ([Siv04]) *La surface construite par l’algorithme du Marching Cubes à partir d’un objet discret et d’un seuil de l’intervalle $]0, 1[$ est une polyédrisation réversible de la surface de l’objet discret en considérant la discrétisation OBQ.*

On peut directement déduire de cette propriété que si l’on déplace un sommet v (centre d’un surfel défini par deux voxels : p appartenant à l’objet et q appartenant au fond) de la triangulation issue du Marching Cubes le long du segment $[p, q]$, alors la propriété de réversibilité est conservée et la surface reste une variété combinatoire de dimension 2.

La deuxième étape va donc consister à déplacer les sommets de la triangulation, afin de rendre certains triangles coplanaires. Pour cela, les informations fournies par l’algorithme de segmentation en plans naïfs avec étiquetage des surfels vont être utilisées. D’une part, comme les sommets de la triangulation issue de l’algorithme du Marching Cubes sont les centres des surfels de la surface de l’objet discret, il est naturel d’utiliser cet algorithme de segmentation avec étiquetage des surfels. D’autre part, cette segmentation a la propriété suivante :

Propriété 2.4 ([Siv04]) *Soit un surfel s de la surface d’un objet discret défini par les deux voxels p et q . Alors si s est étiqueté comme appartenant à une face f , tous les plans euclidiens solutions pour f coupent le segment $[p, q]$.*

Ainsi, étant donné une face discrète f , reconnue lors de la segmentation, et P , un plan euclidien solution associé, l’idée est de projeter sur P tous les sommets de la triangulation issue de l’algorithme du Marching Cubes qui sont centres des surfels étiquetés comme appartenant à f – la projection se faisant selon les directions \vec{pq} , pour p et q définis comme auparavant. Nous obtenons ainsi, pour chaque face discrète, un ensemble de triangles coplanaires.

Enfin, une troisième étape va permettre de simplifier cette triangulation en fusionnant ces ensembles de triangles coplanaires. Si l’on appelle triangles *homogènes*, les triangles qui sont formés de sommets appartenant tous à une même face discrète, et triangles *non-homogènes* les autres, alors les ensembles connexes de triangles homogènes sont séparés les uns des autres par des bandes de triangles non-homogènes. Pour

simplifier la triangulation on va donc parcourir les composantes connexes de triangles homogènes et les remplacer par une facette polygonale, sans modifier les triangles non-homogènes.

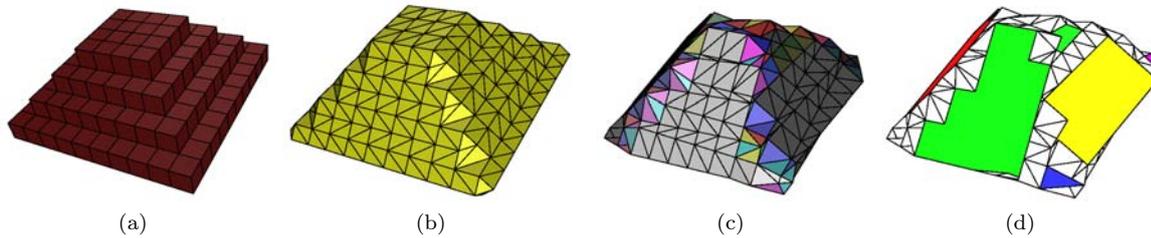


Figure 2.21 – Illustration des différentes étapes de l'algorithme de simplification d'un Marching Cubes : (a) l'objet discret, (b) la triangulation obtenue par Marching Cubes, (c) cette même triangulation après projection des sommets sur les plans solutions obtenus lors de la segmentation et (d) le résultat final après fusion des triangles coplanaires (source [Siv04]).

La figure 2.21 résume les différentes étapes de la méthode et des résultats obtenus sur divers objets discrets sont illustrés à la figure 2.22.

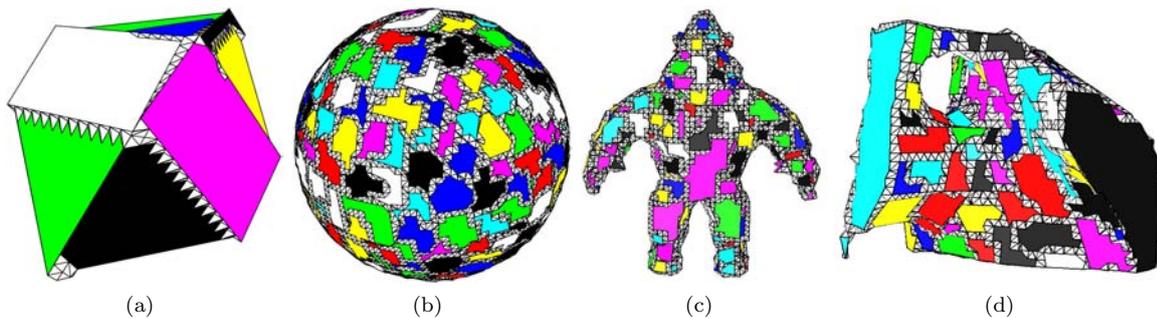


Figure 2.22 – Quelques résultats de l'algorithme de simplification d'un Marching Cubes à l'aide d'une segmentation en plans naïfs par étiquetage des surfels sur (a), (b) des objets mathématiques et (c), (d) sur des objets réels scannés (source [Siv04]).

Comme on peut le remarquer, cette fois-ci la surface obtenue est bien un polyèdre euclidien. On voit que la méthode produit un nombre de facettes bien inférieur à celui obtenu avec un Marching Cubes, tout en restant réversible. Néanmoins la polyédrisation est très hétérogène et de nombreux très petits triangles persistent afin de conserver une « couture » entre les plus grandes faces polygonales.

2.3.3 Reconstruction analytique inversible

Dans [DCA06, Dex06], Martine Dexet propose une méthode de polyédrisation réversible (relativement au modèle standard de discrétisation) dont le principe est inspiré de la méthode de reconstruction faces par faces présentée auparavant. Néanmoins elle propose une solution à l'épineux problème de « couture » des faces polygonales obtenues. On a ainsi une vraie polyédrisation réversible.

On travaille ici avec la surface en surfels d'un objet discret 6-connexe et les points discrets considérés sont les pointels de cette surface. L'idée est en fait de lier les étapes de segmentation et de polygonalisation en contraignant un peu plus la phase de reconnaissance des morceaux de plans discrets qui représentent les faces discrètes. Ainsi, tout comme dans la méthode de reconstruction faces par faces, on commence par reconnaître une première face discrète à partir d'un surfel s de départ. La différence est qu'ici l'algorithme de reconnaissance de morceaux de plans discrets standards utilisant la préimage généralisée (présentée

en 1.8.2.2) est utilisé. De plus les points discrets ajoutés au morceau de plan en cours de reconnaissance sont tous les pointels appartenant au dernier surfel visité. Quand le processus de reconnaissance s'arrête (c'est-à-dire quand la préimage généralisée est vide), un plan euclidien P est choisi dans l'ensemble des solutions calculé dans la préimage et le bord C de la face discrète reconnue est extrait. Un processus de polygonalisation de face est alors appliqué sur C afin d'obtenir le contour d'une face euclidienne contenue dans P et dont la discrétisation standard est identique au morceau de plan discret reconnu. Un nouveau surfel de départ, non traité, est alors choisi et les processus de reconnaissance et de polygonalisation sont à nouveau appliqués, jusqu'à ce que tous les surfels de la surface soient traités. Là où le procédé diffère de la méthode de reconstruction faces par faces, c'est que le processus de reconnaissance va être plus contraint. En effet, pendant cette étape, le prochain surfel s' à être reconnu peut être adjacent à des faces discrètes déjà reconstruites. Dans le but de conserver une polyédrisation topologiquement correcte, il faudra donc que la face en cours de reconnaissance respecte le contour des faces déjà reconstruites. Pour cela des *contraintes de sommets* (un des points discrets de s' appartient à la discrétisation d'un sommet d'une face déjà reconstruite) et des *contraintes d'arêtes* (un des points discrets de s' appartient à la discrétisation d'une arête d'une face déjà reconstruite) vont être prises en compte durant la reconnaissance. De plus, afin d'assurer l'unicité de la reconstruction, le choix d'un surfel de départ, ainsi qu'un ordre de reconnaissance des surfels pour chaque face s'avère nécessaire.

En ce qui concerne le choix du surfel de départ, quatre stratégies ont été explorées dans [Dex06] :

1. On choisit un surfel contraint par au moins une contrainte, et parmi les candidats possibles on retient le surfel le moins contraint.
2. On choisit en priorité les surfels adjacents aux faces reconnues les plus anciennes.
3. Le surfel de départ est choisi arbitrairement parmi les surfels les moins contraints de la surface.
4. Le surfel de départ est choisi arbitrairement parmi les surfels les plus contraints de la surface.

Les résultats expérimentaux laissent penser que les méthodes 1 et 2 sont un peu mieux adaptées au problème car elles produisent en général un nombre de facettes moins important que les deux autres, même si le résultat visuel reste plutôt proche entre toutes ces méthodes. L'ordre de parcours des surfels lors de la reconnaissance est quant à lui illustré à la figure 2.23.

Pour ce qui est des contraintes introduites durant la phase de reconnaissance :

- La mise à jour de l'ensemble solution à cause d'une contrainte de sommet est simplement effectuée en faisant l'intersection entre la préimage généralisée et le plan dual de la contrainte. Si le résultat est vide la reconnaissance s'arrête, sinon un plan solution peut être choisi dans l'intersection calculée.
- Dans le cas d'une contrainte d'arête, la mise à jour de l'ensemble solution est effectuée en faisant l'intersection entre la préimage généralisée et la droite associée aux plans solutions de la droite discrète contenant l'arête.

En fin de reconnaissance, le choix du plan euclidien solution est effectué en calculant le barycentre de la préimage généralisée (ou de la préimage généralisée restreinte à un quart d'espace quand cette dernière n'est pas convexe). Le plan associé à ce point dans l'espace primal euclidien est le plan solution retenu.

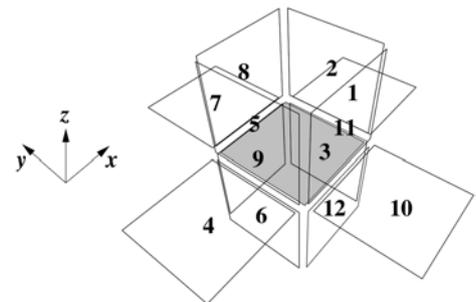


Figure 2.23 – Ordre d'ajout des surfels lors de la reconnaissance (source [Dex06]).

Enfin, la polygonalisation d'une face discrète reconnue à l'étape précédente est accomplie en deux phases. La première consiste à calculer l'intersection entre la courbe discrète définissant le contour de la face reconnue et le plan euclidien choisi. La seconde consiste à calculer une courbe polygonale coupant toutes les cellules de la courbe irrégulière issue de l'intersection, comme illustré à la figure 2.24.

Le calcul de l'intersection entre la courbe discrète et le plan euclidien solution P s'effectue incrémentalement en considérant chaque voxel l'un après l'autre. Les pointels du voxel sont triés en fonction de

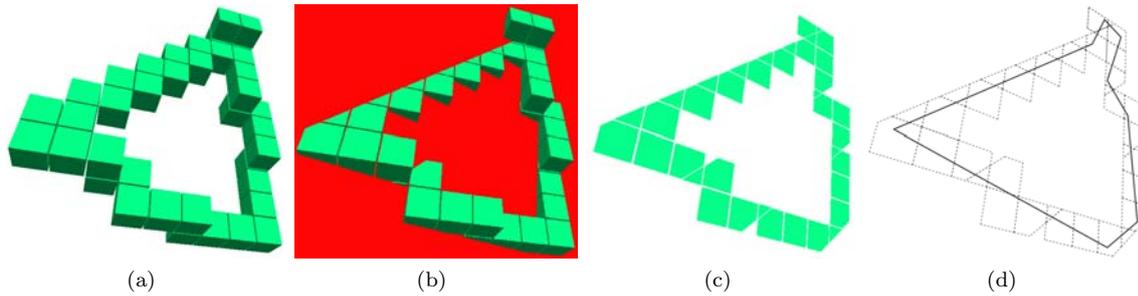


Figure 2.24 – Exemple de polygonalisation de courbe : (a) une courbe discrète, (b) un plan euclidien solution coupant tous les voxels et (c) les cellules résultantes formant la courbe irrégulière. (d) Une polygonalisation possible de cette courbe (source [Dex06]).

leur position par rapport à P (au-dessus, au-dessous ou sur) permettant ainsi de déterminer les arêtes du voxel coupées par P (figure 2.25(a)). Les points d'intersection obtenus et appartenant à une même face du voxel engendreront des arêtes, qui elles-mêmes engendreront des facettes qui seront les cellules de la courbe discrète irrégulière issue de l'intersection avec P (figure 2.25(b)). La G-carte représentant cet ensemble de cellules pourra alors être constituée, et les relations entre les différentes cellules seront établies en respectant les relations d'adjacences qui existaient entre les voxels.

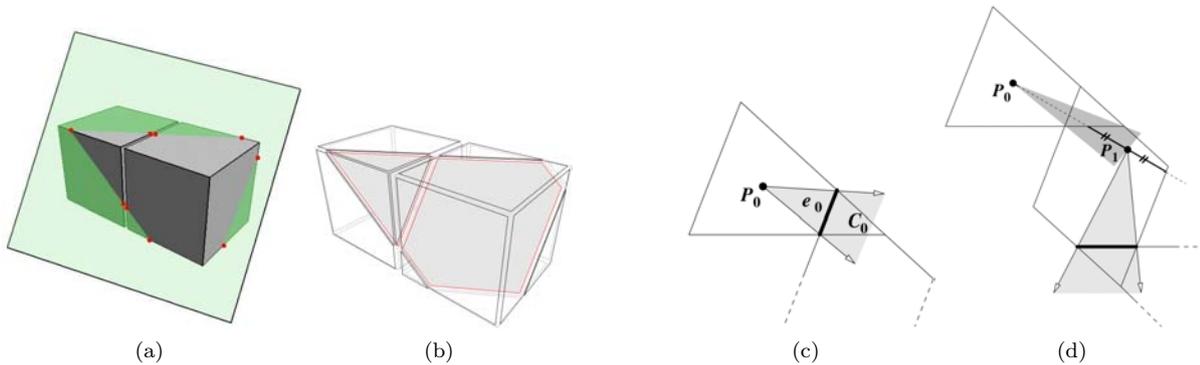


Figure 2.25 – (a) L'intersection entre le plan euclidien solution et les voxels du contour de la face forme (b) les cellules d'une courbe discrète. (c) Un cône de visibilité entre deux cellules. (d) La création d'un nouveau sommet de la courbe polygonale (source [Dex06]).

Pour terminer, l'étape de polygonalisation de la courbe obtenue est effectuée grâce à une extension aux espaces irréguliers de l'algorithme proposé par D. Coeurjolly et L. Zerarga [CZ06] dans le cadre de grilles isothétiques irrégulières. La méthode est basée sur le calcul de *cônes de visibilité*. Le principe est le suivant : on choisit un point de départ P_0 dans une première cellule de la courbe, puis on calcule un cône de visibilité C_0 vers la cellule suivante en fonction de leur arête commune e_0 (cf. figure 2.25(c)). On répète ensuite l'opération avec les cellules suivantes (relativement à l'adjacence par arête) de la courbe et on calcule l'intersection des différents cônes de visibilité. Lorsque l'intersection est vide, cela signifie que la cellule courante n'est plus accessible depuis P_0 en suivant une ligne droite. On fixe alors un point P_1 , nouveau sommet de la courbe polygonale en construction, à l'intersection de la bissectrice du dernier cône de visibilité et de la cellule précédente, comme illustré sur la figure 2.25(d). On répète ainsi le procédé jusqu'à la dernière cellule. Si le dernier cône de visibilité contient P_0 la courbe polygonale est entièrement reconstruite, sinon on fixe un point P_k , visible depuis P_0 , dans la dernière cellule et un segment est ajouté entre ce deux points pour terminer la polygonalisation.

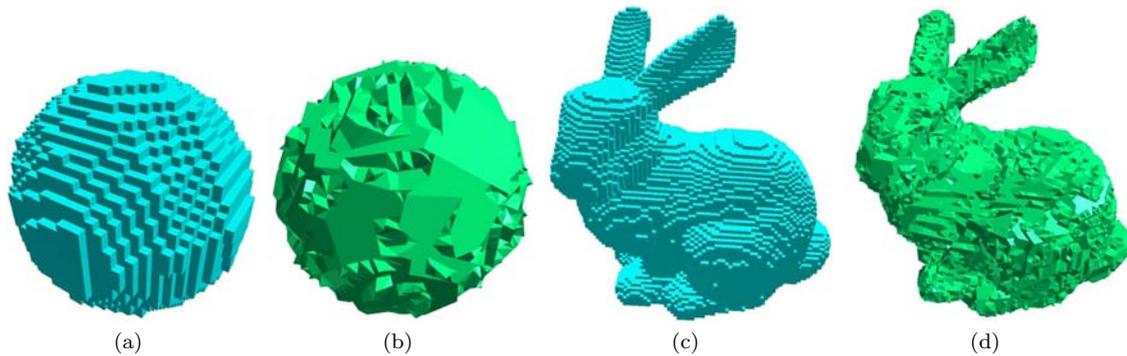


Figure 2.26 – Exemple de résultats obtenus avec le procédé de reconstruction analytique inversible sur (a) une sphere et (b) le Stanford Bunny à partir de (a) et (c) leur homologue discret (source [Dex06]).

Des résultats obtenus avec ce procédé de reconstruction analytique inversible sont présentés à la figure 2.26. Comme on peut le voir, la surface obtenue est bien un polyèdre euclidien et le maillage est topologiquement correct. Néanmoins, la surface est visuellement très accidentée du fait des contraintes de sommets et d'arêtes ajoutées au cours de la reconnaissance.

2.4 Synthèse sur la polyédrisation en géométrie discrète

Comme nous avons pu le voir à travers ce panorama de méthodes de polyédrisation en géométrie discrète, de nombreux travaux ont déjà été entrepris pour répondre à la problématique. Une classification suivant le type de polyédrisation (approximation ou exacte) et le type de facette composant la surface polygonale reconstruite (triangles ou faces polygonales basées sur des primitives discrètes) est proposée dans le tableau 2.1.

Type de faces	Approximation polyédrique	Reconstruction exacte réversible
Triangles	Approche par squelettisation Delaunay Discret	Marching Cubes
Primitives discrètes	Intersection de demi-espaces	Polyèdres discrets Reconstruction faces par faces Simplification Marching Cubes Reconstruction analytique inversible

Tableau 2.1 – Une classification possible des méthodes présentées dans cet état de l'art.

Suivant l'approche adoptée, les techniques possèdent certains avantages et certains inconvénients, et leur utilisation dépendra donc du contexte applicatif. Ainsi les techniques d'approximation polyédriques aboutissant à un maillage triangulaire de la surface, telles que l'approche Delaunay Discret ou celle par squelettisation, semblent bien adaptées à des applications de visualisation, par exemple, car le maillage obtenu est plutôt homogène, proche de la surface discrète initiale et de bonne qualité visuelle. Ceci est dû, notamment, au choix des graines, qui tient compte de la géométrie de l'objet. Néanmoins le nombre de triangles reste encore très élevé et les paramètres contrôlant l'approximation effectuée ne sont pas très intuitifs.

Au contraire les méthodes de reconstruction exacte réversible (Marching Cubes excepté) fournissent un polyèdre, qui, même s'il est souvent de moins bonne qualité visuelle (facettes non homogènes ou surface plutôt accidentée, comme dans la méthode de reconstruction analytique inversible), est généralement formé d'un plus petit nombre de facettes. Dans un contexte de représentation compacte d'objets discrets, par exemple, ces techniques revêtent un intérêt particulier et la contrainte de réversibilité nous assure

qu'on pourra toujours revenir à l'objet discret initial. De plus, comme les facettes sont reconstruites en se basant sur une décomposition du bord de l'objet discret en primitives linéaires, ces algorithmes paraissent idéaux pour une reconstruction d'objets linéaires par morceaux.

On peut néanmoins se poser des questions quant à l'utilisation de ces différentes méthodes sur des objets discrets scannés, issus du monde réel. En effet, les techniques d'acquisition n'étant pas parfaites, des irrégularités peuvent apparaître à la surface de l'objet et perturber les algorithmes décrits précédemment.

Si l'on regarde du côté des méthodes conduisant à une approximation polyédrique, on peut voir que les techniques employées semblent assez flexibles pour pouvoir traiter ce genre d'entrées, du moment que la topologie de la surface considérée répond aux critères exigés. En effet, après la phase de placement des graines (qui elle tient compte de la géométrie de l'objet) c'est principalement la topologie de la surface considérée qui est utilisée pour résoudre le problème de polyédrisation. Par contre, le nombre de facettes reste plutôt élevé, et ne correspond pas aux faces naturelles (quand cela a un sens) des objets manipulés.

En ce qui concerne les méthodes basées sur une décomposition du bord de l'objet discret en morceaux de plans discrets, les perturbations au niveau de la géométrie de la surface ont par contre une grande importance, comme le montre la figure 2.27. Ceci est principalement dû à l'utilisation d'algorithmes de reconnaissance qui s'appuient sur des définitions de primitives discrètes régulières, très sensibles à ce type de perturbations. Malgré tout, ces primitives permettent d'obtenir des résultats intéressants sur des objets réguliers, où l'on peut retrouver les faces naturelles des objets à travers les morceaux de la segmentation, comme l'illustre la figure 2.27(a) pour le cas du cube.

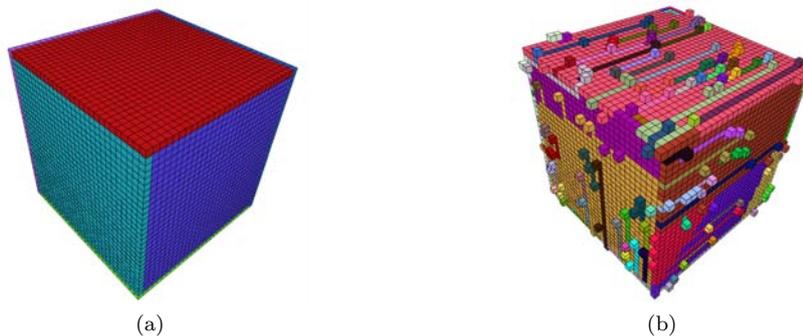


Figure 2.27 – (a) Segmentation du bord d'un cube discret en morceaux de plans discrets naïfs avec étiquetage des voxels (algorithme DPS proposé par I. Sivignon) (a) L'application du même algorithme sur un cube dont la surface contient des irrégularités.

Dans cette thèse on se propose donc d'aborder la problématique de la polyédrisation d'objets discrets bruités en apportant plus de flexibilité durant le processus de reconnaissance. L'objectif est ici d'obtenir des résultats pour lesquels on retrouve les parties linéaires naturelles des objets discrets à travers les faces de la polyédrisation. Nous avons donc, suite à nos observations des algorithmes existants, choisi de rester dans le cadre d'une approche reposant sur une segmentation en morceaux de plans discrets.

Premièrement, pour une démarche plus souple, nous avons décidé de ne pas imposer la contrainte de réversibilité, nous plaçant ainsi dans le domaine de l'approximation polyédrique. Dans ce but, nous avons introduit une nouvelle primitive discrète, le morceau flou de plan discret, permettant de reconnaître les zones plates des objets, relativement à un paramètre d'épaisseur ν contrôlant l'approximation effectuée. Nous la présentons dans le chapitre suivant.

Deuxièmement, dans l'idée de rester fidèle à la géométrie des objets discrets manipulés, nous avons opté pour un guidage des processus de décomposition de l'objet, basé sur des estimateurs de caractéristiques géométriques que nous présentons au chapitre 4.

Chapitre 3

Primitives discrètes floues

Sommaire

3.1	Introduction	47
3.2	Segments flous de droites discrètes	48
3.3	Morceaux flous de plans discrets	51
3.4	Reconnaissance de morceaux flous de plans discrets	53
3.4.1	Approche par programmation linéaire	53
3.4.2	Une approche géométrique	59
3.4.3	Adaptation de méthodes existantes	63
3.5	Résultats expérimentaux	64
3.5.1	Morceaux de plans discrets	65
3.5.2	Morceaux de sphères discrètes	67
3.5.3	Morceaux de surfaces discrètes	70
3.5.4	Synthèse et discussion	72
3.6	Conclusion	73

3.1 Introduction

Nous avons présenté dans le chapitre 2 plusieurs méthodes de polyédrisation d'objets discrets 3D et avons mis en évidence des problèmes potentiels que l'on peut rencontrer lorsque l'on travaille sur des données qui peuvent être bruitées. Lorsqu'on se restreint aux méthodes basées sur une décomposition du bord en morceaux de plans discrets, notamment pour conserver la propriété de réversibilité, on s'aperçoit que la sensibilité aux perturbations pouvant apparaître sur le bord de l'objet discret est très importante, du fait de la régularité des primitives discrètes employées.

Sans surprise, ces problèmes existaient déjà dans les images discrètes 2D pour des polygonalisations de courbes en segments de droites discrètes. Dans le cadre applicatif, les courbes discrètes étudiées contiennent souvent des irrégularités induites par les outils utilisés pour les obtenir. Sur ce type de courbes, les algorithmes de segmentation issus des travaux en géométrie discrète ne fournissaient pas de résultats satisfaisants, comme nous pouvons le constater à la figure 3.1(b); la segmentation obtenue contient beaucoup trop de segments alors que la courbe ne semble pas bruitée.

Pour résoudre ce type de problème, une nouvelle approche a été envisagée : découper la courbe en droites épaisses qui permettent d'obtenir de plus long segments, comme illustré à la figure 3.2. Pour atteindre cet objectif, un nouvel objet discret a été introduit, le *segment flou arithmétique*, reposant sur la définition d'une droite discrète arithmétique d'épaisseur variable.

Dans le même esprit, nous avons choisi d'apporter plus de flexibilité dans l'utilisation des morceaux de plans discrets. Pour cela, une nouvelle primitive discrète, le *morceau flou de plan discret*, a été introduite.

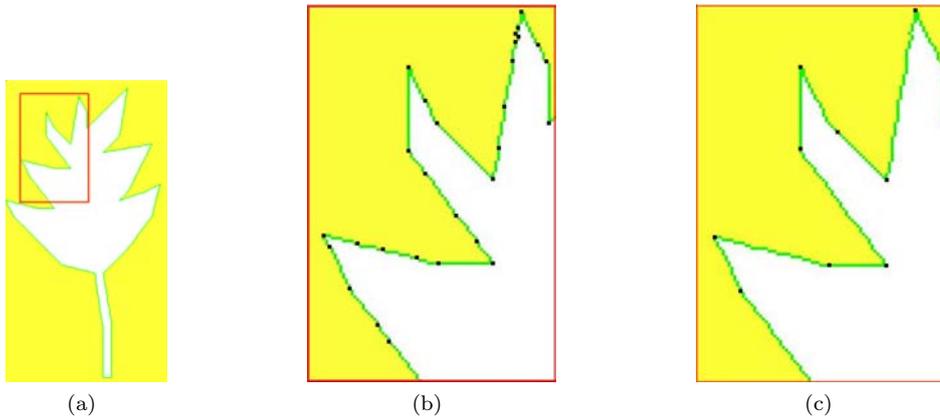


Figure 3.1 – (a) Un exemple de courbe discrète obtenue dans un contour de l’image d’une feuille. (b) Un agrandissement sur une partie du contour et segmentation obtenue avec un algorithme de reconnaissance de segments de droites naïves. (c) La segmentation souhaitée sur la même partie de la courbe.

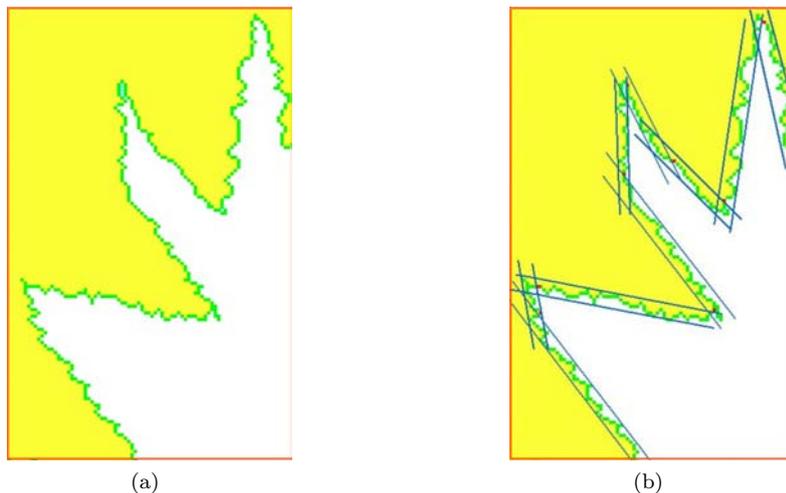


Figure 3.2 – (a) Un exemple de courbe discrète fortement bruitée et (b) une segmentation souhaitée.

La suite de ce chapitre est consacrée à sa présentation et à son étude.

Pour mieux appréhender le concept de primitive discrète floue, nous allons dans un premier temps le présenter en 2D, à travers la notion de segment flou de droite discrète. Puis, en nous basant sur les études effectuées en 2D, une nouvelle définition, adaptée aux morceaux de plans, sera proposée. Plusieurs algorithmes de reconnaissances seront finalement présentés.

3.2 Segments flous de droites discrètes

La notion de primitive discrète floue, qui correspond à l’apport de flexibilité pour l’étude d’objets discrets possiblement bruités, a été introduite en 2003 par I. Debled-Rennesson et ses coauteurs à travers l’étude des segments de droites discrètes bruitées [DRRRD03].

Les définitions et algorithmes présentés ici, qu’on peut retrouver dans [DR07], correspondent à une évolution de ceux proposés initialement dans [DRRRD03] et améliorés dans [DRFRD05, DRFRD06].

L’idée est la suivante : en reposant sur l’approche arithmétique des droites discrètes, les segments

flous les généralisent, en admettant que certains points soient « manquants ». Nous donnons ci-dessous les définitions permettant de formaliser cette idée. Sans perte de généralité, ces dernières sont données pour une étude dans le premier octant du plan, c'est-à-dire pour $0 < a < b, b \neq 0$.

Définition 3.1 Soit \mathcal{S}_b une suite de points de \mathbb{Z}^2 . On dit que la droite discrète $\mathcal{D}(a, b, \mu, \omega)$ est une *droite englobante de \mathcal{S}_b* si tous les points de \mathcal{S}_b appartiennent à \mathcal{D} .

Définition 3.2 On appelle *épaisseur verticale de la droite* $\mathcal{D}(a, b, \mu, \omega)$, la distance verticale (différence des ordonnées) entre les droites d'appui de \mathcal{D} .

Proposition 3.3 L'épaisseur verticale de $\mathcal{D}(a, b, \mu, \omega)$ est égale à $\frac{\omega-1}{\max(|a|, |b|)}$.

Définition 3.4 La *distance verticale d'un ensemble convexe* C est la longueur maximale de l'intersection de l'enveloppe convexe de C avec une droite verticale.

Définition 3.5 Une *droite support* d'un ensemble convexe C est une droite l telle que C est contenu entièrement dans un demi-plan déterminé par l et telle que l'intersection entre C et l n'est pas vide.

Proposition 3.6 La distance verticale d'un ensemble convexe C est la distance verticale minimale entre toutes les paires de droites support de C parallèles.

Les différentes notions définies jusqu'à présent sont illustrées à la figure 3.3.

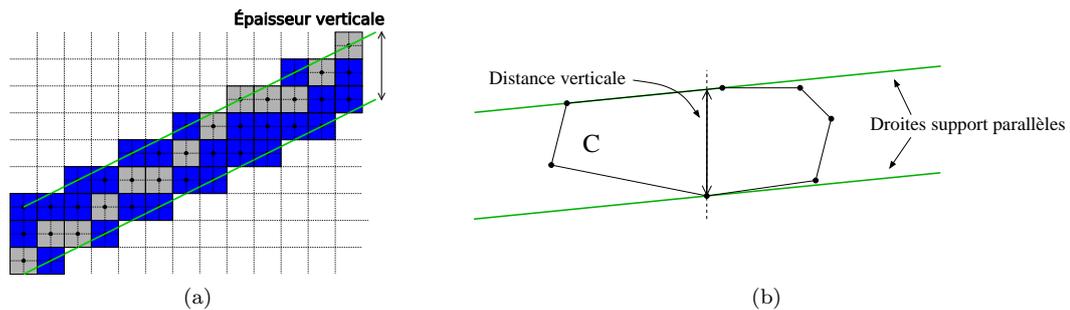


Figure 3.3 – (a) $\mathcal{D}(1, 2, -4, 6)$, une droite englobante de la suite de points grisés et ses droites d'appui en vert. (b) La distance verticale d'une enveloppe convexe avec deux droites support parallèles.

Définition 3.7 Une droite englobante d'une suite de points entiers \mathcal{S}_b est dite *optimale* si son épaisseur verticale est égale à la distance verticale de l'enveloppe convexe de \mathcal{S}_b .

Ceci nous amène finalement à la définition d'un segment flou, avec un exemple fourni à la figure 3.4 :

Définition 3.8 Une suite de points entiers \mathcal{S}_b est un *segment flou d'épaisseur ν* si et seulement si sa droite englobante optimale a une épaisseur verticale inférieure ou égale à ν .

Reconnaissance de segments flous par une approche géométrique

Les définitions précédentes mettent en évidence une propriété intéressante : le problème de la reconnaissance d'un segment flou \mathcal{S}_b est équivalent à celui du calcul de la distance verticale de l'enveloppe convexe de \mathcal{S}_b (notée $\text{conv}(\mathcal{S}_b)$).

Or, si l'on note L_i et U_i respectivement les plus bas et les plus hauts points d'intersection entre $\text{conv}(C)$ et la droite verticale d'équation $x = i$, et $\mathcal{V}(i)$ la distance entre L_i et U_i , alors $\mathcal{V}(\cdot)$ a la propriété d'être une fonction concave. Il en résulte que chaque maximum local de $\mathcal{V}(\cdot)$ est aussi un maximum global. De plus, la proposition suivante est montrée dans [DRFRD05] :

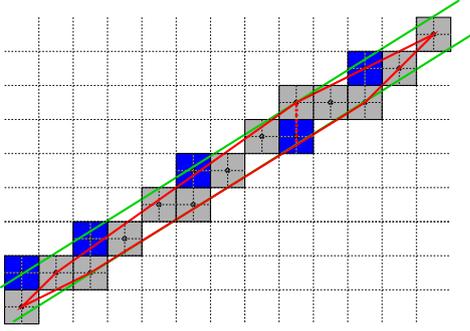


Figure 3.4 – L'ensemble \mathcal{S}_b , formé des points gris, est un segment flou d'épaisseur 2. $\mathcal{D}(5, 8, -8, 11)$, formé des points gris et bleus, est une droite englobante optimale de \mathcal{S}_b (d'épaisseur verticale $\frac{10}{8} = 1.25$). On aperçoit en rouge l'enveloppe convexe du segment flou et en vert les droites d'appui de \mathcal{D} qui sont aussi les droites support de l'enveloppe convexe de \mathcal{S}_b entre lesquelles la distance verticale est minimale.

Proposition 3.9 Pour chaque ensemble convexe fini C , la fonction $\mathcal{V}(\cdot)$ a une valeur maximale à la position i avec L_i ou U_i pouvant être choisi comme un sommet de $\text{conv}(C)$.

Ainsi, lors de la recherche de la distance verticale d'un convexe C :

- nous pouvons ne considérer que les points du bord de $\text{conv}(C)$;
- l'extremum peut être localisé à partir de certaines positions des abscisses des sommets de $\text{conv}(C)$; en fait trois cas sont possibles : (arête, sommet), (sommet, arête) et (sommet, sommet) où le premier élément représente la partie inférieure et le second la partie supérieure du bord.

L'algorithme de reconnaissance de segments flous est directement basé sur ces observations. La procédure incrémentale de reconnaissance est décrite ci-dessous en présentant les modifications engendrées par l'ajout d'un nouveau point M .

Soit $\mathcal{S}_b = \{(x_i, y_i), 0 \leq i < n\}$ un segment flou dans le premier octant ayant $\mathcal{D}(a, b, \mu, \omega)$ comme droite englobante optimale. Nous supposons que \mathcal{S}_b possède deux points d'appui supérieurs, U_F et U_L , et un point d'appui inférieur, L_L . U_F , U_L et L_L sont des sommets de l'enveloppe convexe de \mathcal{S}_b et la distance verticale de \mathcal{D} et de l'enveloppe convexe est atteinte en L_L . La situation est décrite à la figure 3.5.

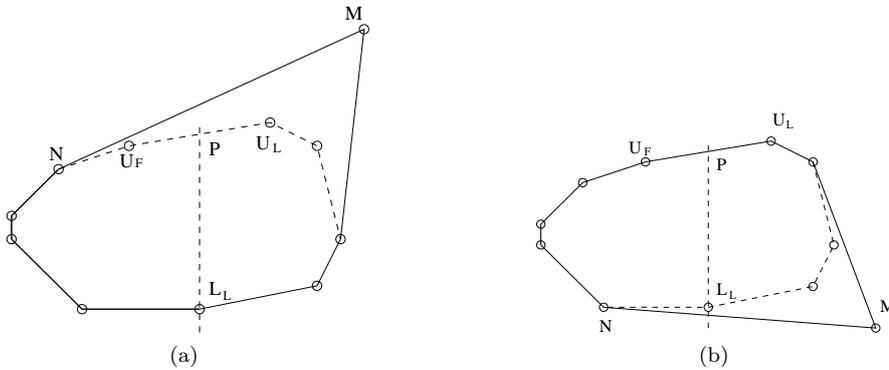


Figure 3.5 – L'ajout d'un nouveau point (a) au-dessus ou (b) au-dessous de l'enveloppe convexe.

Supposons que nous ajoutons un point M à \mathcal{S}_b , $\mathcal{S}'_b = \mathcal{S}_b \cup \{M\}$. Tout d'abord la mise à jour de l'enveloppe convexe C est faite grâce à l'algorithme de Melkman⁸ [Mel87]. Ensuite, nous calculons le reste r de M par rapport à \mathcal{D} . Trois cas sont alors possibles : M est au-dessus, au-dessous ou bien appartient à \mathcal{D} .

Si $M \in \mathcal{D}$, c'est-à-dire si $r \in [0, \omega - 1]$, alors après la mise à jour de l'enveloppe convexe, la distance verticale reste la même, et donc \mathcal{S}'_b est un segment flou de même épaisseur et de même droite englobante \mathcal{D} que \mathcal{S}_b .

8. L'algorithme de Melkman calcule l'enveloppe convexe de n points formant une ligne polygonale de manière incrémentale en $\mathcal{O}(n)$.

Dans les deux autres situations, c'est-à-dire quand $r \notin [0, \omega - 1]$, cela signifie que le point a été ajouté au-dessus ou au-dessous de \mathcal{D} . Nous devons alors recalculer la distance verticale maximale.

Supposons que M soit ajouté au-dessus de \mathcal{D} . L'enveloppe convexe est modifiée en conséquence en utilisant l'algorithme de Melkman et nous appelons N le point situé avant M dans la partie supérieure de l'enveloppe convexe résultante, comme indiqué sur la figure 3.5(a). N est nécessairement situé sur ou avant le point U_F . Par conséquent, la projection verticale de L_L est dans $[NM]$. Alors, la distance verticale de ce nouvel ensemble convexe croît strictement. N ne peut pas être localisé à une position maximale puisque $\mathcal{V}(N) \leq \mathcal{V}(L_L)$. De plus, M ne peut pas se projeter verticalement strictement à l'intérieur d'une arête de la partie inférieure de l'enveloppe convexe de $\text{conv}(\mathcal{S}'_b)$. Par conséquent, la nouvelle position du maximum est nécessairement obtenue sur un point à droite de L_L dans la partie inférieure de $\text{conv}(\mathcal{S}'_b)$. Et grâce à la propriété 3.9, la recherche des points candidats s'arrêtera au premier maximum trouvé, que nous nommerons T .

Supposons maintenant que M soit ajouté au-dessous de \mathcal{D} . Après la mise à jour de l'enveloppe convexe, N est comme précédemment situé avant M dans la partie inférieure de l'enveloppe convexe résultante (voir figure 3.5(b)). N se trouve nécessairement à gauche de L_L . Par conséquent, N ne peut pas être à la position du maximum puisque $\mathcal{V}(N) \leq \mathcal{V}(L_L)$. Nous ne savons pas exactement où N se situe par rapport à $[U_F U_L]$. Cependant, il est aisé de constater qu'aucune position à gauche de U_L ne peut être la position maximale. Comme précédemment, ni M ni N ne sont à une position maximale. Ainsi la position du nouveau maximum se situe à droite de U_L , sur la partie supérieure de l'enveloppe convexe et il suffit de tester les points candidats de la suite de points et de s'arrêter sur le premier maximum obtenu, que nous nommerons T .

Dans ces deux situations, \mathcal{S}'_b est un segment flou avec pour droite englobante optimale \mathcal{D}' pour laquelle les points M , N , et T sont des points d'appui. De plus, la distance verticale de $\text{conv}(\mathcal{S}'_b)$ est égale à l'épaisseur verticale de \mathcal{D}' et peut être calculée au point T .

Cette étude de l'évolution d'un segment flou a permis de mettre au point un algorithme de reconnaissance incrémental linéaire de segments flous d'épaisseur ν . Pour cela, les points de l'ensemble à reconnaître sont ajoutés un à un par ordre croissant de leur coordonnée x et on applique la procédure incrémentale décrite ci-dessus. Si au cours de l'algorithme l'épaisseur verticale de la droite englobante optimale devient supérieure à ν , alors l'ensemble de points n'est pas un segment flou d'épaisseur ν , sinon il l'est.

Un exemple du déroulement de l'algorithme est illustré à la figure 3.6. L'ensemble de points connectés est représenté en haut de la figure, et les différentes droites englobantes optimales obtenues au cours de l'algorithme de reconnaissance incrémental sont données au-dessous.

Comme on peut le constater, l'approche géométrique, qui exploite les propriétés de l'enveloppe convexe des points du segment flou, permet d'obtenir un algorithme simple à comprendre, visuel et efficace (complexité en $\mathcal{O}(n)$ où n représente le nombre de points du segment flou). Nous allons donc nous inspirer des techniques utilisées pour les segments flous, afin de proposer une extension pour les plans discrets en dimension 3. C'est ce que nous présentons dans les sections suivantes.

3.3 Morceaux flous de plans discrets

Tout comme pour les segments flous, nous allons proposer dans cette partie une généralisation de la notion de plan discret, reposant sur l'approche arithmétique et apportant plus de flexibilité en admettant que certains points soient manquants. Nous nous inspirons dans un premier temps des définitions proposées pour les segments flous afin de poser la définition d'une nouvelle primitive discrète, le *morceau flou de plan discret*. Néanmoins, pour plus de généralité, nous n'allons pas considérer la distance verticale, mais plutôt proposer une définition relativement à une norme. Ainsi, étant donnée une norme N de \mathbb{R}^3 , nous pouvons généraliser la notion de droite englobante de la façon suivante :

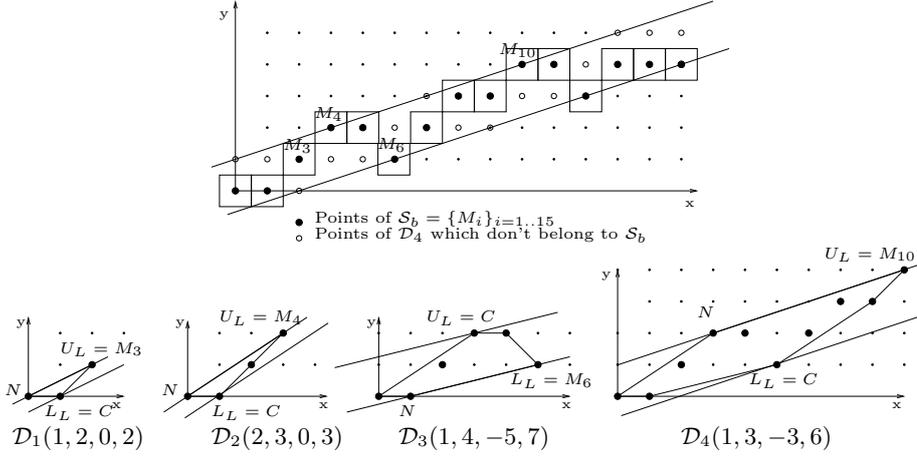


Figure 3.6 – Un exemple de reconnaissance incrémentale d'un segment flou d'épaisseur 2.

Définition 3.10 Soit \mathcal{E}_b un ensemble de points de \mathbb{Z}^3 . On dit que le plan discret $\mathcal{P}(a, b, c, \mu, \omega)$ est un **plan englobant de \mathcal{E}_b** si tous les points de \mathcal{E}_b appartiennent à \mathcal{P} . On appelle *hauteur du plan $\mathcal{P}(a, b, c, \mu, \omega)$* , la valeur $\frac{\omega-1}{N(a,b,c)}$.

La hauteur d'un plan discret peut être interprétée de la façon suivante :

1. si $N = \|\cdot\|_2$, alors la hauteur $\frac{\omega-1}{N(a,b,c)}$ représente la distance euclidienne entre les deux plans d'appui du plan $\mathcal{P}(a, b, c, \mu, \omega)$. En effet, soient $P_1 : ax + by + cz = \mu$ et $P_2 : ax + by + cz = \mu + \omega - 1$ les deux plans d'appui de \mathcal{P} . Alors comme P_1 and P_2 sont parallèles, la distance entre P_1 et P_2 est égale à $\frac{|\mu + \omega - 1 - \mu|}{\sqrt{a^2 + b^2 + c^2}}$, i.e. $\frac{\omega-1}{\|(a,b,c)\|_2}$ puisque $\omega > 0$.
2. si $N = \|\cdot\|_\infty$, alors la hauteur $\frac{\omega-1}{N(a,b,c)}$ représente la distance selon la direction principale du vecteur (a, b, c) . En effet, et sans perte de généralité, nous pouvons supposer que $\max(|a|, |b|, |c|) = |c|$, ce qui signifie que la direction principale est colinéaire à l'axe Oz . Soient $M_1(x_1, y_1, z_1) \in P_1$ et $M_2(x_2, y_2, z_2) \in P_2$ tels que $x_1 = x_2$ et $y_1 = y_2$. La distance entre P_1 et P_2 est égale à $|z_1 - z_2| = \frac{|c(z_1 - z_2)|}{|c|} = \frac{|a(x_1 - y_2) + b(y_1 - y_2) + c(z_1 - z_2)|}{|c|} = \frac{|ax_1 + by_1 + cz_1 - (ax_2 + by_2 + cz_2)|}{|c|} = \frac{|\mu - (\mu + \omega - 1)|}{|c|}$ car $M_1 \in P_1$ et $M_2 \in P_2$, i.e. $\frac{\omega-1}{\|(a,b,c)\|_\infty}$ puisque $\omega > 0$.

Nous proposerons dans la suite de ce chapitre différents algorithmes de reconnaissance relativement aux deux normes considérées ci-dessus.

Définition 3.11 Soit \mathcal{E}_b un ensemble de points de \mathbb{Z}^3 . Un plan englobant de \mathcal{E}_b est dit **optimal** si sa hauteur est minimale.

Ceci nous amène à la définition d'un *morceau flou de plan discret*, dont un exemple est illustré à la figure 3.7 :

Définition 3.12 Un ensemble \mathcal{E}_b de points de \mathbb{Z}^3 est un **morceau flou de plan discret d'épaisseur ν** (noté MFDP $_\nu$) si et seulement si la hauteur d'un plan englobant optimal de \mathcal{E}_b est inférieure ou égale à ν .

Ainsi, intuitivement, un morceau flou de plan discret d'épaisseur ν , est un ensemble de points discrets qui peut être englobé par une paire de plans euclidiens parallèles dont la distance qui les sépare, relativement à la norme considérée, est inférieure à ν . Suivant l'interprétation de cette observation, plusieurs algorithmes de reconnaissance de MFDP peuvent être mis au point, et nous allons en présenter quelques uns dans la section suivante.

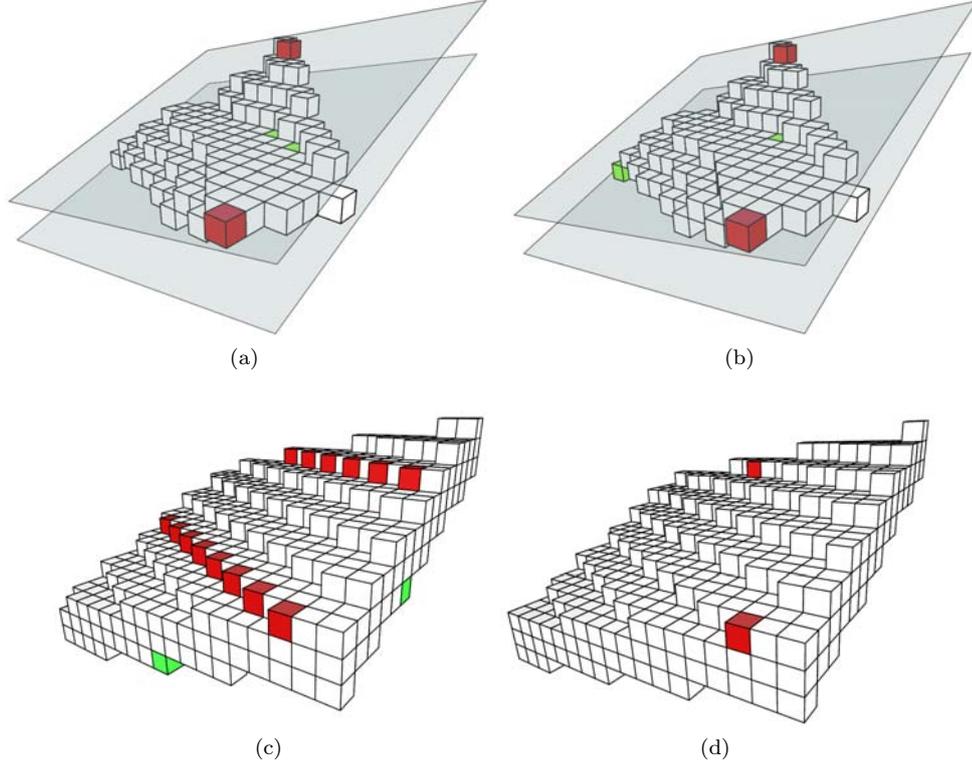


Figure 3.7 – (a) et (b) Un morceau flou de plan discret d'épaisseur 3 et un plan englobant optimal pour (c) la norme euclidienne : $\mathcal{P}_2(4, 8, 19, -80, 49)$ et la hauteur de $\mathcal{P}_2 = 2.28$ (d) la norme infinie : $\mathcal{P}_\infty(31, 65, 157, -680, 397)$ et la hauteur de $\mathcal{P}_\infty = 2.52$. Les plans d'appui et les points d'appui correspondants de \mathcal{P}_2 et \mathcal{P}_∞ sont respectivement représentés sur (a), (c) et (b), (d).

3.4 Reconnaissance de morceaux flous de plans discrets

3.4.1 Approche par programmation linéaire

Nous proposons ici un algorithme de reconnaissance de MFPD qui permet de résoudre le problème en considérant la norme infinie $\|\cdot\|_\infty$. Nous rappelons dans un premier temps la formulation générale d'un problème de programmation linéaire, ainsi qu'une méthode de résolution : l'algorithme du simplexe. Nous verrons ensuite comment modéliser le problème de reconnaissance de MFPD en ces termes pour pouvoir lui appliquer l'algorithme du simplexe et nous proposerons une interprétation géométrique des opérations du problème dual.

3.4.1.1 Programmation linéaire et algorithme du simplexe

Nous essayons de trouver un point minimum $x^* \in \mathbb{R}^d$ d'une fonction $f(x) : \mathbb{R}^d \rightarrow \mathbb{R}$ où $x = (x_1, \dots, x_d)$. De plus, x^* doit satisfaire un ensemble de n contraintes $G = (g_i(x) \leq b_i)_{1 \leq i \leq n}$. La Programmation Linéaire (PL) désigne l'étude et l'optimisation de tels problèmes dans le cas où la *fonction objectif* f et l'ensemble des contraintes G du problème sont tous deux linéaires.

Soit $A(n \times d)$ une matrice de n lignes et d colonnes. Soit $c(d)$, $b(n)$ et $x(d)$ trois vecteurs colonnes de taille d et n . Un problème de PL peut alors formellement être énoncé de la façon suivante :

$$\min c^t \cdot x \quad \text{tel que} \quad A \cdot x \leq b \text{ et } x \geq 0$$

On appelle alors *forme standard* la réécriture équivalente suivante :

$$\min c'^t \cdot x' \quad \text{tel que} \quad A' \cdot x' \leq b \text{ et } x' \geq 0$$

où $A' = [A | Identite(n \times n)]$, $c' = [c | Zero(n)]$. Les n variables ajoutées dans la forme standard sont appelées les *variables d'écart*.

L'algorithme du simplexe Cette méthode, développée par George Dantzig en 1947, fournit un outil de calcul puissant (cf. [Chv83]) pour la résolution de tels problèmes. Il opère sur l'énoncé en forme standard du problème. Nous avons $n + d$ variables et n égalités dans le système $A.x = b$, et nous pouvons extraire de A une matrice non singulière B de rang n relativement à ce système d'équations. La *base* correspond aux indices des colonnes extraites de A pour créer B . Dans la méthode du simplexe, les *variables hors base*, dénotées $x_N = (x_i)_{\substack{1 \leq i \leq n+d \\ i \notin base}}$, sont fixées à zéro. Les *variables de bases* $x_B = (x_i)_{i \in base}$ sont donc égales à $B^{-1}b$. Une solution x associée à une base B est appelée *faisable* quand elle vérifie $x_B \geq 0$.

L'algorithme du simplexe démarre à partir d'une solution faisable. À chaque itération, une nouvelle base est calculée de sorte que la solution associée à cette nouvelle base soit faisable et telle que la fonction objectif ait diminué (ou soit inchangée).

Pour construire cette nouvelle base, une variable hors base est choisie pour entrer dans la base et vice versa. Afin de choisir quelle variable entre dans la base, notons N les colonnes de A dont les indices ne sont pas dans la base. À partir de $A.x = b$, nous avons : $[B|N].x_B, x_N = b$. Comme B est une matrice non singulière, nous obtenons : $x_B = B^{-1}.(b - N.x_N)$. La fonction objectif peut donc être réécrite de la façon suivante : $f(x) = c^t.x = c_B^t.x_B + c_N^t.x_N = (c_N^t - c_B^t B^{-1}N)x_N + c_B^t B^{-1}b$. Et cette transformation n'est pas dépendante des variables x_B . Ainsi, puisque les variables sont positives, s'il n'existe aucune valeur négative dans le *vecteur coût réduit* $rc^t = c_N^t - c_B^t B^{-1}N$, nous avons trouvé le minimum x^* . S'il existe une valeur négative, alors nous pouvons diminuer la valeur courante de la fonction objectif en augmentant la variable correspondante x_l de x_N . Comme x_l n'est plus à zéro, lors de la prochaine itération, elle sera reclassifiée comme variable de base. En augmentant x_l , les valeurs des variables de bases sont modifiées. Si elles augmentent toutes, alors le problème n'est pas *borné*, ce qui signifie que la valeur minimale de la fonction objectif est $-\infty$. Dans l'autre cas, où certaines variables diminuent quand x_l augmente, la première variable de base x_k atteignant zéro va stopper l'accroissement de x_l , et par la même occasion x_k va sortir de la base.

Pour déterminer l'index k en question, considérons les égalités $x_B = B^{-1}.(b - N.x_N)$. Seule x_l est actuellement non nulle parmi x_N , donc nous avons $x_B = B^{-1}.b - B^{-1}A_l x_l$.

Notons \bar{b} et P respectivement $B^{-1}b$ et $B^{-1}A_l$. Les valeurs dans \bar{b} sont positives, donc seuls les indices associés à une valeur positive dans P nous intéressent. La condition précédente $\bar{b} - P.x_l \geq 0$ implique donc que pour tout i dans la base tel que $P_i > 0$, nous avons : $x_l \leq \bar{b}_i/P_i$. Et il en découle que $k = \text{index de } \min_{i, P_i > 0} \{\bar{b}_i/P_i\}$.

Fonction MIN-SIMPLEX(A, b, c, base)

Entrées : A, b : matrice et vecteur du problème, c : vecteur coût, base : base initiale

```

1 répéter
2   extraire B, c_B de A relativement à base
3    $\bar{b} = B^{-1}b$ 
4    $rc' = c^t - (c_B^t B^{-1})A$  // reformulation équivalente de rc
5   si ( $rc' \geq 0$ ) alors // optimum trouvé ( $\leq 0$  pour MAX-SIMPLEX)
6     retourner  $\bar{b}$ 
7   choisir l tel que  $rc'_l < 0$  //  $x_l$  va entrer dans la base ( $> 0$  pour MAX-SIMPLEX)
8    $P = B^{-1}A_l$ 
9   si  $P \leq 0$  alors
10    retourner non borné
11   $k = \min_{i, P_i > 0} \{\bar{b}_i/P_i\}$ 
12  base  $\leftarrow$  base  $\setminus \{k\} \cup \{l\}$ 

```

Théorème de dualité À chaque problème *primal* de PL, on peut associer un problème « compagnon » appelé problème *dual*. Un théorème principal de PL montre que le problème primal est non faisable si et seulement si son problème dual associé est non borné, et vice versa. De plus, un problème admet un optimum si et seulement si l'autre problème admet aussi un optimum, et les valeurs optimales de ces deux problèmes sont dans ce cas égales. De plus, si c_B et B sont des matrices associées à un optimum du problème dual, alors l'optimum du problème primal est égal à $c_B^t B^{-1}$. La correspondance entre les énoncés d'un problème primal et son dual est représentée sur le diagramme 3.1.

	Primal	(i) $\min c^t \cdot x$	\longleftrightarrow	Dual	$\max b^t \cdot \lambda$
tel que :		(ii) $A \cdot x \geq b$	\longleftrightarrow	tel que :	$\lambda \geq 0$
		(iii) $A \cdot x = b$	\longleftrightarrow		$\lambda \in \mathbb{R}$
		(iv) $x \geq 0$	\longleftrightarrow		$A^t \cdot \lambda \leq c$
		(v) $x \in \mathbb{R}$	\longleftrightarrow		$A^t \cdot \lambda = c$

Diagramme 3.1 – Correspondances entre les énoncés d'un problème primal et son dual.

Nous allons à présent montrer comment modéliser le problème de reconnaissance de MFPD en ces termes pour pouvoir appliquer l'algorithme du simplexe.

3.4.1.2 Modélisation du problème de reconnaissance

Selon la définition d'un MFPD, le problème de reconnaissance d'une telle primitive revient à rechercher les caractéristiques a, b, c, μ, ω d'un plan discret englobant optimal \mathcal{P} pour un ensemble de n points, telles que la distance entre les deux plans d'appui de \mathcal{P} , relativement à leur axe principal, soit minimale. Sans perdre en généralité, nous pouvons nous placer dans le cas où $\|a, b, c\|_\infty = c$, et nous considérons alors la distance verticale entre les plans d'appui de \mathcal{P} . Nous pouvons alors, en posant $\alpha = -\frac{a}{c}$, $\beta = -\frac{b}{c}$, $h = \frac{\mu}{c}$ et $e = \frac{\omega-1}{c}$, reformuler le problème de la façon suivante : pour un ensemble \mathcal{S}_b de n points entiers $(x_i, y_i, z_i)_{1 \leq i \leq n}$, nous recherchons deux plans euclidiens $P : z(x, y) = \alpha \cdot x + \beta \cdot y + h$ et $P' : z'(x, y) = \alpha \cdot x + \beta \cdot y + h + e$ tels que tous les points de \mathcal{S}_b se situent entre les deux plans P et P' , et tels que e est minimal. On obtient par conséquent un couple d'inéquations pour chacun des points de \mathcal{S}_b : $\alpha \cdot x_i + \beta \cdot y_i + h \leq z_i$ et $\alpha \cdot x_i + \beta \cdot y_i + h + e \geq z_i$. Et la formulation du problème, en terme de PL, sous forme standard devient :

Primal	Forme standard duale
$\min e$ $\begin{cases} -\alpha \cdot x_i - \beta \cdot y_i - h \geq -z_i \\ \alpha \cdot x_i + \beta \cdot y_i + h + e \geq z_i \\ i = 1, \dots, n \end{cases}$ $ \alpha \leq 1, \beta \leq 1$ $\alpha, \beta, h \in \mathbb{R}, e \geq 0$	$\max [-z_1 \dots -z_n \mid z_1 \dots z_n \mid -1 \ -1 \ -1 \ -1 \ 0] \cdot \lambda$ $\begin{bmatrix} -x_1 & \dots & -x_n & x_1 & \dots & x_n & -1 & 1 & 0 & 0 & 0 \\ -y_1 & \dots & -y_n & y_1 & \dots & y_n & 0 & 0 & -1 & 1 & 0 \\ -1 & \dots & -1 & 1 & \dots & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \dots & 0 & 1 & \dots & 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \dots \\ \lambda_{2n+5} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$ $\lambda \geq 0$

Nous avons regroupé les différents types d'inégalités au sein de la matrice. En fait, travailler sur la forme standard du problème primal nous aurait forcé à manipuler une matrice creuse de taille $(2n + 4) \times (2n + 8)$. La forme duale nous permet d'éviter ce désagrément en ne manipulant qu'une matrice de taille $4 \times (2n + 5)$. Le passage d'une formulation à l'autre s'effectue grâce aux correspondances (i), (ii) et (v) du diagramme 3.1.

Nous pouvons facilement vérifier que la base $\{\lambda_1, \lambda_{2n+1}, \lambda_{2n+3}, \lambda_{2n+5}\}$ où $B^{-1}b = [0 \ 0 \ 0 \ 1]^t \geq 0$ est toujours associée à une solution faisable du problème dual. Nous avons alors tout ce qu'il faut pour

appliquer l'algorithme du simplexe à ce problème.

Nous venons de voir comment formuler le problème de reconnaissance de MFPD en termes de PL, pour le résoudre en appliquant l'algorithme du simplexe. Mais, une fois le problème modélisé, d'autres méthodes de PL peuvent être appliquées pour le résoudre. Notamment les méthodes qualifiées de « points intérieurs » (cf. [Kar84, Wri97]) qui peuvent être, en théorie, plus performantes que le simplexe. Il serait alors intéressant de regarder en pratique si cela se vérifie. Néanmoins, nous nous sommes intéressés de plus près à la méthode du simplexe car les opérations effectuées au cours de l'algorithme sur le problème dual possèdent une interprétation géométrique intéressante que nous présentons dans la section suivante.

3.4.1.3 Interprétation géométrique du problème dual

La base du problème dual est associée à quatre inégalités du problème primal. Ainsi, quand λ_i est dans la base, la i^e inégalité du problème primal est en fait une égalité. Par exemple, quand $\lambda_i, 1 \leq i \leq n$ est dans la base, la i^e inégalité implique que $\alpha.x_i + \beta.y_i + h = z_i$, ce qui signifie que le point p_i appartient au plan inférieur P . Quand $n < i \leq 2n$, le point p_{i-n} appartient au plan supérieur P' . De la même manière, les variables $\lambda_{2n+1}, \dots, \lambda_{2n+5}$ sont associées aux cas : $\alpha = 1, \alpha = -1, \beta = 1, \beta = -1$ ou $e = 0$.

Le vecteur $c_B^t B^{-1}$ dans le dual transforme la base courante en des variables du problème primal. Ceci provient de l'interprétation précédente. Soit K la matrice correspondant aux égalités retenues au niveau du problème primal. Le système vérifie $K \cdot [\alpha \ \beta \ h \ e]^t = b_B^{Primal}$. C'est-à-dire, nous avons $[\alpha \ \beta \ h \ e] = (K^{-1} \cdot b_B^{Primal})^t = (b_B^{Primal})^t \cdot (K^t)^{-1} = c_B^{Dual} \cdot B_{Dual}^{-1}$, et nous pouvons ainsi retrouver les caractéristiques de notre MFPD.

La condition d'optimalité du vecteur coût réduit : L'algorithme du simplexe maximise une fonction dans le problème dual. Ainsi, il s'arrête quand il trouve un vecteur coût réduit rc composé uniquement de valeurs négatives (ligne 5 de l'algorithme du simplexe). Nous pouvons facilement vérifier que : $rc^t = c^t - (c_B^t B^{-1}) \cdot A = [(-z_i + [\alpha.x_i + \beta.y_i + h])_{1 \leq i \leq n} \mid (z_i - [\alpha.x_i + \beta.y_i + h + e])_{1 \leq i \leq n} \mid -1 + \alpha \mid -1 - \alpha \mid -1 + \beta \mid -1 - \beta \mid -e]$. Comme toutes ces valeurs sont négatives, ceci implique que les inégalités dans le primal sont toutes vérifiées. Ainsi, quand l'algorithme dans le dual s'arrête, cela signifie que les deux plans parallèles recherchés enserrant bien tous les points.

La fonction objectif du dual peut paraître obscure, néanmoins le théorème de dualité nous affirme que la fonction objectif du dual représente la même chose que la fonction objectif du primal. C'est-à-dire $f(\lambda) = c_B^{Dual} (B^{-1} b_{Dual}) = (c_B^{Dual} B^{-1}) b_{Dual} = [\alpha \ \beta \ h \ e] \times c_{Primal} = e$, soit la hauteur verticale entre les deux plans parallèles recherchés.

Le cœur de l'algorithme : Chaque itération est associée à une base représentant une solution faisable. Nous ne présentons que les cas généraux, qui sont les plus importants, c'est-à-dire avec toutes les variables λ_i telles que $1 \leq i \leq 2n$. La configuration $1 \leq i, j, k, l \leq n$ pour les indices des variables de base n'est pas possible et correspond à un cas où la matrice B serait singulière.

- **Configuration 1 :** $1 \leq i, j, k \leq n < l \leq 2n$. Dans ce cas les trois points $p_i(x_i, y_i)$, $p_j(x_j, y_j)$ et $p_k(x_k, y_k)$ définissent le plan inférieur P et le plan parallèle P' passe par p_{l-n} . La matrice B est égale à $[-x_i \ -y_i \ -1 \ 0 \mid -x_j \ -y_j \ -1 \ 0 \mid -x_k \ -y_k \ -1 \ 0 \mid x_{l-n} \ y_{l-n} \ 1 \ 1]$. Sans perte de généralité, on peut supposer que le point p_{l-n} correspond à l'origine, ce qui permet de simplifier l'écriture de B en $[-x_i \ -y_i \ -1 \ 0 \mid -x_j \ -y_j \ -1 \ 0 \mid -x_k \ -y_k \ -1 \ 0 \mid 0 \ 0 \ 1 \ 1]$. Soit N_i le vecteur bidimensionnel (x_i, y_i) . Le vecteur $B^{-1}b$ est égal à $[\frac{N_k \wedge N_j}{\det(B)}; \frac{N_i \wedge N_k}{\det(B)}; \frac{N_j \wedge N_i}{\det(B)}; 1]$. Comme la matrice B est non singulière et $\det(B) = -\det([x_i \ y_i \ 1 \mid x_j \ y_j \ 1 \mid x_k \ y_k \ 1])$, les trois points p_i, p_j, p_k ne peuvent pas être colinéaires. Supposons que les trois points $N_i N_j N_k$ soient orientés dans le sens horaire, alors $\det(B) \geq 0$. Et comme $B^{-1}b \geq 0$, alors $N_k \wedge N_j, N_i \wedge N_k$ et $N_j \wedge N_i$ sont positifs. Une telle situation ne peut apparaître que lorsque le point p_{l-n} se situe dans le triangle $N_i N_j N_k$, relativement à la projection

dans le plan Oxy .

- **Configuration 2** : $1 \leq i, j \leq n < k, l \leq 2n$. Le plan P (resp. P') est supporté par le segment $p_i p_j$ (resp. $p_{k-n} p_{l-n}$). Puisqu'ils sont parallèles, ce couple de plan est unique. Considérons que le point p_{l-n} est centré à l'origine, nous avons $B^{-1}b = [N_k \wedge N_j / \Delta; N_i \wedge N_k / \Delta; \dots; \dots]$ avec $\Delta = (N_i - N_j) \wedge N_k$. Δ n'est pas nul si et seulement si le segment $p_i p_j$ et le segment $p_{k-n} p_{l-n}$ ne sont pas colinéaires dans le plan Oxy . Il en résulte que $N_k \wedge N_j$ et $N_k \wedge N_i$ n'ont pas le même signe. Ainsi, le segment $p_i p_j$ coupe la ligne portée par $(p_{k-n} p_{l-n})$. Quand c'est p_i qui est centré à l'origine, nous obtenons de manière symétrique le même résultat. Par conséquent, ce cas est associé à deux segments $p_i p_j$ et $p_{k-n} p_{l-n}$ qui s'intersectent dans la projection dans le plan Oxy .
- **Configuration 3** : $1 \leq i \leq n < j, k, l \leq 2n$. Ce cas est équivalent à la première configuration, avec cette fois-ci un plan supérieur défini par trois points et un plan inférieur passant par un unique point.

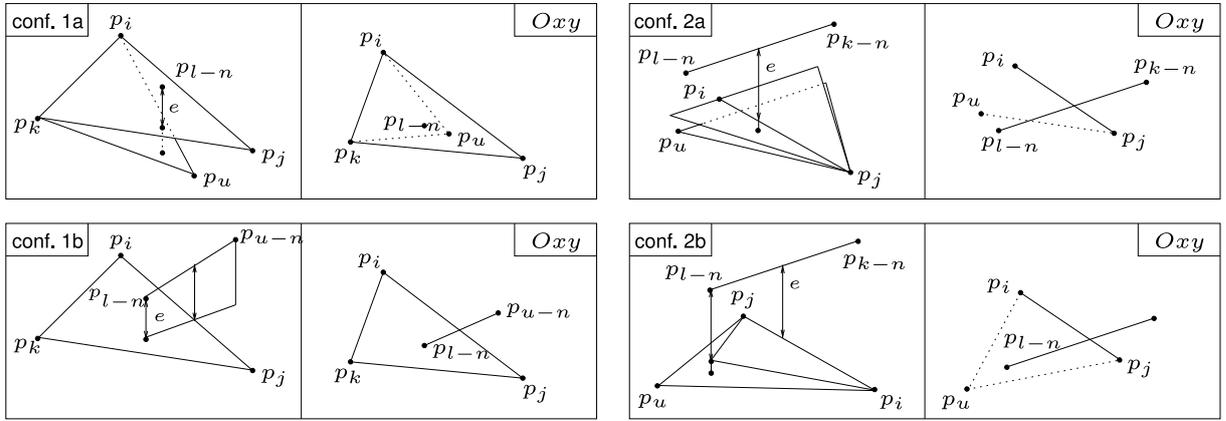


Figure 3.8 – Différentes configurations, en fonction de la base et des variables entrantes.

Échanges des variables : Nous parcourons l'ensemble des points de \mathcal{S}_b . Pour chaque point, nous considérons sa distance verticale par rapport à P lorsqu'il se situe au-dessous de P , ou par rapport à P' quand il se situe au-dessus de P' . Si aucun point n'est trouvé, alors le problème est résolu puisque tous les points sont cloisonnés entre les deux plans. Sinon, nous sélectionnons le point dont la distance verticale considérée est la plus grande. La variable associée à ce point, λ_u , entre alors dans la base.

Dans la **configuration 1**, nous avons trois inégalités du type : $\alpha.x + \beta.y + h = z$. Quand nous sélectionnons une variable λ_u du même type, c'est-à-dire avec $1 \leq u \leq n$, nous ne pouvons pas retirer p_{l-n} , sinon nous obtiendrions une base avec quatre égalités de même type et cette configuration n'est pas possible. La nouvelle base va donc rester en configuration 1. Ainsi, le nouveau point p_u remplace l'unique point parmi p_i, p_j et p_k qui permettra de conserver la contrainte que p_{l-n} se situe à l'intérieur du nouveau triangle dans la projection sur le plan Oxy . Comme p_u est sous le plan P défini par $p_i p_j p_k$, alors l'épaisseur courant e a aussi augmenté (cf. figure 3.8, conf. 1a).

Dans l'autre cas, quand $n < u \leq 2n$, deux possibilités sont envisageables. Si p_{u-n} se situe à l'intérieur du triangle, il remplace simplement son homologue p_{l-n} et e augmente. Si p_{u-n} n'appartient pas au triangle, nous ne pouvons pas atterrir dans une configuration de type 1. Nous passons alors à une configuration de type 2. Pour cela, le segment qui supporte P' est $p_{u-n} p_{l-n}$. L'autre segment définissant la configuration correspond à l'unique arête du triangle qui coupe $p_{u-n} p_{l-n}$ dans la projection sur le plan Oxy . p_{u-n} se situe à une distance verticale supérieure à celle définie par le triangle et p_{l-n} . De plus, cette distance est égale à la distance entre les deux segments retenus, et par conséquent la valeur de e augmente (cf. figure 3.8, conf. 1b).

Dans la **configuration 2**, lorsqu'une λ_u , $1 \leq u \leq n$ est sélectionnée, il y a deux possibilités. Pour rester dans la même configuration, p_u doit remplacer un point de sorte que les deux nouveaux segments

s'intersectent dans la projection sur le plan Oxy (cf. figure 3.8, conf. 2a). Lorsque cela n'est pas faisable, un des deux points p_{k-n} ou p_{l-n} appartient inévitablement au triangle $p_i p_j p_u$ et nous passons alors à une configuration de type 1 (cf. figure 3.8, conf. 2b).

3.4.1.4 Convergence et complexité

Comme le problème primal est faisable (on trouvera toujours une valeur e assez grande), le problème dual n'est jamais non borné et nous pouvons donc supprimer le traitement de ce cas particulier dans l'algorithme. Comme nous sélectionnons à chaque fois un point en dehors de la zone entre les deux plans parallèles, nous savons que la distance verticale (la fonction objectif) croît strictement à chaque itération. Ainsi, au contraire du cas général, l'algorithme du simplexe appliqué au problème de reconnaissance de MFPD ne peut pas boucler. Enfin, nous avons au plus $C_{2n+5}^4 = \mathcal{O}(n^4)$ bases faisables possibles, ce qui place la complexité en temps au pire cas à $\mathcal{O}(k^4)$, où k représente le nombre de sommets de l'enveloppe convexe de l'ensemble de points \mathcal{S}_b . Et en pratique, k est généralement relativement petit par rapport au nombre de points dans \mathcal{S}_b .

3.4.1.5 Une version incrémentale

Afin d'utiliser les algorithmes de reconnaissance de primitives discrètes dans un cadre de segmentation ou de polyédrisation, il est toujours intéressant de disposer d'une version incrémentale, qui permet notamment de contrôler l'ordre d'ajout des points.

Pour l'algorithme proposé ici, l'ajout d'un point se traduit par un agrandissement de matrice. Ainsi,

lorsqu'un nouveau point $p(x, y, z)$ est inséré, les deux colonnes $\begin{bmatrix} -x \\ -y \\ -1 \\ 0 \end{bmatrix}$ et $\begin{bmatrix} x \\ y \\ 1 \\ 1 \end{bmatrix}$ sont ajoutées à la matrice

A du dual.

Deux cas sont alors à envisager. Tout d'abord p est susceptible de se trouver entre les deux plans P et P' . Dans ce cas, la solution précédente reste optimale et il n'y a rien à faire. Dans l'autre cas, on utilise la dernière base calculée, associée à la solution optimale précédente, pour amorcer une nouvelle séquence d'itérations jusqu'à ce qu'une nouvelle solution optimale soit trouvée.

3.4.1.6 Synthèse

Nous avons proposé dans cette section un algorithme de reconnaissance de MFPD permettant de résoudre le problème en considérant la norme infinie $\|\cdot\|_\infty$. Après avoir reformulé notre problème en termes de Programmation Linéaire, nous avons pu appliquer une méthode bien connue de ce domaine, l'algorithme du simplexe, pour pouvoir résoudre notre problème d'optimisation. L'avantage, une fois la formulation du problème sous forme d'un problème d'optimisation effectuée, est que d'autres méthodes de PL, comme celle des points intérieurs, peuvent aussi être employées pour le résoudre. Au vu des résultats théoriques (meilleure complexité au pire cas, par exemple) ces dernières pourraient même s'avérer plus efficaces. Néanmoins nous avons choisi d'utiliser ici l'algorithme du simplexe car il permettait de fournir une interprétation géométrique intéressante de chaque opération de l'algorithme du simplexe.

Cette méthode a d'ailleurs fait l'objet d'une première partie d'un article publié, en collaboration avec Lilian Buzer, à la conférence internationale Discrete Geometry for Computer Imagery en 2006 [PBDR06]. J'ai aussi programmé la version incrémentale de l'algorithme, et le code source ainsi qu'un programme de test sont disponibles à l'adresse suivante :

http://www.loria.fr/~debled/BlurredPlane/LPMethod/lp_algorithm.html.

Un petit désavantage de cette technique, néanmoins, est qu'il faut en pratique gérer trois instances du programme, une pour chaque situation $\|(a, b, c)\|_\infty = a$, $\|(a, b, c)\|_\infty = b$ ou $\|(a, b, c)\|_\infty = c$. En effet, il n'est généralement pas possible de savoir à l'avance lequel des trois axes principaux de \mathbb{R}^3 sera l'axe principal du MFPD reconnu. Dans la section suivante, nous allons donc proposer un autre algorithme de reconnaissance qui lui n'a pas ce désavantage.

3.4.2 Une approche géométrique

Nous présentons ici un algorithme de reconnaissance de MFPD que nous avons développé pour résoudre le problème en considérant la norme $\|\cdot\|_2$. L'approche exploite en fait la géométrie de l'enveloppe convexe de l'ensemble de points S_b , en énumérant l'ensemble des paires de plans euclidiens parallèles « intéressants », afin de conserver celles pour lesquelles la distance euclidienne entre les deux plans de la paire est la plus petite.

Plus précisément, nous avons combiné une méthode destinée à calculer la largeur d'un ensemble de points de \mathbb{R}^3 [GH01] avec une méthode de mise à jour incrémentale d'enveloppe convexe [dBvKOS00], afin de proposer un algorithme incrémental de reconnaissance de MFPD, relativement à la norme $\|\cdot\|_2$. Nous présentons dans la suite de cette section ce qu'est la *largeur* d'un ensemble de points réels et explicitons le lien avec notre problème de reconnaissance. Nous expliquons ensuite comment la calculer et présentons les étapes de mise à jour incrémentale d'une enveloppe convexe pour finir sur notre algorithme incrémental de reconnaissance de MFPD.

La définition de la largeur d'un ensemble de points repose sur la notion de *plan support*, dont la définition, ci-dessous, est illustrée à la figure 3.9 :

Définition 3.13 Soit E un ensemble de points de \mathbb{R}^3 et \mathcal{P} un plan réel. On dit que \mathcal{P} est un **plan support** de E si tous les points de E appartiennent à un seul des demi-espaces délimités par \mathcal{P} et tels que $\mathcal{P} \cap E \neq \emptyset$.

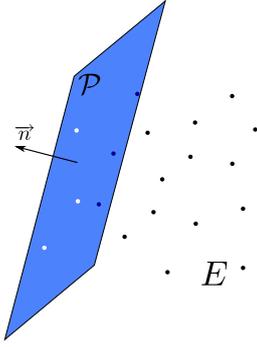


Figure 3.9 – En bleu un plan support de l'ensemble de points E . Les points se situent tous à droite de \mathcal{P} (dans le demi-espace négatif délimité par le plan) et les points blancs appartiennent à \mathcal{P} .

Définition 3.14 Soit E un ensemble de points de \mathbb{R}^3 . On appelle **largeur** de E , la plus petite distance (euclidienne) entre deux plans support parallèles distincts de E .

Ainsi, si E est un ensemble de points de \mathbb{Z}^3 , alors deux plans support parallèles dont la distance est égale à la largeur de E coïncident avec les plans d'appui d'un plan englobant optimal de E et la largeur de E est égale à la hauteur de ce plan englobant optimal. Par conséquent, *calculer la largeur d'un ensemble de points et une paire de plans support parallèles associés nous permet de résoudre le problème de reconnaissance de MFPD*.

3.4.2.1 Calcul de la largeur d'un ensemble de points E

Nous recherchons deux plans euclidiens parallèles $\mathcal{P}_1 : \alpha x + \beta y + \gamma z + \delta_1 = 0$ et $\mathcal{P}_2 : \alpha x + \beta y + \gamma z + \delta_2 = 0$ qui minimisent la distance $\frac{|\delta_2 - \delta_1|}{\sqrt{\alpha^2 + \beta^2 + \gamma^2}}$ entre \mathcal{P}_1 et \mathcal{P}_2 et tels que, pour tous points $p(p_x, p_y, p_z) \in E$:

$$\begin{aligned} \alpha p_x + \beta p_y + \gamma p_z + \delta_1 &\leq 0 \\ \alpha p_x + \beta p_y + \gamma p_z + \delta_2 &\geq 0 \end{aligned}$$

Comme l'enveloppe convexe d'un ensemble E est l'intersection de tous les demi-espaces contenant entièrement E , alors la largeur de E est égale à la largeur de son enveloppe convexe $conv(E)$ [HT88]. De plus, l'introduction de la notion de *paires antipodales* permet de simplifier le problème.

Définition 3.15 Soient E un ensemble de points et $\text{conv}(E)$ son enveloppe convexe. Deux arêtes a_1 et a_2 de $\text{conv}(E)$ sont dites **antipodales** s'il existe deux plans support de E parallèles, dont l'un contient a_1 et l'autre contient a_2 . On dit aussi que a_1 et a_2 forment une **paire arête-arête antipodale**. On définit de la même manière les paires **sommet-sommet**, **sommet-arête**, **sommet-face**, **arête-face** et **face-face** antipodales.

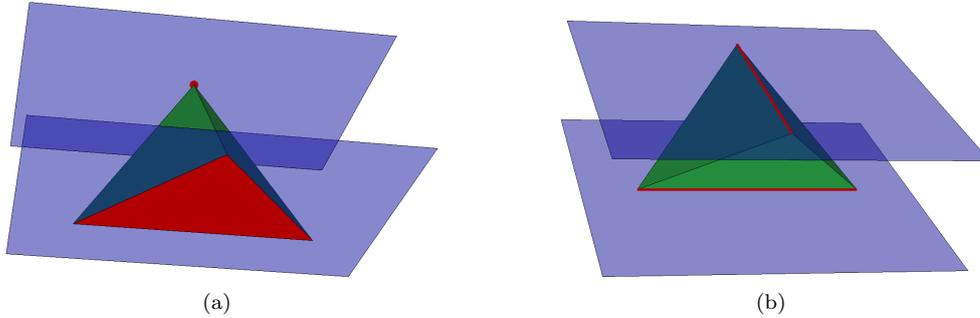


Figure 3.10 – Les deux paires antipodales pour lesquelles la distance entre les deux plans parallèles qu'elles supportent peut être minimale : (a) une paire sommet-face antipodale et (b) une paire arête-arête antipodale. Les éléments définissant la paire antipodale sont représentés en rouge et les deux plans support parallèles en bleu.

Dans [HT88], M.E. Houle et G.T. Toussaint montrent que pour calculer la largeur de E il suffit de s'intéresser aux plans support définis par une paire antipodale *sommet-face* ou *arête-arête* (cf. figure 3.10). Notre objectif est donc d'énumérer toutes les paires antipodales *sommet-face* et *arête-arête* de $\text{conv}(E)$ et de retenir celle qui définit des plans support parallèles dont la distance est minimale.

Dans [GH01], B. Gärtner et T. Herrmann proposent une approche directe reposant sur les propriétés géométriques et combinatoires de l'enveloppe convexe. Le procédé d'énumération de l'ensemble des paires antipodales recherchées s'inspire de la méthode du *rotating caliper* [Tou83], mais généralisée en dimension 3. La technique est la suivante : une face de départ f est choisie arbitrairement sur l'enveloppe convexe $\text{conv}(E)$ et les sommets antipodaux $V = v_1, \dots, v_k$ sont recherchés en explorant tous les sommets de $\text{conv}(E)$. Ceci permet d'obtenir une paire antipodale face-sommet initiale et les deux plans parallèles associés, \mathcal{P}_1 et \mathcal{P}_2 supportant respectivement V et f . Une rotation de ces deux plans est ensuite effectuée autour d'une arête a incidente à f , jusqu'à ce que \mathcal{P}_2 contienne l'autre face f' incidente à a . Pendant la rotation, la propriété d'être des plans support et le parallélisme entre \mathcal{P}_1 et \mathcal{P}_2 sont préservés, et on reporte toutes les paires arête-arête antipodales contenant a ainsi que les paires face-sommet antipodales contenant f' .

Ainsi, étant donné une paire sommet-arête antipodale (w, a) et des plans parallèles \mathcal{P}_1 et \mathcal{P}_2 contenant respectivement w et a , lors de la rotation de \mathcal{P}_2 autour de a , deux événements d'importance peuvent se produire :

1. \mathcal{P}_2 rencontre une nouvelle face f' incidente à a , auquel cas une paire sommet-face antipodale (w, f') vient d'être détectée,
2. \mathcal{P}_1 rencontre un nouveau sommet v , et dans ce cas, une paire arête-arête antipodale $((wv), a)$ est détectée.

Par conséquent, une rotation autour d'une arête a de $\text{conv}(E)$ permet de relever toutes les paires arête-arête antipodales contenant a et toutes les paires sommet-face antipodales contenant les deux faces incidentes à a . Donc, en effectuant la rotation autour de toutes les arêtes de $\text{conv}(E)$, on relève toutes les paires antipodales sommet-face et arête-arête possibles de $\text{conv}(E)$. Au moins l'une d'elles définit deux plans support parallèles pour lesquels la distance \mathcal{H} qui les sépare est minimale et correspond à la largeur de E .

Comme \mathcal{H} représente la hauteur d'un plan englobant optimal $\mathcal{P}(a, b, c, \mu, \omega)$ de E , si $\mathcal{H} \leq \nu$, E est un morceau flou de plan discret d'épaisseur ν .

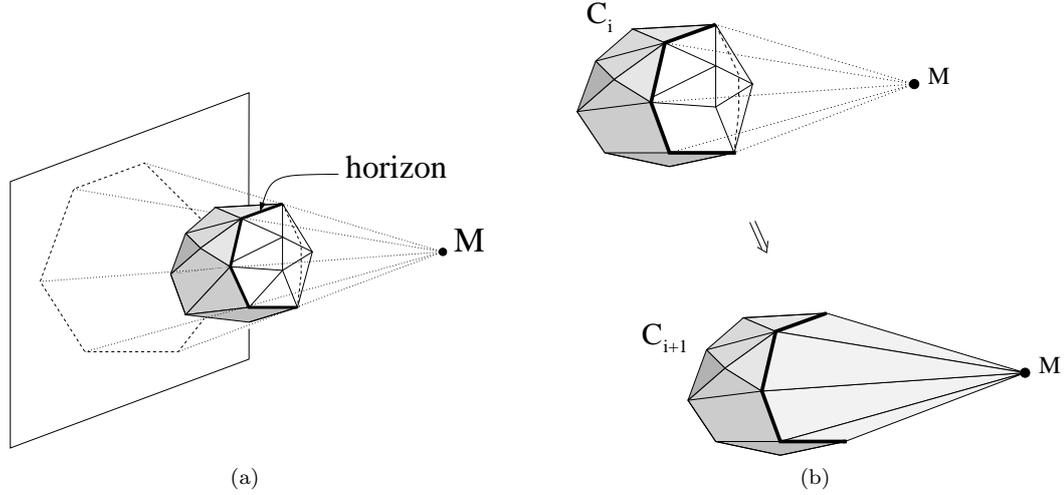


Figure 3.11 – (a) La ligne d'horizon de l'enveloppe convexe. (b) L'évolution de l'enveloppe convexe après ajout du point M .

De plus, les plans support \mathcal{P}_1 et \mathcal{P}_2 coïncident avec les plans d'appui de $\mathcal{P}(a, b, c, \mu, \omega)$, donc nous obtenons $a = \alpha$, $b = \beta$, $c = \gamma$, $\omega = |\delta_2 - \delta_1| + 1$ et $\mu = \min(-\delta_1, -\delta_2)$ puisque l'équation d'un plan d'appui est $ax + by + cz = \mu$.

3.4.2.2 Algorithme de reconnaissance incrémental

Nous proposons maintenant une version incrémentale de la méthode décrite précédemment, de façon à obtenir un algorithme qui, à chaque ajout de point à un ensemble E , retourne les caractéristiques du plan englobant optimal de E .

Dans le cadre du processus incrémental, l'ensemble E ne diffère que d'un point entre chaque ajout de point à E . Nous pouvons donc avantageusement utiliser un calcul incrémental de l'enveloppe convexe comme celui proposé dans [dBvKOS00] pp 235–246). Rappelons brièvement la technique.

À une étape i du processus, nous avons une enveloppe convexe C_i et nous ajoutons un point M . S'il se trouve à l'intérieur de (ou sur) C_i alors nous n'avons rien à faire. Dans le cas contraire, nous recherchons toutes les faces de C_i visibles⁹ depuis M . Cet ensemble de faces forme une région délimitée par une courbe polygonale qu'on appelle *horizon* (cf. figure 3.11(a)). Ces faces sont alors supprimées de C_i et remplacées par de nouvelles faces triangulaires obtenues en joignant chaque extrémité des segments formant l'horizon au point M (cf. figure 3.11(b)). Certaines des nouvelles faces de l'enveloppe convexe peuvent être coplanaires à des faces non visibles depuis M dans C_i ; il faut alors fusionner ces faces coplanaires. Le polyèdre résultant est la nouvelle enveloppe convexe C_{i+1} .

De plus, il n'est pas nécessaire de calculer la largeur de l'ensemble à chaque étape. En effet, pour chaque point ajouté, nous connaissons les caractéristiques a , b , c , μ et ω d'un plan englobant optimal \mathcal{P} de E . Nous pouvons donc calculer la valeur du reste $r_{\mathcal{P}}(M) = ax_M + by_M + cz_M - \mu$ au point M relativement à \mathcal{P} , et dans le cas où $r_{\mathcal{P}}(M) \in [0, \omega - 1]$ alors $M \in \mathcal{P}$, et par conséquent il est inutile de recalculer la largeur de E .

Nous obtenons finalement l'algorithme 3.2 décrivant la procédure permettant de mettre au point la reconnaissance incrémentale d'un morceau flou de plan discret d'épaisseur ν .

9. Considérons un plan \mathcal{P}_f contenant une face f de l'enveloppe convexe. Alors \mathcal{P}_f divise l'espace en deux sous-espaces, l'un qui contient tous les points de l'enveloppe convexe et que l'on qualifie de fermé et l'autre que l'on qualifie d'ouvert. Alors on dit que f est *visible* depuis un point p , si ce dernier se situe dans le demi-espace ouvert délimité par \mathcal{P}_f .

Algorithme 3.2 : Étape de la reconnaissance incrémentale d'un MFPD $_{\nu}$

Données : Un ensemble de points $\mathcal{E} \in \mathbb{Z}^3$, l'enveloppe convexe C de \mathcal{E} , les caractéristiques a, b, c, μ et ω d'un plan englobant optimal de \mathcal{E} et ν l'épaisseur maximale autorisée

Entrée : Un point $M \in \mathbb{Z}^3$

Résultat : Les données mises à jour après l'ajout de M et le booléen `estMFPD` vrai si $\mathcal{E} \cup \{M\}$ est morceau flou de plan discret d'épaisseur ν , faux sinon

```

1 début
2    $\mathcal{E} \leftarrow \mathcal{E} \cup \{M\}$ 
3   Mise à jour de  $C$  selon le procédé incrémental
4    $r_M \leftarrow ax_M + by_M + cz_M - \mu$ 
5   si  $r_M \notin [0, \omega - 1]$  alors
6      $\langle \alpha, \beta, \gamma, \delta_1, \delta_2 \rangle \leftarrow \text{CalculPlanSupportMin}(C)$ 
7      $a \leftarrow \alpha$ 
8      $b \leftarrow \beta$ 
9      $c \leftarrow \gamma$ 
10     $\mu \leftarrow \min(-\delta_1, -\delta_2)$ 
11     $\omega \leftarrow |\delta_2 - \delta_1| + 1$ 
12     $\text{estMFPD} \leftarrow \frac{\omega-1}{\sqrt{a^2+b^2+c^2}} \leq \nu$ 
13 fin

```

La fonction `CalculPlanSupportMin(C)` appelée à la ligne 6 de l'algorithme 3.2 permet de calculer les plans support parallèles de C entre lesquels la distance euclidienne est minimale grâce à la méthode décrite en 3.4.2, et les coefficients de ces plans sont retournés.

3.4.2.3 Complexité

Dans [GH01], Gärtner et Herrmann démontrent que la fonction `CalculPlanSupportMin(C)` a une complexité en $\mathcal{O}(n^2)$ où n désigne le nombre de points de E . La mise à jour de l'enveloppe convexe après ajout d'un point s'effectue en $\mathcal{O}(n)$. Comme les autres instructions de l'algorithme 3.2 s'effectuent toutes en temps constant, on obtient une complexité en $\mathcal{O}(n^2)$ pour notre procédure incrémentale. Étant donné qu'il faut appliquer cette procédure incrémentale à chaque fois que l'on ajoute un point à E , cela nous donne une complexité totale en $\mathcal{O}(n^3)$ dans le pire cas, pour un ensemble E de n points. Cependant, cette complexité ne semble généralement pas atteinte en pratique, comme le confirment les résultats présentés dans la section 3.5.

3.4.2.4 Synthèse

Nous avons proposé dans cette section un algorithme de reconnaissance de MFPD permettant de résoudre le problème en considérant la norme euclidienne $\|\cdot\|_2$. Ce dernier exploite la géométrie de l'enveloppe convexe des points considérés en énumérant toutes les paires antipodales susceptibles de définir des configurations de plans parallèles « intéressantes », et conserve celle pour laquelle la distance euclidienne entre les deux plans qui la compose est minimale. En fait, dans certains cas, la distance minimale est atteinte au niveau de plusieurs paires antipodales différentes. Si les caractéristiques du MFPD nous intéressent, nous pouvons alors conserver la liste de ces paires antipodales, et ainsi retrouver l'ensemble des MFPD solutions. La version incrémentale de l'algorithme repose sur une mise à jour incrémentale de l'enveloppe convexe et une procédure de rotation autour des arêtes de cette enveloppe convexe (similaire au *rotating caliper* mais adaptée en 3D) qui n'est effectuée que lorsque le reste d'un point par rapport au MFPD reconnu jusqu'alors n'est pas compatible avec les caractéristiques de ce dernier.

Cette méthode a fait l'objet de la seconde partie d'un article publié à la conférence internationale Discrete Geometry for Computer Imagery en 2006 [PBDR06]. J'ai aussi programmé cet algorithme, et le code source ainsi qu'un programme de test sont disponibles à l'adresse suivante :

http://www.loria.fr/~debled/BlurredPlane/GeometricalMethod/geometrical_algorithm.html.

3.4.3 Adaptation de méthodes existantes

Dans cette section, nous indiquons comment modifier certaines techniques de reconnaissance pour qu'elles puissent reconnaître des MFPD $_{\nu}$.

3.4.3.1 Méthode utilisant l'ensemble des cordes

L'algorithme de reconnaissance présenté à la section 1.8.3, utilise, pour un ensemble S de points de \mathbb{Z}^3 , l'enveloppe convexe de $chord(S)$ pour savoir s'il existe un plan discret naïf contenant S .

Cet algorithme, comme nous l'avons vu au premier chapitre, repose sur le théorème 1.2, qui dit que, pour un ensemble fini $S \subset \mathbb{Z}^3$ et dans le cas où $c = \max(|a|, |b|, |c|)$, il existe un plan discret naïf $\mathcal{P}(a, b, c, \mu, c)$ tel que $S \subset \mathcal{P}$ si et seulement si $\tau_{Oz}(S) < 1$.

Si nous nous plaçons dans les mêmes conditions que [GDRZ05], où, sans perte de généralité, $c = \max(|a|, |b|, |c|)$, alors ce théorème peut être généralisé pour reconnaître des plans discrets épais, de la façon suivante :

Théorème 3.1 *Soit $S \subset \mathbb{Z}^3$ un ensemble fini et $\omega \in \mathbb{N}^*$, alors il existe $(a, b, c, \mu) \in \mathbb{Z}^4$ vérifiant $\forall (x, y, z) \in S, \mu \leq ax + by + cz \leq \mu + \omega - 1$ si et seulement si $\tau_{Oz}(S) \leq \frac{\omega-1}{c}$.*

La preuve est similaire à celle faite dans [GDRZ05], en substituant 1 par $\frac{\omega-1}{c}$.

Preuve : Supposons dans un premier temps qu'il existe $(a, b, c, \mu) \in \mathbb{Z}^4$ vérifiant $\mu \leq ax + by + cz \leq \mu + \omega - 1$ pour tout point $(x, y, z) \in S$. Soient $m = (x, y, z)$ et $m' = (x', y', z')$ deux points de S . La double inégalité précédente est donc vérifiée pour ces deux points et nous avons $-\omega + 1 \leq a(x - x') + b(y - y') + c(z - z') \leq \omega - 1$. Ceci prouve que tous les points $M = (x'', y'', z'')$ de $chord(S)$ vérifient la double inégalité $-\omega + 1 \leq ax'' + by'' + cz'' \leq \omega - 1$. C'est donc en particulier le cas pour les points de $conv(chord(S))$. Comme le point $(0, 0, \tau_{Oz}(S))$ est, par définition, l'un de ces points, alors nous avons $-\omega + 1 \leq a.0 + b.0 + c.\tau_{Oz}(S) \leq \omega - 1$, et donc, en particulier, $\tau_{Oz}(S) \leq \frac{\omega-1}{c}$, qui était l'inégalité recherchée.

Réciproquement, supposons que l'épaisseur de S dans la direction Oz est inférieure ou égale à $\frac{\omega-1}{c}$. L'ensemble $conv(chord(S))$ est un polytope convexe de \mathbb{Z}^3 contenant l'origine. Considérons une de ses facettes coupant la demi-droite Oz . Par définition, $(0, 0, \tau_{Oz}(S))$ est un point de cette facette et nous notons (a, b, c) , avec $c > 0$ et $(a, b, c) \in \mathbb{Z}^3$, son vecteur normal. Pour n'importe quel point $M = (x, y, z)$ de $conv(chord(S))$, et n'importe quel point M de $chord(S)$ d'ailleurs, nous avons l'inégalité

$$\begin{aligned} ax + by + cz &\leq a.0 + b.0 + c.\tau_{Oz}(S) \\ \text{c'est-à-dire} \quad ax + by + cz &\leq c\tau_{Oz}(S) \end{aligned} \tag{i}$$

Soient alors deux points de S , $m_{\min} = (x_{\min}, y_{\min}, z_{\min})$ et $m_{\max} = (x_{\max}, y_{\max}, z_{\max})$, réalisant respectivement le minimum et le maximum de $ax + by + cz$ sur l'ensemble fini S . Si nous notons $h = ax_{\min} + by_{\min} + cz_{\min}$ et $H = ax_{\max} + by_{\max} + cz_{\max}$, alors nous avons la double inégalité

$$\forall (x, y, z) \in S, \quad h \leq ax + by + cz \leq H \tag{ii}$$

Avec $M = m_{\max} - m_{\min}$, il découle de l'inégalité (i) que $a(x_{\max} - x_{\min}) + b(y_{\max} - y_{\min}) + c(z_{\max} - z_{\min}) \leq c\tau_{Oz}(S)$, à savoir $H - h \leq c\tau_{Oz}(S)$. L'inégalité $\tau_{Oz}(S) \leq \frac{\omega-1}{c}$ nous donne alors $H \leq h + \omega - 1$ et (ii) implique que $\forall (x, y, z) \in S, h \leq ax + by + cz \leq h + \omega - 1$. Par conséquent, avec $\mu = h$, il existe bien $(a, b, c, \mu) \in \mathbb{Z}^4$ vérifiant $\forall (x, y, z) \in S, \mu \leq ax + by + cz \leq \mu + \omega - 1$, ce qui termine la preuve. \square

Ainsi, pour un ensemble S de points de \mathbb{Z}^3 , en considérant la norme $\|\cdot\|_{\infty}$ et en utilisant la technique incrémentale de recherche du plus haut triangle de $conv(chord(S))$ rappelée à la section 1.8.3, nous obtenons un algorithme de reconnaissance de MFPD. Nous devons évidemment effectuer cette procédure

dans les trois directions principales de l'espace (c'est-à-dire calculer $\tau_{Ox}(S)$, $\tau_{Oy}(S)$ et $\tau_{Oz}(S)$). Soit $h = \min(\tau_{Ox}(S), \tau_{Oy}(S), \tau_{Oz}(S))$, et (a, b, c, μ) les paramètres, obtenus à partir du plus haut triangle associé à ce minimum, renvoyés par l'algorithme. Alors en prenant $\omega = c.h + 1$ nous avons que $\mathcal{P}(a, b, c, \mu, \omega)$ est un plan englobant optimal de S , comme conséquence du théorème 3.1.

3.4.3.2 Méthodes basées sur la préimage

Nous pouvons, dans le même ordre d'idées, définir une préimage pour reconnaître des MFPD $_{\nu}$. Nous reprenons la définition analytique d'un plan discret épais de normale (a, b, c) , de paramètre de translation μ et d'épaisseur arithmétique ω , avec a, b, c, μ, ω entiers :

$$\mathcal{P}(a, b, c, \mu, \omega) = \{(x, y, z) \in \mathbb{Z}^3 \mid \mu \leq ax + by + cz \leq \mu + \omega - 1\}$$

Nous nous plaçons dans le cas où $c = \max(|a|, |b|, |c|)$. Pour un point $(x_0, y_0, z_0) \in \mathcal{P}$ nous avons $\mu \leq ax_0 + by_0 + cz_0 \leq \mu + \omega - 1$, c'est-à-dire $0 \leq \frac{a}{c}x_0 + \frac{b}{c}y_0 + z_0 - \frac{\mu}{c} \leq \frac{\omega-1}{c}$. En posant $\nu(\alpha, \beta, \gamma) = \frac{\omega-1}{c}$, l'ensemble des plans à ν constant passant par le point (x_0, y_0, z_0) est donc décrit par les paramètres α, β, γ rationnels qui vérifient $0 \leq \alpha x_0 + \beta y_0 + \gamma z_0 - \nu \leq \nu$, avec $|\alpha| \leq 1, |\beta| \leq 1$. Nous retrouvons un espace des paramètres proche de celui défini à la section 1.8.2.1, néanmoins nous ne considérons que les points rationnels de ce dernier. De plus, pour un ensemble X de points de \mathbb{Z}^3 , chaque point $(x, y, z) \in X$ engendre dans l'espace dual une bande épaisse $B(x, y, z) = \begin{cases} \alpha x + \beta y + \gamma + z \leq 0 \\ \alpha x + \beta y + \gamma + z \geq \nu \end{cases}$.

La préimage pour reconnaître des MFPD $_{\nu}$, considérés selon la norme $\|\cdot\|_{\infty}$, est donc définie, pour un ensemble de points X , comme l'ensemble $\mathcal{Pr}(X) = \{(\alpha, \beta, \gamma) \in \mathbb{Q}^3, |\alpha| \leq 1, |\beta| \leq 1 \mid 0 \leq \alpha x + \beta y + \gamma + z \leq \nu(\alpha, \beta, \gamma)\}$.

Notons que pour une épaisseur ν fixée, savoir si un ensemble X de points de \mathbb{Z}^3 est un MFPD $_{\nu}$ revient à regarder si $\mathcal{Pr}(X)$ est vide ou non. Par contre, trouver l'épaisseur ν minimale pour que X soit un MFPD $_{\nu}$ est plus compliqué. En effet, ν devient alors un paramètre, et nous devons considérer une préimage d'une dimension supérieure. Nous revenons alors à une formulation du problème identique à celle donnée à la section 3.4.1.2 et que nous avons résolu avec l'algorithme du simplexe.

Nous ne l'avons pas fait ici, mais il pourrait être intéressant d'étudier la structure d'une telle préimage. Certaines propriétés pourraient amener à la mise au point d'algorithmes plus efficaces, ou permettant d'avoir une compréhension plus aboutie des morceaux flous de plans discrets.

3.5 Résultats expérimentaux

Nous proposons dans cette section une étude expérimentale permettant de mieux cerner le comportement de l'algorithme 3.2 de reconnaissance incrémentale de MFPD en pratique.

Nous avons préféré nous concentrer sur ce dernier plutôt que sur l'algorithme utilisant la norme infinie (présenté en section 3.4.1) car le comportement des algorithmes de résolution de problèmes d'optimisation fait l'objet d'études spécialisées dans des domaines comme la Recherche Opérationnelle. Les résultats que nous présentons ci-dessous sont donc uniquement ceux obtenus à l'aide de la méthode géométrique (section 3.4.2).

Afin d'étudier le comportement de notre algorithme dans différents types de situations, nous avons choisi de faire varier les paramètres suivants :

1. le nombre de points en entrée ;
2. le type d'entité représenté par les points en entrée de notre algorithme :
 - morceaux de plans discrets épais ;
 - huitièmes de sphères discrètes ;
 - morceaux de surfaces discrètes fonctionnelles.
3. le type de parcours employé lors de l'ajout des points :

- parcours aléatoire ;
- parcours en suivant l'ordre lexicale canonique sur les coordonnées des points ;
- parcours géodésique approché par rapport à un point central.

Faire varier le nombre n de points permet évidemment d'avoir une idée du comportement en pratique de l'algorithme, et notamment de savoir si ce dernier s'approche du comportement théorique déterminé dans la section 3.4.2.3, à savoir une complexité au pire cas en $\mathcal{O}(n^3)$.

Le type d'entité discrète manipulée à travers l'ensemble de points en entrée a été choisi pour représenter des cas intéressants pour certaines parties de l'algorithme. Par exemple, un bout de sphère représente un cas critique pour le calcul de l'enveloppe convexe, puisqu'on sait que chaque point ajouté (si l'on considère une sphère dans \mathbb{R}^3) va entraîner une modification de l'enveloppe convexe et probablement nécessiter un nouveau calcul de la hauteur du MFPD. Le morceau de plan discret, au contraire, représente un cas plus favorable où la probabilité de recalculer la hauteur du MFPD reste assez faible, puisqu'à partir d'un certain temps, du fait de la structure d'un plan discret épais, les points ajoutés auront de grandes chances d'avoir un reste compatible avec le MFPD déjà reconnu. Mais ceci reste aussi très dépendant de l'ordre d'ajout des points lors de la reconnaissance. C'est pourquoi nous avons choisi d'explorer différents parcours de l'ensemble de points pour pouvoir comprendre leur influence sur l'algorithme. Ainsi, le parcours aléatoire permet en quelque sorte de représenter une situation générale où l'on ajoute des points sans connaissance a priori sur l'ensemble manipulé. Lorsque la structure discrète permettant de représenter l'objet manipulé est un tableau 3D (classique en informatique), une manière naturelle d'ajouter des points est de parcourir le tableau par ordre croissant des indices. Le parcours en suivant l'ordre lexicographique canonique sur les coordonnées des points permet de simuler une telle situation. Enfin le parcours géodésique approché (cf. section 4.2.2.2.0 du chapitre suivant, page 86) permet quand à lui de représenter des situations où la reconnaissance s'effectue à partir d'un point initial en « étendant » le MFPD de manière homogène dans toutes les directions. Cette manière de faire est souvent employée dans des algorithmes de segmentation, par exemple, où la recherche de segment peut se faire par croissance de région.

Les sections qui suivent présentent chacune l'évolution des temps de calcul de l'algorithme 3.2 en fonction du nombre de points en entrée, pour un des différents objets discrets énumérés précédemment. Dans chaque cas, l'ajout des points est effectué selon les différents parcours proposés, et représenté visuellement sur une image qui accompagne chacun des graphes. L'échelle en niveaux de gris utilisée dans les images est un dégradé du noir, pour les voxels ajoutés en premier, au blanc, pour les voxels ajoutés en dernier.

3.5.1 Morceaux de plans discrets

Les résultats de l'algorithme de reconnaissance de MFPD sur des morceaux de plans discrets sont représentés à la figure 3.12.

Chaque graphe représente les temps de calcul en secondes (en ordonnées) de l'algorithme de reconnaissance de MFPD lancé sur un millier de plans discrets dont le nombre de points est indiqué en abscisses.

Les morceaux de plans ont été générés aléatoirement de la façon suivante : les caractéristiques a, b, c et μ ont été choisies aléatoirement dans l'intervalle $[-100, 100]$, puis l'épaisseur arithmétique ω dans l'intervalle $[\|(a, b, c)\|_\infty, 20\|(a, b, c)\|_1]$ de manière à avoir un plan discret au minimum aussi épais qu'un plan discret naïf et au maximum 20 fois plus épais qu'un plan standard. Un polygone convexe euclidien a ensuite été généré dans le plan orthogonal à la composante principale de vecteur normal (a, b, c) du plan discret à générer. Puis, les points 3D du plan discret ont été créés en générant le plus grand ensemble de points tel que le projeté orthogonal de chaque point de l'ensemble selon la composante principale du vecteur (a, b, c) appartienne au polygone convexe. Pour fixer les idées, si $0 < a < b < c$ alors un polygone convexe P_{conv} a été généré aléatoirement dans le plan Oxy , puis, pour chaque point $p(x, y) \in P_{conv}$ l'ensemble de points $\left\{ (x, y, z) \in \mathbb{Z}^3 \mid \lfloor -\frac{ax+by-\mu}{c} \rfloor \leq z < \lfloor -\frac{ax+by-\mu-\omega}{c} \rfloor \right\}$ a été créé.

Au vu des graphes obtenus, nous pouvons nous rendre compte qu'en pratique le comportement de

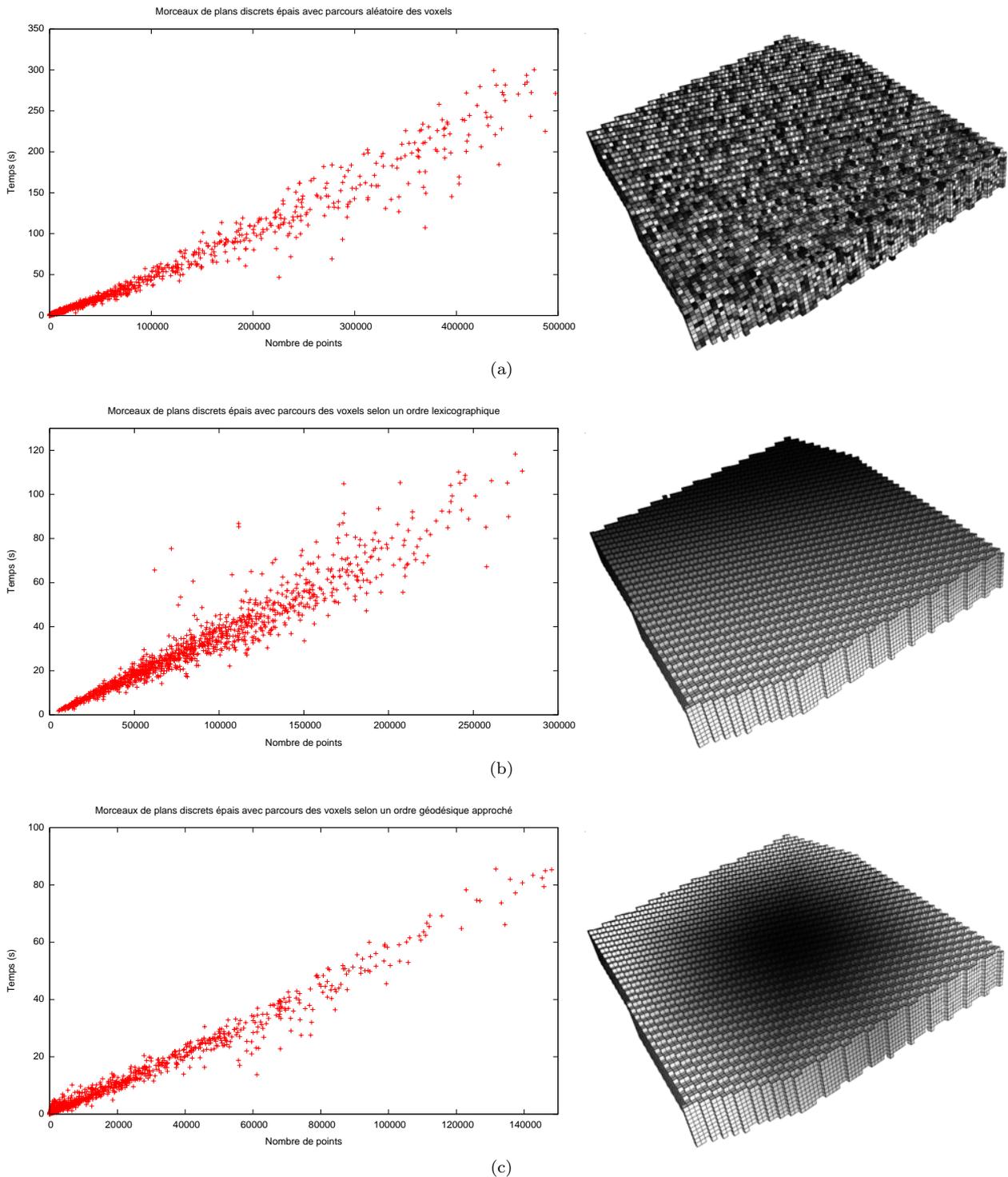


Figure 3.12 – Les résultats de l’algorithme de reconnaissance incrémentale de MFPD (Algorithme 3.2) sur des morceaux de plans discrets. (a) En ajoutant les points de manière aléatoire. (b) En ajoutant les points suivant l’ordre lexicale canonique sur les coordonnées des points. (c) En ajoutant les points suivant un parcours géodésique approché par rapport à un point central. Chaque ligne présente le graphe des temps de calcul en fonction du nombre de points du plan, ainsi qu’un exemple permettant de visualiser le parcours utilisé. Les points sont ajoutés suivant l’échelle en niveau de gris, du plus foncé (ajouté en premier), au plus clair (ajouté en dernier).

l'algorithme de reconnaissance sur des morceaux de plans discrets semble linéaire, et ceci, quel que soit le parcours employé. Les temps de calcul tendent d'ailleurs à montrer que l'ordre d'ajout des points, reflété par le parcours employé, ne semble pas avoir une grande importance dans le cas des morceaux de plans discrets puisqu'ils sont du même ordre de grandeur (autour d'une cinquantaine de secondes pour 100000 points).

Sur des morceaux de plans discrets, l'algorithme se comporte donc beaucoup mieux en pratique que ce que nous pouvions espérer au vu de la complexité théorique en $\mathcal{O}(n^3)$.

3.5.2 Morceaux de sphères discrètes

Les résultats de l'algorithme de reconnaissance de MFPD sur des morceaux de sphères discrètes (et plus particulièrement des huitièmes de sphères) sont représentés à la figure 3.13.

Chaque graphe représente les temps de calcul en secondes (en ordonnées) de l'algorithme de reconnaissance de MFPD lancé sur des huitièmes de sphères discrètes dont le rayon varie par incrément de 1. Le nombre de points est indiqué en abscisses et la courbe polygonale en pointillés permet de mieux appréhender l'évolution des temps de calcul en fonction du rayon des sphères.

Les huitièmes de sphères discrètes ont été générés d'après une définition de sphères discrètes naïves proposée dans [FT06]. Plus précisément, une sphère discrète naïve de centre (a, b, c) et de rayon R est l'ensemble des points :

$$\left\{ (x, y, z) \in \mathbb{Z}^3 \mid -\|(a-x, b-y, c-z)\|_\infty \leq (x-a)^2 + (y-b)^2 + (z-c)^2 - R^2 < \|(a-x, b-y, c-z)\|_\infty \right\}$$

Le huitième de sphère conservé est celui pour lequel $x, y, z \geq 0$.

Cette fois-ci les résultats montrent que l'algorithme ne se comporte plus de manière linéaire. En fait, une régression du nuage de points au modèle $a.x^b$ par une technique des moindres carrés non linéaire (algorithme Levenberg-Marquardt (LMA) [Lev44, Mar63] implémenté dans Gnuplot) nous donne :

- $a = 1,1357.10^{-4}$ (ASE* 67.95%) et $b = 1,49199$ (ASE 4,453%) pour le parcours aléatoire ;
- $a = 1,50529.10^{-5}$ (ASE 23,1%) et $b = 1,85438$ (ASE 1,293%) pour le parcours lexicographique ;
- $a = 9,11642.10^{-5}$ (ASE 127,5%) et $b = 1,87578$ (ASE 7,61%) pour le parcours géodésique.

De plus, le comportement est différent en fonction du parcours employé. Dans le cas du parcours lexicographique, on note en effet une meilleure stabilité des temps de calcul. Dans les cas d'un parcours aléatoire et géodésique, au contraire, on peut observer des grandes fluctuations. Ce qui est remarquable c'est que certains huitièmes de sphères (de rayon R) sont reconnus beaucoup plus rapidement que leurs prédécesseurs (de rayon $R-i, 0 < i < R$) qui comportent pourtant moins de points. La zone encadrée en bleu sur le graphe de la figure 3.13(c) illustre parfaitement cette observation, et les chiffres obtenus en résultat pour cette zone, et résumés dans le tableau 3.1, nous permettent de la quantifier. Par exemple, le huitième de sphère retenu pour $R = 75$ comporte 7503 points et celui pour $R = 76$ est composé de 7690 points, pourtant, dans le cas du parcours géodésique approché, il faut 2245 secondes pour reconnaître le premier – les caractéristiques du MFPD reconnu étant $(a, b, c, \mu, \omega, \nu) = (1, 1, 1, 75, 56, 31.7543)$ – et seulement 1001 secondes (moitié moins de temps) pour le second – les caractéristiques du MFPD reconnu étant $(a, b, c, \mu, \omega, \nu) = (1, 1, 1, 76, 57, 32.3316)$.

La combinatoire des enveloppes convexes des points, illustrées sur la figure 3.14 pour les cas (a) $R = 75$, (b) $R = 76$ et (c) $R = 77$ ne semble pas être la cause *directe* de ce comportement. En effet, dans les cas $R = 75$ et $R = 76$, le nombre de sommets/arêtes/faces est quasiment identique (+6/-9/-15 dans le cas $R = 76$ par rapport au cas $R = 75$, comme le montrent les chiffres du tableau 3.1) et n'explique pas un tel écart de temps. D'autant plus que dans le cas $R = 77$, nous observons des différences sur le nombre de sommets/arêtes/faces de -21/-75/-54 par rapport au cas $R = 76$, donc beaucoup plus conséquentes,

*. ASE : Asymptotique Standard Error

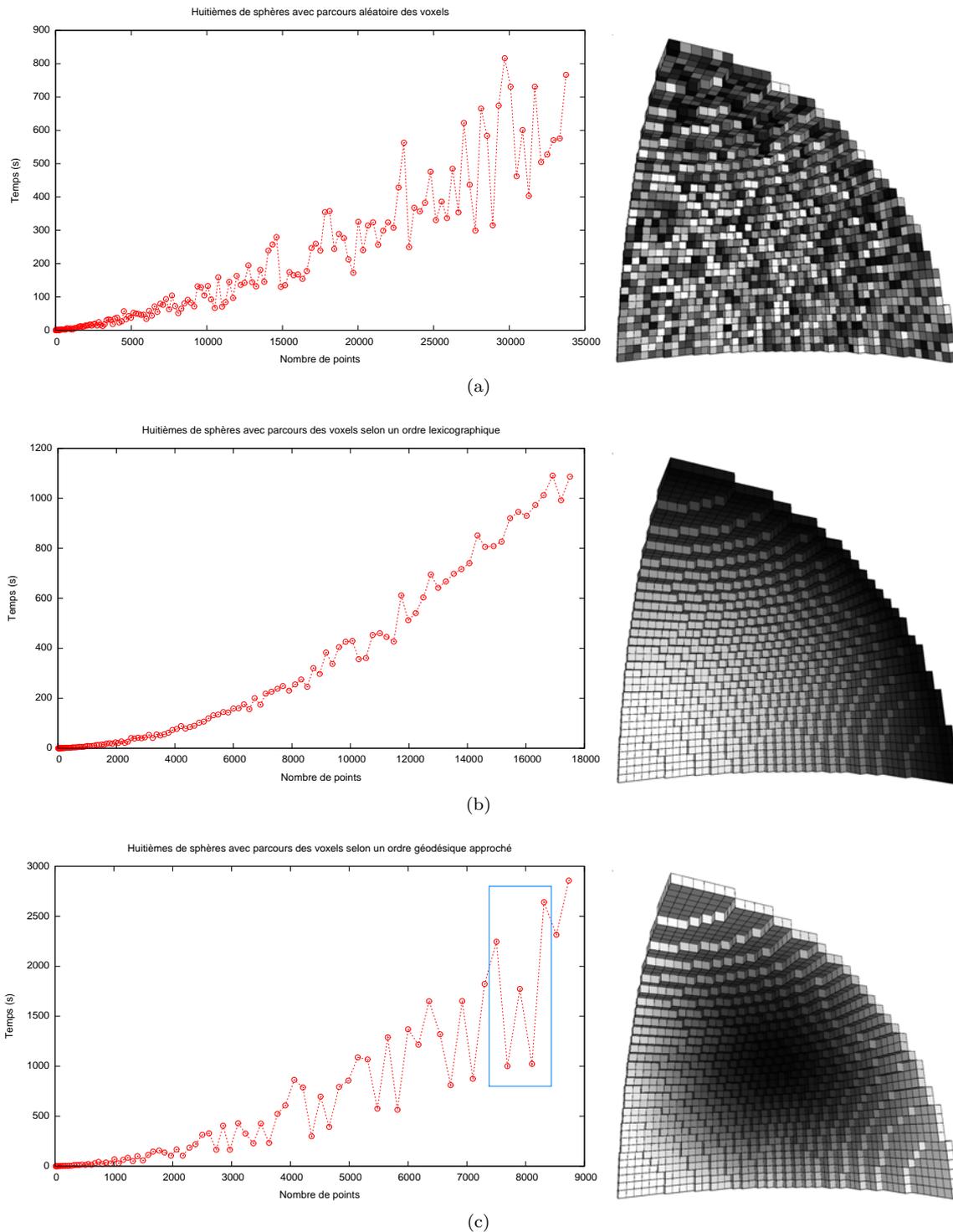


Figure 3.13 – Les résultats de l’algorithme de reconnaissance incrémentale de MFPD (Algorithme 3.2) sur des huitièmes de sphères discrètes. (a) En ajoutant les points de manière aléatoire. (b) En ajoutant les points suivant l’ordre lexicale canonique sur les coordonnées des points. (c) En ajoutant les points suivant un parcours géodésique approché par rapport à un point central. Chaque ligne présente le graphe des temps de calcul en fonction du nombre de points du huitième de sphère, ainsi qu’un exemple permettant de visualiser le parcours utilisé. Les points sont ajoutés suivant l’échelle en niveau de gris, du plus foncé (ajouté en premier), au plus clair (ajouté en dernier).

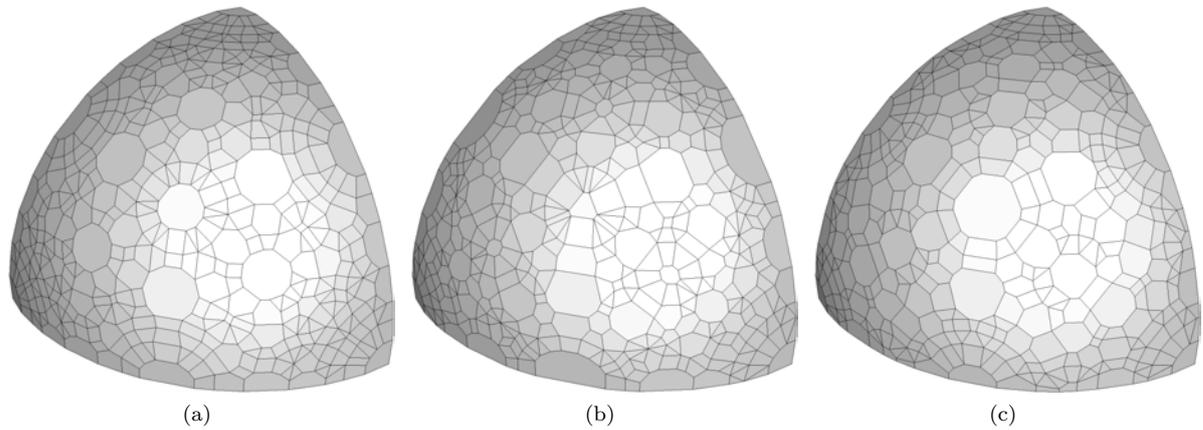


Figure 3.14 – Les enveloppes convexes des huitièmes de sphères discrètes pour un rayon (a) $R = 75$, (b) $R = 76$ et (c) $R = 77$.

1/8 ^e de sphères		Enveloppe convexe			Temps (s) de reconnaissance	Recalculs hauteur	Nombres de tests de	
Rayon	Points	Sommets	Arêtes	Faces			paires SF ^a	paires AA ^b
75	7503	540	1059	521	2245	422	1299224	3768020
76	7690	546	1050	506	1001	196	574996	1622271
77	7902	525	975	452	1773	386	1143610	3207578
78	8107	552	1053	503	1023	195	562546	1636168
79	8313	564	1116	554	2642	471	1459254	4218279

Tableau 3.1 – Quelques résultats obtenus sur les huitièmes de sphères discrètes correspondant à la zone de fluctuations encadrée en bleu sur le graphe 3.13(c) pour le parcours géodésique approché.

a. paire sommet-face antipodale

b. paire arête-arête antipodale

bien que le temps de reconnaissance soit cette fois-ci plus élevé d'un facteur d'à peu près 1,5.

Par contre, en traçant les appels de la fonction `CalculPlanSupportMin(C)` de l'algorithme 3.2, nous pouvons voir que le temps de reconnaissance est plus directement lié au nombre de fois où la largeur de l'enveloppe convexe des points doit être recalculée. Même si en théorie cela ne change pas la complexité en temps au pire cas de l'algorithme, on s'aperçoit bien, ici, de tout l'intérêt de n'effectuer en pratique un recalcul de la hauteur du segment flou que lorsque le reste du point ajouté par rapport au MFPD déjà reconnu n'est pas compatible avec ce dernier.

3.5.3 Morceaux de surfaces discrètes

Les résultats de l'algorithme de reconnaissance de MFPD sur des morceaux de surfaces discrètes sont représentés à la figure 3.16.

Chaque graphe représente les temps de calcul en secondes (en ordonnées) de l'algorithme de reconnaissance de MFPD lancé sur 600 morceaux de surfaces discrètes fonctionnels dont le nombre de points est indiqué en abscisses. Nous avons choisi ce type d'objet discret car il représente assez bien le type de surface que l'on peut rencontrer lorsqu'on travaille avec des points du bord d'un objet discret non régulier.

Les morceaux de surfaces discrètes ont été générés aléatoirement de la façon suivante : tout d'abord un morceau P de plan discret épais est généré de la manière décrite à la section 3.5.1. Ensuite, ce morceau de plan est rendu fonctionnel, dans le plan orthogonal à sa direction principale, en utilisant la liste L des tricubes appartenant au même 48^e d'espace que le morceau de plan. Un voxel de départ v est choisi aléatoirement dans P , puis un tricube t , centré en v , et dont tous les voxels appartiennent à P est recherché dans L . Un fois trouvé, tous les voxels de P se trouvant au-dessus et au-dessous de t , relativement à la direction principale de P , sont supprimés. Le voxel central de t est marqué comme traité et ses voisins sont insérés dans une liste V représentant les voxels à visiter. Cette procédure est répétée en recherchant de nouveaux tricubes centrés en l'un des voxels de V , non traité. Un affinage de P est ainsi effectué, localement, grâce aux tricubes, pour arriver à des morceaux de surfaces discrètes comme ceux illustrés à la figure 3.15. L'algorithme d'affinage est décrit plus en détail dans mon mémoire de DEA [Pro05].

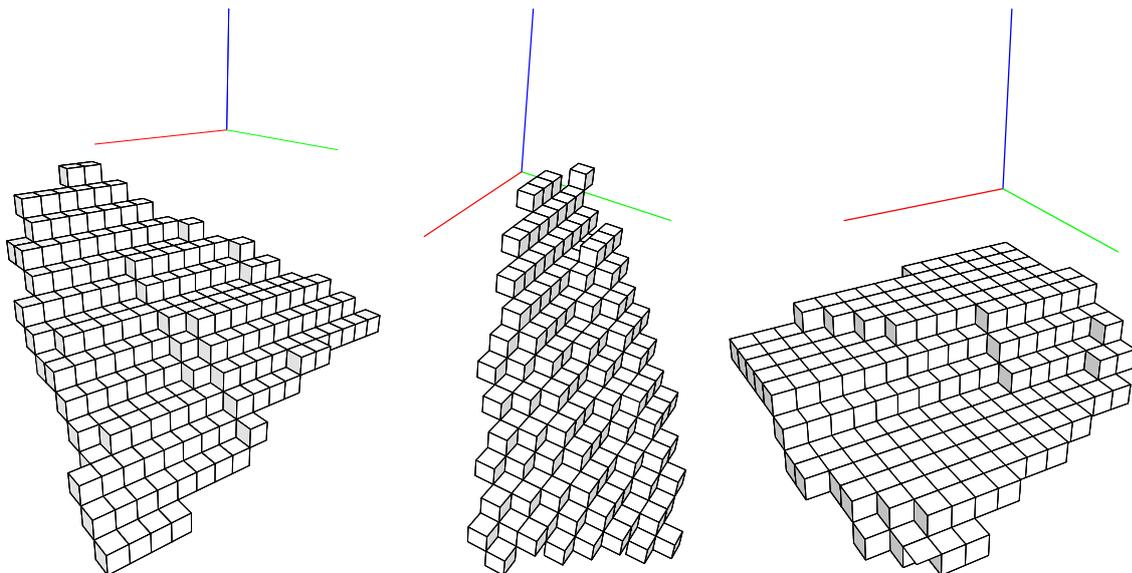


Figure 3.15 – Exemples de morceaux de surfaces discrètes, générés pour tester le comportement de l'algorithme incrémental de reconnaissance de MFPD.

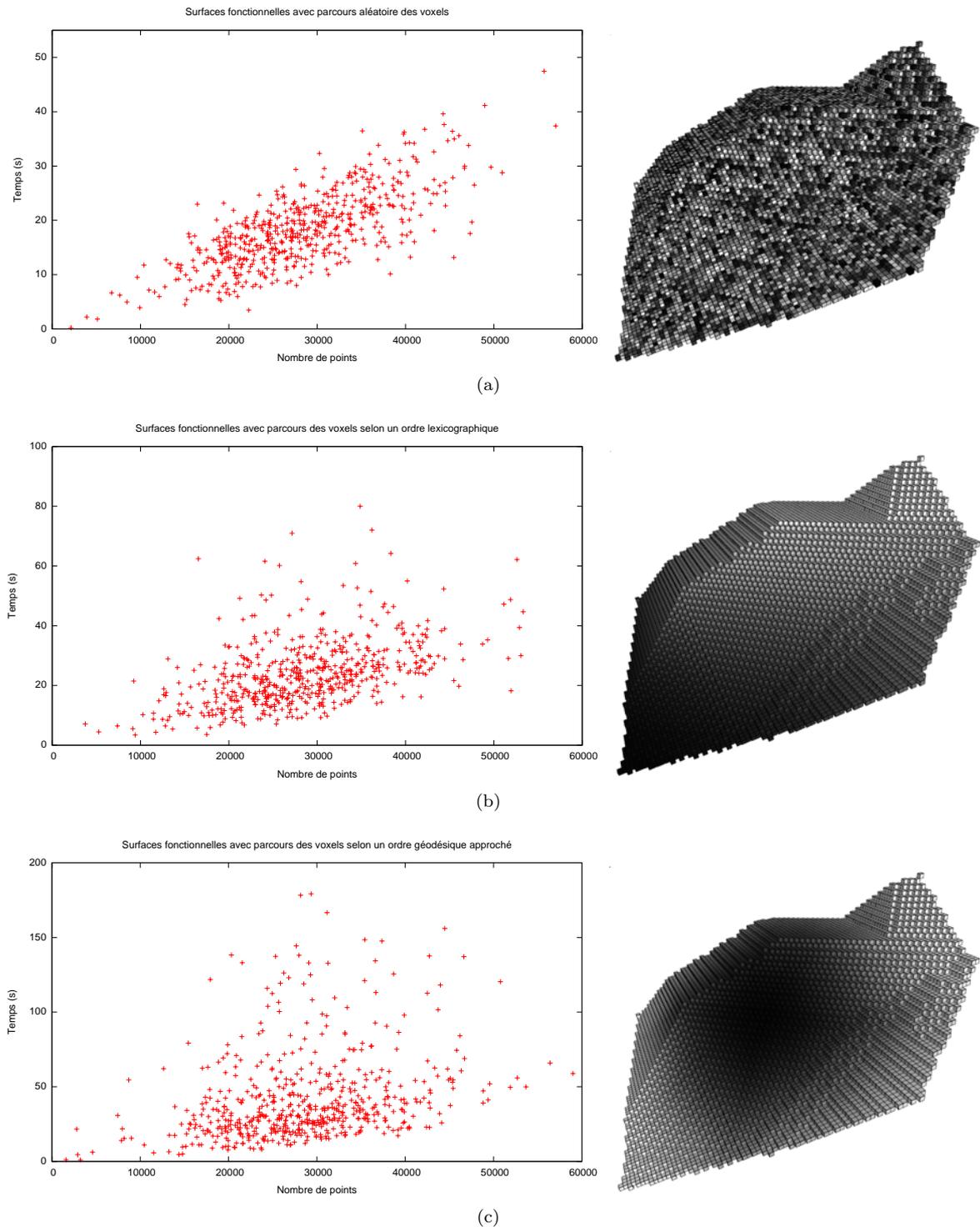


Figure 3.16 – Les résultats de l’algorithme de reconnaissance incrémentale de MFPD (Algorithme 3.2) sur des morceaux de surfaces discrètes fonctionnelles par rapport à leur axe principal. (a) En ajoutant les points de manière aléatoire. (b) En ajoutant les points suivant l’ordre lexicale canonique sur les coordonnées des points. (c) En ajoutant les points suivant un parcours géodésique approché par rapport à un point central. Chaque ligne présente le graphe des temps de calcul en fonction du nombre de points de la surface, ainsi qu’un exemple permettant de visualiser le parcours utilisé. Les points sont ajoutés suivant l’échelle en niveau de gris, du plus foncé (ajouté en premier), au plus clair (ajouté en dernier).

Si nous nous intéressons à présent aux graphes de l'évolution des temps de calculs obtenus, nous nous apercevons que le nuage de points est plutôt épars. Même s'il semble alors difficile de faire des conjectures sur le comportement de l'algorithme avec de tels données, nous pouvons néanmoins remarquer que les temps de calculs restent assez proches de ceux obtenus dans le cas d'une reconnaissance sur des morceaux de plans discrets épais. En tout cas ils restent bien inférieurs à ceux obtenus avec des morceaux de sphères.

3.5.4 Synthèse et discussion

Après analyse des résultats sur différents ensembles de points discrets, une première constatation est que la borne de complexité théorique au pire cas, de $\mathcal{O}(n^3)$ où n représente le nombre de points de l'ensemble, ne reflète pas vraiment le comportement de l'algorithme en pratique. Il semblerait donc que les pires cas théoriques soient très rarement, voire jamais, atteints quand on manipule des ensembles de points correspondants au points du bord d'un objet discret.

Ceci était quelque peu prévisible puisque ces pires cas sont atteints lorsque les n points sont disposés de manière particulière, sur les grands arcs d'une sphère, formant ainsi un réseau où le nombre de paires arête-arête antipodales est de l'ordre de $\Theta(n^2)$.

Néanmoins, même si ces cas ne se présentent pas en pratique, nous avons pu remarquer qu'un ensemble de points correspondant au bord d'une sphère discrète reste une configuration coûteuse pour l'algorithme de reconnaissance de MFPD. Une question naturelle est donc de savoir si des améliorations sont envisageables afin de réduire les temps de reconnaissance. En fait, nous avons observé que l'opération de calcul de la largeur de l'enveloppe convexe des points est très coûteuse, dû notamment aux nombreux tests de détections de paires antipodales (les deux dernières colonnes du tableau 3.1). Une idée serait donc de chercher à réduire le nombre de ces tests.

À travers les nombreux tests de reconnaissance de morceaux flous de plans discrets que j'avais effectués lors de mon DEA, il m'avait semblé constater un fait intéressant (déjà observé en 2D avec les segments flous et qui tendait à se confirmer en 3D) : chaque point ajouté qui modifie les caractéristiques du plan englobant optimal, et entraîne un recalcul de la hauteur de ce dernier, semble devenir un point d'appui du nouveau plan englobant optimal, et donc, par conséquence, aussi un point appartenant à une paire antipodale définissant les deux plans support parallèles entre lesquels la distance est minimale. Ainsi, pour mettre à jour la hauteur du MFPD, il suffirait de s'intéresser aux paires antipodales contenant le point ajouté.

Malheureusement, un contre-exemple à cette observation a été trouvé. Il mettait en évidence le fait qu'une paire antipodale, P_a , définissant des plans support parallèles entre lesquels la distance d est minimale pour la nouvelle enveloppe convexe, était déjà présente dans l'ancienne enveloppe convexe (mais la distance entre les plans support qu'elle définissait n'était alors pas minimale). L'ajout d'un nouveau point entraînait des modifications qui ne supprimaient pas P_a mais créaient de nouvelles paires antipodales pour lesquelles la distance entre les plans qu'elles définissaient était supérieure à d .

Une première tentative fut donc d'essayer de conserver l'ensemble des paires antipodales de l'enveloppe convexe des points et de le maintenir à jour incrémentalement, en même temps que l'enveloppe convexe. Le problème est que la structure de données retenue, un arbre rouge et noir (car les opérations **Suppression**, **Insertion**, **Recherche** et **Minimum** qui nous étaient nécessaires s'exécutaient en $\mathcal{O}(\log n)$ au pire cas, où n représente le nombre de nœuds dans l'arbre, c'est-à-dire le nombre de paires antipodales) était en pratique plus coûteuse à maintenir à jour que de recalculer l'ensemble des paires antipodales.

Ensuite, en analysant le contre-exemple, nous avons aussi remarqué qu'avec le même ensemble de points, mais en considérant cette fois-ci une distance issue de la norme infinie $\|\cdot\|_\infty$, le cas de figure ne se présentait plus. En fait, en utilisant des arguments similaires à ceux utilisés en 2D, nous avons montré que si l'on considère la distance issue de la norme infinie, alors un point ajouté au cours de l'algorithme dont le reste n'est pas compatible avec le MFPD courant va entraîner un recalcul de la hauteur du nouveau MFPD, et une des composante de la paire antipodale définissant les plans support parallèles entre lesquels la distance est minimale contient le nouveau point ajouté. Il n'est alors plus nécessaire d'effectuer une mise à jour de la hauteur en effectuant des rotations autour de toutes les arêtes de l'enveloppe convexe, mais

il suffit de s'intéresser aux arêtes issues du nouveau point ajouté. On réduit ainsi de manière conséquente le nombre de tests de détection de paires antipodales, d'autant plus qu'en général le nombre de nouvelles arêtes de l'enveloppe convexe créées lors de l'ajout d'un point n'est pas très élevé (il correspond en fait au nombre de sommets appartenant à l'horizon).

Enfin, toujours en considérant une distance issue de la norme infinie, une technique similaire à celle décrite en 2D pour la *reconnaissance de segments flous par une approche géométrique* (voir page 49) pourrait être envisagée. L'idée serait de séparer l'enveloppe convexe en une partie supérieure et une partie inférieure et d'utiliser à la place des points L_L , U_F et U_L , les sommets de l'horizon pour définir des points de départ pour la recherche de la nouvelle paire antipodale minimale. L'argument de concavité de la fonction $\mathcal{V}(i)$ resterait valable en 3D, permettant d'orienter la recherche sans revenir « en arrière », mais les notions relatives à l'organisation des points sur l'enveloppe convexe sont beaucoup plus complexes en 3D. Il est notamment difficile de définir un ordre naturel sur les points et des notions qui sont intuitives en 2D comme « en avant » ou « en arrière » sont beaucoup moins évidentes en 3D. De plus les objets à parcourir pour rechercher une composante d'une paire antipodale sont plus complexes, puisque nous devons manipuler des morceaux de maillages en 3D en lieu et place de simples lignes polygonales en 2D.

Cette dernière approche n'a pas été concrétisée, mais mérite d'après moi une étude plus poussée. Elle est d'ailleurs en priorité sur ma liste de perspectives. Je suis en effet convaincu que les problèmes soulevés sont surmontables et que l'algorithme obtenu pourrait être en pratique beaucoup plus efficace que l'algorithme 3.2.

3.6 Conclusion

Nous avons présenté dans ce chapitre une nouvelle primitive discrète, le *morceau flou de plan discret d'épaisseur ν* , généralisant à la dimension supérieure la notion de segment flou de droite discrète, introduite pour amener plus de flexibilité dans l'étude d'objets discrets possiblement bruités.

Nous avons proposé deux algorithmes de reconnaissance de MFPD_ν , l'un utilisant des techniques de programmation linéaire et permettant de résoudre le problème en considérant la norme infinie $\|\cdot\|_\infty$, et l'autre, exploitant directement la géométrie de l'enveloppe convexe des points et permettant de résoudre le problème du point de vue de la norme euclidienne $\|\cdot\|_2$ en énumérant l'ensemble des plans euclidiens parallèles définis par les paires sommet-face et arête-arête antipodales de l'enveloppe convexe et en conservant celles pour lesquelles la distance entre les plans est minimale.

Nous avons aussi montré comment modifier certaines techniques de reconnaissance de morceaux de plans discrets déjà existantes afin qu'elles puissent être réutilisées pour reconnaître des MFPD_ν . Il serait alors intéressant d'implémenter ces modifications afin de comparer les méthodes entre-elles.

Enfin, nous avons présenté une étude expérimentale de l'algorithme de reconnaissance de MFPD_ν suivant l'approche géométrique et nous avons proposé quelques pistes pour une amélioration possible des temps d'exécution en pratique.

Dans les chapitres suivants, nous allons voir comment utiliser ces morceaux flous de plans discrets afin d'extraire certaines caractéristiques géométriques du bord d'objets discrets, ainsi que leur utilisation dans un contexte de segmentation du bord d'un objet discret en morceaux de plans discrets.

Chapitre 4

Estimateurs de caractéristiques géométriques

Sommaire

4.1	Introduction	75
4.2	Estimateurs de vecteur normal	77
4.2.1	Approches par tranches	77
4.2.2	Approche multi-directionnelles	83
4.2.3	Résultats	88
4.3	Estimateur de forme	93
4.3.1	Notion de courbure en géométrie différentielle	96
4.3.2	Estimation de la courbure sur des surfaces discrètes	97
4.3.3	Approche par tranches	101
4.3.4	Approche par patches discrets d'épaisseur ν	102
4.4	Conclusion	107

4.1 Introduction

De manière générale, l'estimation de caractéristiques géométriques à la surface d'un objet apporte des informations très utiles aux applications qui manipulent cet objet. Par exemple, dans les domaines de la synthèse d'images ou de la vision par ordinateur, les algorithmes de rendu d'image utilisent souvent une estimation du champ de vecteurs normaux à la surface de l'objet pour produire une vue réaliste, avec un éclairage cohérent, de l'objet. En analyse d'image, des caractéristiques comme l'aire ou la courbure peuvent être utilisées pour mettre au point des fonctions de classification (des *signatures*) permettant de trier les objets par taille ou par forme, ou encore de comparer ces objets entre eux. En bio-informatique, dans le contexte de l'appariement de protéines (protein-protein docking), une partie du problème consiste à associer deux protéines par complémentarité géométrique. Il est donc primordial de localiser des zones critiques à la surface de ces protéines, comme les creux, les bosses ou encore les grandes zones plates.

Dans le cas plus particulier du problème de la polyédrisation d'objets discrets, nous cherchons à transformer la représentation d'un objet sous forme de nuage de points en une représentation sous forme de maillage polyédrique. Pour que cette transformation ait un sens, il faut évidemment veiller à conserver au mieux la géométrie de l'objet discret initial. Il semble alors naturel d'utiliser certaines caractéristiques géométriques du bord de l'objet pour diriger cette transformation.

Dans cette thèse, nous avons choisi d'effectuer une polyédrisation reposant sur une segmentation préalable du bord de l'objet discret considéré en morceaux flous de plans discrets. L'idée est de découper le

bord de l'objet discret en MFPD en donnant une priorité plus élevée aux zones estimées « plates », laissant ainsi la possibilité aux morceaux de s'étendre plus. En outre, comme nous le verrons dans le chapitre suivant, le processus d'extension incrémental des faces exploitera aussi ces caractéristiques afin de choisir judicieusement le point à ajouter.

La caractéristique géométrique généralement utilisée pour discriminer ce type de zone est la courbure. Par exemple, l'approximation polyédrique par squelettisation (section 2.2.2.1 page 26) et la méthode Delaunay Discret (section 2.2.2.2 page 29) utilisent la courbure moyenne à la surface de l'objet discret afin de déterminer la position des graines. L'idée est de placer plus de graines au niveau des zones fortement courbées, afin de raffiner le maillage résultant à ces endroits, pour éviter de s'éloigner trop de l'objet. Ensuite ce sont principalement des critères topologiques qui sont utilisés pour développer les faces.

Dans les méthodes de type *polyédrisation réversible*, basées sur une segmentation du bord de l'objet en morceaux de plans discrets (section 2.3 page 32) la géométrie de l'objet est surtout considérée, de manière implicite, lors de la vérification d'appartenance des points du bord de l'objet discret à un plan discret arithmétique ; c'est-à-dire lors du développement des faces. Dans ce cas, comme le notent les auteurs, le choix de la graine influe beaucoup sur la polyédrisation (ou la segmentation) obtenue. L'idée est alors de placer des graines dans les endroits les plus « plats » du bord de l'objet, laissant ainsi la possibilité aux morceaux de plans discrets de s'étendre plus. Malheureusement, une méthode pratique permettant de caractériser ces zones n'est pas systématiquement proposée (bien que la notion de courbure soit suggérée, aucun résultat n'est proposé), et le plus souvent le choix d'une nouvelle graine est effectué en prenant comme candidat le dernier point n'ayant pu être ajouté à la dernière face en cours de reconnaissance. Néanmoins, sans toutefois faire appel à la notion de courbure, la méthode de *polyédrisation en polyèdre discret standard* (section 2.3.1) répond à ce problème grâce au concept de *palier* (voir page 34), en considérant le champ de vecteurs normaux au bord de l'objet.

De nombreuses études [YCK92, TW97, LMR96, Len97, PF98, TDR99, HK03] ont déjà été menées sur des estimateurs de vecteur normal et de courbure. Mais comme par définition ces caractéristiques sont très locales, les estimateurs proposés peuvent être très sensibles aux perturbations des points de l'objet (dues par exemple au bruit introduit par des appareils d'acquisition comme le scanner ou l'IRM) et ne pas fournir de résultats exploitables. Puisque nous souhaitons justement manipuler des objets discrets possiblement bruités, nous nous sommes naturellement intéressés à ces problèmes et, en nous inspirant de certains travaux précédemment cités, nous avons mis au point un estimateur de vecteur normal en un point du bord d'un objet discret ainsi qu'un estimateur de forme permettant de distinguer les parties *plates*, les *creux* et les *bosses* sur le bord d'un objet discret. Ces estimateurs reposent sur des primitives floues et sont donc robustes à des perturbations du bord de l'objet.

La suite de ce chapitre est dédiée à leur présentation. Nous décrivons deux approches, l'une exploitant les tranches de l'objet dans les trois directions canoniques de l'espace discret, et l'autre exploitant un voisinage planaire adaptatif autour d'un point du bord, relativement à la définition des MFPD. Nous considérons ici des objets discrets 6-connexes, possiblement bruités, et nous appelons *bord* de cet objet l'ensemble des points de l'objet ayant un 6-voisin n'appartenant pas à l'objet. Nous appellerons *surface* de cet objet discret l'ensemble des surfels du bord de l'objet « visibles » depuis l'extérieur, c'est-à-dire l'ensemble des surfels s de l'objet tel que $s \in v$ et $s \in v'$ où v est un voxel appartenant à l'objet et v' un voxel n'appartenant pas à l'objet. De plus, nous pouvons supposer que nous travaillons sur une surface qui est une 2-variété combinatoire, sachant qu'il est toujours possible de s'y ramener, quitte à considérer la topologie de l'objet modulo les quelques transformations locales proposées dans [Bol07], et correspondant à celles proposées dans [Nie04] dans le cadre d'une étude sur une surface duale de celle issue du Marching Cubes. Attention, il faudra dans ce cas utiliser des structures permettant de dissocier la géométrie et la topologie, par exemple graphes ou G-cartes.

4.2 Estimateurs de vecteur normal

4.2.1 Approches par tranches

Dans le but d'estimer le vecteur normal en un point p du bord d'un objet discret tri-dimensionnel, certains auteurs ont eu l'idée de réduire le problème à un ensemble de sous-problèmes 2D en considérant les tranches de l'objet le long des trois plans canoniques de l'espace passant par p .

4.2.1.1 Méthodes existantes utilisant une approche par tranches

De manière générale, pour un objet défini sur un espace discret de dimension n , on appelle *tranche d'indice i au point p* , avec $p(p_1, \dots, p_n)$ et $1 \leq i \leq n$, l'ensemble des points de l'objet ayant la même i^e coordonnée que p , c'est-à-dire l'ensemble des points $q(q_1, \dots, q_n)$ appartenant à l'objet tels que $q_i = p_i$. Dans la suite nous considérons des objets discrets tri-dimensionnels, donc nous pouvons voir une tranche de l'objet comme un objet en deux dimensions. Le *contour* associé à une tranche au point p de l'objet est composé des voxels (resp. surfels) de la courbe fermée appartenant au bord (resp. à la surface) de l'objet et passant par p . Ces notions sont illustrées à la figure 4.1.

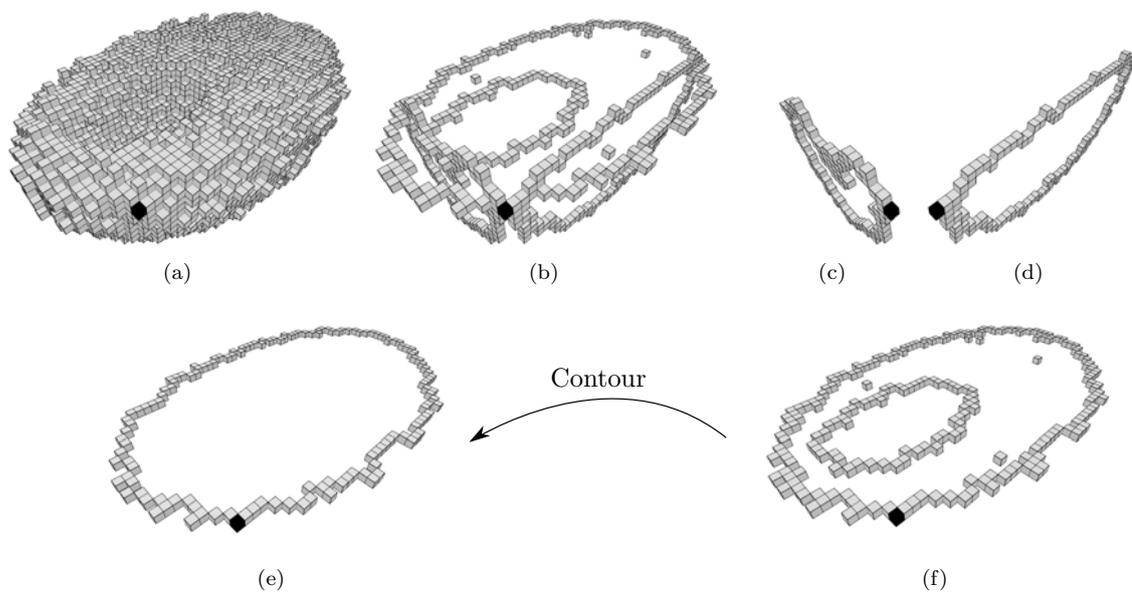


Figure 4.1 – (a) Un objet discret et (b) ses tranches selon les trois plans de coupe canoniques passant par le point noir ; (c),(d) et (f) les tranches isolées de l'objet ; (e) le contour associé à la tranche (f).

A. Lenoir et ses coauteurs [LMR96, Len99] ont été les premiers à proposer un estimateur de vecteur normal reposant sur cette approche. En considérant la surface en surfels d'un objet discret, nous pouvons extraire exactement deux tranches associées à chaque surfel s de la surface de l'objet. Les auteurs proposent alors d'estimer les vecteurs tangents \vec{t}_1 et \vec{t}_2 en s dans chacune des deux courbes définissant le contour de chaque tranche en s et de définir le vecteur normal en s comme étant le produit vectoriel de \vec{t}_1 et \vec{t}_2 . L'estimation des tangentes est effectuée en convoluant de manière récursive des fonctions sur les courbes de tranches (décrivant la géométrie de la surface au voisinage du surfel) avec un filtre passe bas, en l'occurrence un noyau de convolution gaussien associé à un paramètre d'échelle σ permettant de contrôler la taille du noyau de convolution c'est-à-dire la taille du voisinage de surfel considéré autour de s . Grâce au paramètre d'échelle cette technique peut-être robuste à des perturbations de la surface de l'objet, néanmoins le voisinage considéré n'est pas adaptatif à la surface, c'est-à-dire, qu'une fois σ fixé, le voisinage considéré est toujours du même ordre de grandeur. De plus, il n'est pas facile d'adapter cette

méthode en utilisant des primitives floues.

C'est pourquoi nous nous sommes intéressés à une autre méthode, proposée par P. Tellier et I. Debled-Rennesson [TDR99], qui considère aussi une approche utilisant les tranches de l'objet, mais où cette fois-ci, l'estimation des tangentes est effectuée grâce à un algorithme de reconnaissance de segments de droites discrètes [DRR95, DR95]. L'estimateur proposé est utilisable aussi bien pour apprécier le champ de vecteurs normaux de la surface en surfels ou du bord en voxels de l'objet discret, en appliquant l'algorithme de reconnaissance de segments de droites discrètes sur, respectivement, les pointels ou les pixels du contour 2D des tranches de l'objet. Plus précisément, la direction de la tangente en un point p est calculée à partir des caractéristiques d'un segment de droite discrète arithmétique centré en p et reconnu incrémentalement, par l'algorithme proposé dans [DRR95], à partir des points du voisinage de p . Ensuite, pour une estimation de la normale en un surfel s de la surface, le résultat est obtenu comme dans la méthode précédente, en calculant le produit vectoriel des directions des deux tangentes obtenues à partir des contours des deux tranches de l'objet issues de s . Dans le cas d'une estimation de la normale en un voxel v du bord de l'objet, les trois tranches de l'objet peuvent fournir d'une à trois directions de tangentes exploitables (c'est-à-dire non colinéaires). Dans le cas de deux directions de tangentes, la normale 3D en v est obtenue comme précédemment, et dans le cas de trois directions de tangentes, elle résulte de la moyenne des trois produits vectoriels des trois directions des tangentes (considérées deux à deux). Dans le cas où une seule direction de tangente est disponible, la normale en v est le vecteur orthogonal à cette direction dans le plan contenant la tranche où elle a été estimée.

4.2.1.2 Estimateur par tranches de normale à l'épaisseur ν

Nous proposons ici une extension de la méthode proposée dans [TDR99] en utilisant en lieu et place de l'algorithme de reconnaissance de segments de droites discrètes, un algorithme de reconnaissance de segments flous. De plus, l'estimation de la direction des tangentes en un point p ne sera pas faite à partir d'un seul segment flou centré en p , mais à partir de l'ensemble des segments flous maximaux contenant p . L'utilisation des segments maximaux, dont la définition a été introduite dans [FT99] pour le cas des segments discrets réguliers et étendue dans [NDR07] pour le cas des segments flous, permet d'avoir une compréhension globale de la géométrie de la courbe discrète considérée. Nous rappelons ci-dessous la définition et quelques propriétés des segments flous avant de présenter l'estimateur par tranches de vecteur normal à l'épaisseur ν , en un point du bord d'un objet discret possiblement bruité.

Considérons une courbe discrète C , dont les points sont indexés de 0 à $n - 1$. Nous notons $C_{i,j}$ l'ensemble des points consécutifs de C indicés par les valeurs croissantes des index de i à j . Le prédicat $C_{i,j}$ est un segment flou d'épaisseur ν sera noté $BS_\nu(C, i, j)$.

Définition 4.1 $C_{i,j}$ est appelé un *segment flou maximal d'épaisseur ν* et noté $MBS_\nu(C, i, j)$ si et seulement si $BS_\nu(C, i, j)$ et $\neg BS_\nu(C, i, j + 1)$ et $\neg BS_\nu(C, i - 1, j)$.

Proposition 4.2 ([NDR07]) Soit C une courbe discrète et $MBS_\nu(C)$ la séquence des m segments flous maximaux d'épaisseur ν de C . Alors $MBS_\nu(C) = \{MBS_\nu(C, B_1, E_1), MBS_\nu(C, B_2, E_2), \dots, MBS_\nu(C, B_m, E_m)\}$ et satisfait les inégalités $B_1 < B_2 < \dots < B_m$. Aussi, nous avons $E_1 < E_2 < \dots < E_m$.

Cette proposition a permis de mettre au point un algorithme de calcul de l'ensemble des segments flous maximaux (appelé *couverture tangentielle*) d'une courbe discrète en temps $\mathcal{O}(n \log^2 n)$ au pire cas, où n représente le nombre de points de la courbe. Pour une épaisseur ν donnée, l'idée est de calculer un premier segment flou maximal d'épaisseur ν , puis de le faire « glisser » le long de la courbe C en maintenant la propriété d'être un segment flou maximal d'épaisseur ν . Pour cela, nous construisons un segment flou S en partant du premier point de la courbe et nous ajoutons incrémentalement à S les successeurs de ce point appartenant à C jusqu'à ce que $\neg BS_\nu(S, 0, j + 1), 0 < j < n$. Nous avons alors notre premier segment flou maximal $MBS_\nu(S, 0, j)$. Comme $\neg BS_\nu(S, 0, j + 1)$, nous revalidons la contrainte d'être un segment flou en retirant de S les premiers points jusqu'à ce que $BS_\nu(S, i, j + 1), 0 \leq i < j$.

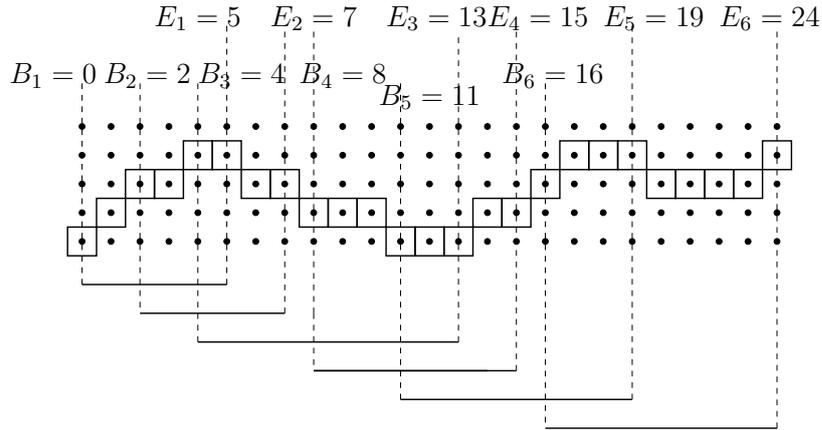


Figure 4.2 – Décomposition d'une courbe discrète en segments flous maximaux d'épaisseur 1.

La proposition 4.2 nous affirme alors que le prochain segment flu maximal est obtenu en ré-étendant S de façon à ce que $MBS_\nu(S, i, k)$ pour $(j + 1 \leq k < n)$, obtenant ainsi le deuxième segment flu. Cette procédure est alors répétée jusqu'au dernier point de la courbe, c'est à dire jusqu'à l'obtention du premier $MBS_\nu(S, l, n - 1)$, $0 \leq l < n$. Dans le cas d'une courbe fermée, les indices peuvent être considérés modulo n . Le premier segment flu sera alors étendu à droite *et* à gauche, et nous nous arrêterons lorsque le dernier segment flu maximal calculé sera identique au premier.

En ce qui concerne la reconnaissance de segments flous, l'algorithme présenté à la section 3.2 page 49 pourrait être utilisé. Néanmoins, cela serait très coûteux car cet algorithme n'est pas décremental, c'est-à-dire qu'il n'autorise pas le retrait d'un point de l'enveloppe convexe. Pour atteindre la complexité au pire cas annoncée auparavant, les auteurs utilisent un algorithme de maintenance dynamique d'enveloppe convexe, proposé dans [OvL81], où l'ajout et le retrait d'un point se font en $\mathcal{O}(\log^2 n)$ au pire cas pour un ensemble de n points, associé à une méthode de recherche binaire double, proposée dans [Buz05], pour déterminer en $\mathcal{O}(\log^2 n)$ la hauteur et la largeur de l'enveloppe convexe. De plus, grâce à la proposition 4.2, l'ensemble des segments flous maximaux de la courbe C est obtenu en ne passant par chaque point de la courbe qu'au maximum deux fois, d'où la complexité annoncée de $\mathcal{O}(n \log^2 n)$ au pire cas. Il est à noter qu'un algorithme de complexité moindre, $\mathcal{O}(n \log n)$, a été proposé dans [FF08, FBF09] pour calculer cette même couverture tangentielle, mais aucune implémentation n'a encore été fournie.

Nous pouvons à présent introduire notre estimateur par tranches de vecteur normal à l'épaisseur ν , en un point p du bord d'un objet discret. La méthode reste utilisable dans les cas d'une estimation du champ de vecteurs normaux à la surface en surfels d'un objet discret. Néanmoins, nous la présentons ici en considérant le bord en voxel de l'objet, afin de la comparer avec notre estimateur de normales d'épaisseur ν présenté dans la section 4.2.2 suivante. L'algorithme 4.1 résume la méthode, qui dans l'ensemble, reste similaire à celle décrite [TDR99]. La description des fonctions utilisées dans l'algorithme met en avant les principales différences.

1. La fonction `Contour`(i, v, \mathcal{O}_b) à la ligne 5 extrait la courbe discrète représentant le contour C de l'objet discret \mathcal{O}_b dans la tranche i relative au voxel v . Ceci peut être aisément calculé grâce à un algorithme de suivi de contour (par exemple [Udu94]).
2. Si ce contour est exploitable, c'est-à-dire s'il contient plus d'un point, alors :
 - La fonction `CalculerMBS`(ν, p) à la ligne 7 va calculer l'ensemble des segments flous maximaux d'épaisseur ν de C qui passent par le point p . Ceci est fait en utilisant la méthode [NDR07]. Néanmoins une simplification peut-être envisagée pour ne pas calculer les segments maximaux de *toute* la courbe, mais seulement ceux contenant p . Supposons que la courbe C contiennent n points indicés à partir de p de 0 à $n - 1$. Comme la courbe est fermée, nous pouvons considérer les indices modulo n . Nous commençons par reconnaître un segment flu S s'étendant le plus possible

Algorithme 4.1 : Estimateur de vecteur normal relativement à une épaisseur ν

Données : Un objet discret tri-dimensionnel 6-connexe \mathcal{O}_b , un point $p \begin{pmatrix} p_1 \\ p_2 \\ p_3 \end{pmatrix} \in \mathcal{B}_b$, le bord de \mathcal{O}_b ,
 et un réel ν

Résultat : Une estimation du vecteur normal au point p de \mathcal{B}_b

```

1  début
2  VT ← { (0), (0), (0) }
3  NbTangentes ← 0
4  pour i ← 1 à 3 faire
5  C ← Contour(i, p, O_b)
6  si C est exploitable alors
7  MBS ← CalculerMBS(ν, p)
8  T ← (0)
9  pour tous les segments flous S ∈ MBS faire
10 (a, b, μ, ω) ← Caractéristiques(S)
11 T ← T + (b)
12 V ← Extension3D(T, i)
13 NbTangentes ← NbTangentes + 1
14 VT[NbTangentes] ← V
15 suivant le nombre N de vecteurs non colinéaires dans VT faire
16 cas où N = 1
17 [ retourner le vecteur perpendiculaire à VT[1] normalisé
18 cas où N = 2
19 [ retourner VT[1] × VT[2] normalisé
20 cas où N = 3
21 [ retourner VT[1] × VT[2] + VT[2] × VT[3] + VT[3] × VT[1] normalisé
22 fin
  
```

« vers la gauche » à partir de p , c'est-à-dire en ajoutant les points suivant les indices décroissants, de façon à ce que $BS_\nu(S, n - i, 0)$ et $\neg BS_\nu(S, n - i - 1, 0)$ pour un certain $i, 0 < i < n$. Puis nous étendons S « vers la droite » de manière à ce que $MBS_\nu(S, n - i, j)$ avec $0 \leq j < n - i$, obtenant ainsi le segment flou maximal « le plus à gauche » contenant p . Nous appliquons alors la procédure de « glissement » décrite auparavant tant que le point p appartient au segment flou maximal S , c'est-à-dire tant qu'il existe des indices k, l tels que $MBS_\nu(S, n - k \bmod n, l)$, avec $0 \leq k < i$ et $j < l < n$.

- Le vecteur directeur de chacun des segments maximaux est obtenu à partir des caractéristiques de la droite discrète supportant le segment flou, récupérées grâce à la fonction `Caractéristiques(S)` de la ligne 10. Ces vecteurs directeurs sont additionnés entre eux et le vecteur résultant est étendu en 3D grâce à la fonction `Extension3D(\vec{T} , i)` à la ligne 12, qui ajoute une coordonnée nulle dans la dimension i . Ce vecteur résultant représente une estimation du vecteur tangent en p dans la tranche i et il est stocké dans le tableau VT. (En pondérant les éléments de la somme, comme cela a été fait dans [LVdV05, LVdV07] pour le cas des segments discrets 2D réguliers, nous pourrions retrouver l'estimateur λ -MST, ici généralisé au cas des segments flous.)
3. Enfin, suivant le nombre de vecteurs non colinéaires contenus dans VT, le vecteur normal en p est obtenu soit comme (somme de) produit(s) vectoriel(s) entre les vecteurs de VT soit, pour le cas d'un seul vecteur \vec{V} dans VT, en prenant le vecteur orthogonal à \vec{V} , pointant vers l'extérieur de l'objet, dans le plan de la tranche où \vec{V} a été estimé. Dans tous les cas, le vecteur renvoyé est normalisé.

La figure 4.3 montre différentes tranches d'une sphère discrète bruitée de rayon 20, avec une estimation du champ de vecteur normaux sur les points du bord de cette sphère dans ces tranches. L'estimation a été faite aux épaisseurs $\nu = 1$ et $\nu = 3$.

Des résultats quantitatifs plus détaillés, notamment sur les différences angulaires avec les normales théoriques, sont proposés ultérieurement, à la section 4.2.3. Nous verrons notamment que les résultats obtenus avec cette méthode ne sont pas aussi bons qu'espéré. Bien que l'utilisation des segments flous apporte une certaine flexibilité au niveau de l'estimateur – on peut le noter visuellement sur la figure 4.3 en remarquant que les normales à l'épaisseur 3 sont « mieux organisées » qu'à l'épaisseur 1 (ce que confirment les chiffres du tableau 4.1) – l'estimateur reste assez sensible au bruit. Et plus précisément certains contours considérés dans les tranches de l'objet sont très perturbés par le bruit et ne fournissent pas d'information fiable et cohérente sur la géométrie de l'objet. On peut s'en rendre compte sur la figure 4.4 : certaines courbes de tranches existent uniquement à cause des perturbations présentent au bord de l'objet discret. Les directions de tangentes obtenues sur ces courbes ne sont du coup pas fiables du tout et faussent complètement l'estimateur. Cette observation n'est évidemment pas caractéristique de cet estimateur, puisque ce comportement apparaîtrait aussi avec l'utilisation des estimateurs déjà existants, décrits à la section précédente. Le problème vient ici du contour de tranche lui-même, qui ne devrait pas être considéré.

Il semblerait donc que l'utilisation d'estimateurs de caractéristiques géométriques 2D et leur recombinaison pour fournir un estimateur 3D soit une approche qui reste très sensible à certaines perturbations de l'objet, même en se basant sur des primitives floues 2D. C'est pour cela que nous avons proposé un estimateur utilisant non pas les tranches de l'objet et des primitives floues 2D, mais un estimateur utilisant le voisinage 3D au bord de l'objet et une primitive floue 3D : les *morceaux flous de plans discrets*.

Malgré tout, il est à noter que de récents travaux, proposés dans [KL09], pourraient aider à résoudre ce problème. En effet, à l'aide d'une analyse multi-échelle de courbes discrètes, les auteurs proposent un moyen d'estimer le « niveau de bruit » localement présent dans ces dernières. Cette information pourrait alors potentiellement nous servir à déterminer si un contour est fiable ou non, ce qui rendrait les estimateurs par tranche plus robustes au bruit. Bien évidemment, des expérimentations doivent être menées afin de valider cette intuition.

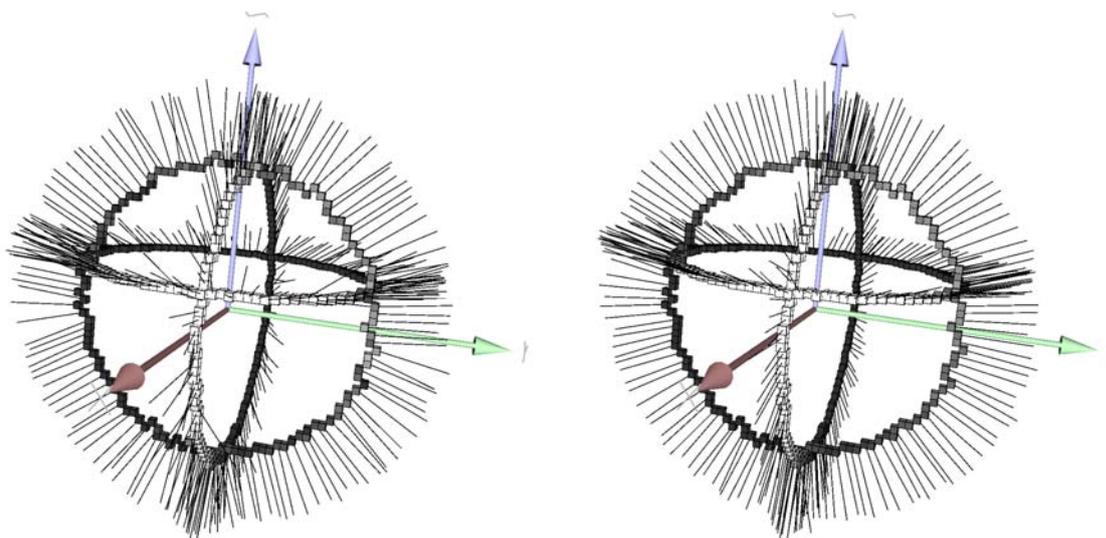
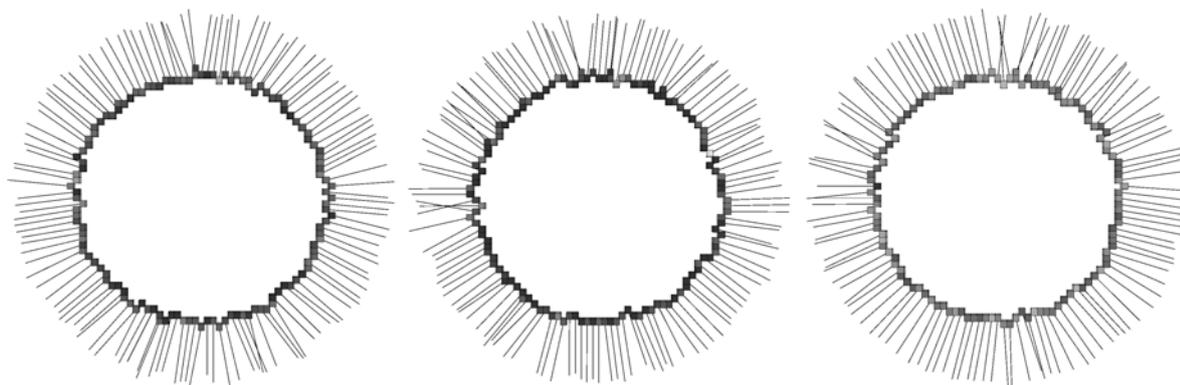
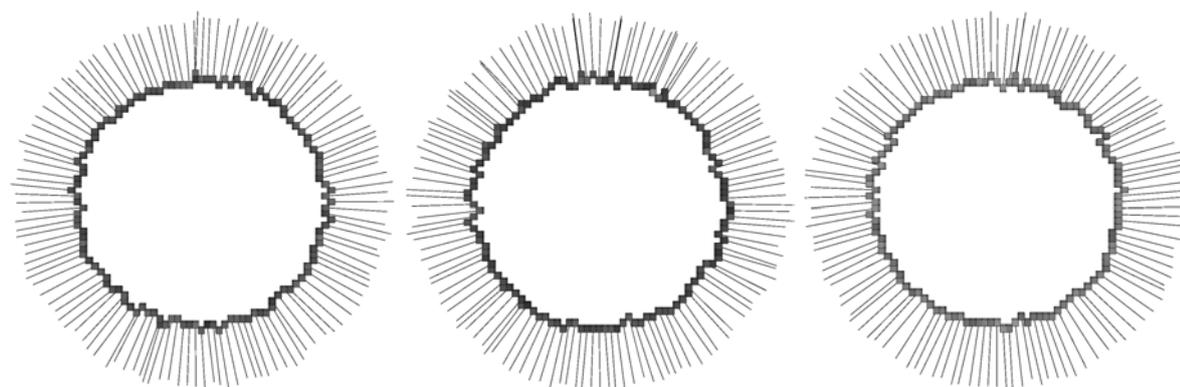
(a) Estimation des normales à l'épaisseur $\nu = 1$ (b) Estimation des normales à l'épaisseur $\nu = 3$ (c) Champ de normales à l'épaisseur $\nu = 1$ pour les tranches (de gauche à droite) $X = 3$, $Y = 2$ et $Z = 6$ (d) Champ de normales à l'épaisseur $\nu = 3$ pour les tranches (de gauche à droite) $X = 3$, $Y = 2$ et $Z = 6$

Figure 4.3 – Estimation du champ de vecteurs normaux sur les voxels du bord d'une sphère discrète bruitée de rayon 20. Seules quelques tranches sont considérées pour une meilleure visualisation.

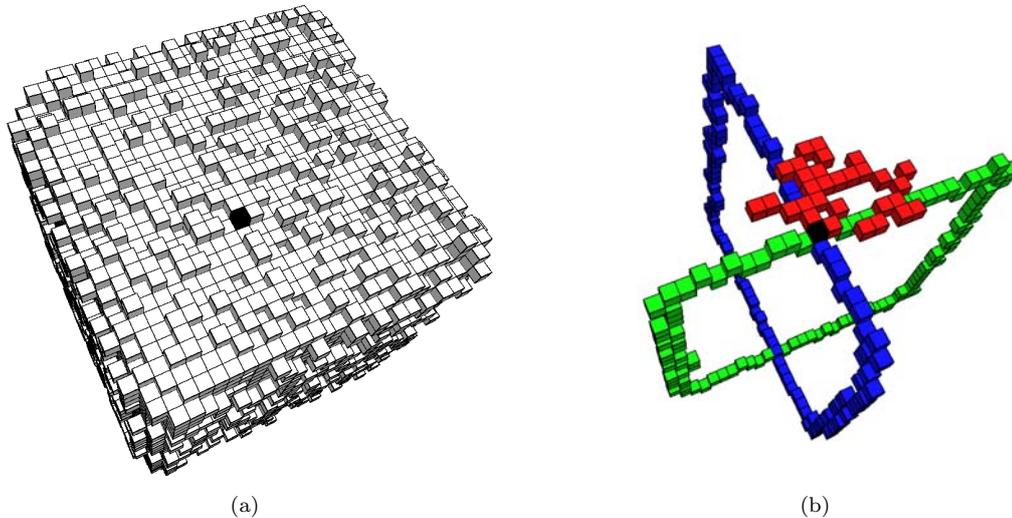


Figure 4.4 – (a) Un cube discret bruité d’arête 25 ; (b) les trois courbes de tranches passant par le point noir. La courbe rouge n’est pas du tout fiable car elle est composée uniquement de bruit, c’est-à-dire que cette courbe n’existerait pas dans le cas d’un cube parfait.

4.2.2 Approche multi-directionnelles

Afin de fournir une estimation plus fiable du vecteur normal en un point p du bord d’un objet discret, nous n’allons pas nous limiter aux tranches de l’objet, mais nous allons proposer une méthode utilisant les informations géométriques fournies par *tous* les points d’un voisinage étendu de p . Cette idée a déjà été exploitée dans différents travaux, mais la taille du voisinage est généralement un paramètre de la méthode.

4.2.2.1 Méthodes utilisant un voisinage de taille fixe

Parmi les approches surfaciques, nous pouvons citer [CHRU85], qui dans le cadre des complexes cellulaires, propose une estimation de la normale en un surfel s de la surface d’un objet discret à partir d’une moyenne des normales élémentaires des quatre surfels adjacents à s par arête. Comme il existe seulement trois configurations possibles pour deux surfels adjacents par arête, cela donne 81 configurations au total, produisant seulement 25 directions différentes de normales. De plus le voisinage étant très local, on devine aisément que du bruit présent au niveau de la surface risque de beaucoup perturber les calculs.

Une solution possible consiste alors à considérer un voisinage plus grand pour estimer la normale. En suivant cette idée, L. Papier et J. Françon proposent un estimateur de vecteur normal basé sur la notion d’*ombrelle d’ordre n* [PF98, Pap99]. Plus précisément, en considérant une approche cuberille, les définitions suivantes amènent à la notion d’ombrelle d’ordre n .

Définition 4.3 On appelle *ombrelle en un pointel P* appartenant à la surface en surfels d’un objet discret, une permutation circulaire, à l’orientation près, de surfels de la surface (2 à 2 adjacents) incidents au pointel P . L’ensemble des surfels de cette ombrelle, noté $\mathcal{O}_0(P)$, sera appelé *ombrelle d’ordre 0* en P .

Définition 4.4 On appelle *ombrelle d’ordre 0* en un voxel V du bord d’un objet discret, et on note $\mathcal{O}_0(V)$, l’ensemble des surfels incidents à v qui appartiennent à la surface en surfels de l’objet.

La notion d’ombrelle d’ordre n généralise ces définitions en étendant de manière récursive l’ensemble des surfels considérés ; l’ombrelle d’ordre supérieur étant obtenue en ajoutant, par voisinage, les surfels de la surface incidents à ceux de l’ombrelle inférieure, comme illustré à la figure 4.5(a).

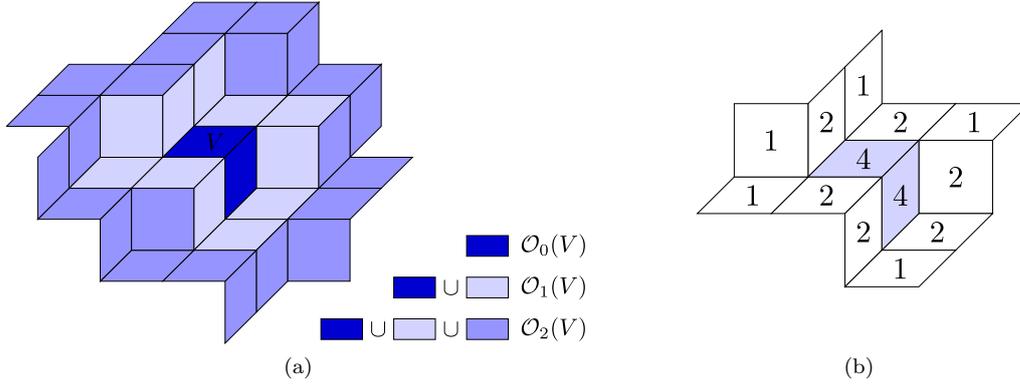


Figure 4.5 – (a) Les surfels de l’ombrelle d’ordre 2 au voxel V . (b) Les coefficients de pondération intervenant dans le calcul de la normale pondérée d’ordre 1 au voxel bleu.

Définition 4.5 On appelle *ombrelle d’ordre n* ($n > 0$) en R (pointel ou voxel), l’ensemble de surfels

$$\mathcal{O}_n(R) = \bigcup_{Q \in \mathcal{P}_{n-1}(R)} \mathcal{O}_0(Q)$$

où $\mathcal{P}_i(R)$ désigne l’ensemble des pointels de l’ombrelle d’ordre i en R .

En se basant sur cette notion d’ombrelle d’ordre n , les auteurs définissent un estimateur de vecteur normal, généralisant celui proposé dans [CHRU85] à un voisinage plus grand. Deux versions sont proposées, l’une sans pondération, où les normales élémentaires en chaque surfel du voisinage considéré ont un apport identique dans le calcul du vecteur normal résultant, et l’autre avec pondération, mais implicite à la définition, destinée à donner plus d’importance aux normales élémentaires des surfels du voisinage proches (relativement à la topologie) du point où est estimé la normale, comme illustré à la figure 4.5(b). Ainsi :

Définition 4.6 La *normale d’ordre n* ($n \geq 0$) en R (pointel ou voxel) est le vecteur

$$\vec{n}_n(R) = \sum_{s \in \mathcal{O}_n(R)} \vec{n}_e(s)$$

Définition 4.7 La *normale pondérée d’ordre n* ($n \geq 0$) en R (pointel ou voxel) est le vecteur

$$\vec{\mathcal{N}}_n(R) = \sum_{i=0}^{n-1} \sum_{Q \in \mathcal{P}_0(R)} \vec{n}_i(Q)$$

où $\mathcal{P}_0(R)$ désigne l’ensemble des pointels de l’ombrelle d’ordre 0 en R .

Les résultats proposés dans [PF98] tendent à montrer que l’utilisation d’un voisinage étendu permet d’améliorer l’estimation du vecteur normal lorsque la taille du voisinage augmente, avec pour effet d’augmenter l’effet lissant au niveau des arêtes vives de l’objet. La méthode fournit donc une certaine flexibilité et pourrait être utilisée sur des objets bruités, mais la taille du voisinage est à fixer a priori, et ne semble pas très intuitive à déterminer.

Dans [Coe02b], D. Coeurjolly propose une modification de la méthode précédente afin d’étendre le voisinage non pas par des critères topologiques, comme la notion d’ombrelle d’ordre n , mais grâce à une métrique géodésique discrète qu’il a proposée dans [Coe02a]. Ainsi, le calcul de la normale en un surfel s de la surface discrète ne considère que les surfels qui sont à une distance géodésique de s inférieure à un

certain seuil δ . L'idée est d'avoir un processus de pondération des normales élémentaires isotropique sur la surface. Peu de résultats sont proposés, ne nous permettant pas de conclure sur la qualité de l'estimateur et les opérations intervenant dans le calcul de la distance géodésique paraissent plutôt lourdes. De plus, le seuil δ doit aussi être fixé a priori, ce qui ne rend pas la technique adaptative.

Au niveau des approches volumiques, G. Thürmer et C.A. Wüthrich proposent d'estimer le vecteur normal en un point p du bord d'un objet discret en utilisant l'information des points situés dans un voisinage cubique de taille ns autour de p [TW97]. Le vecteur normal au point $p(x, y, z)$ est obtenu en normalisant le vecteur n' dont les composantes x' , y' et z' sont calculées comme suit :

$$\begin{aligned} x' &= \sum_{i=-ns}^{ns} \sum_{j=-ns}^{ns} \sum_{k=-ns}^{ns} \frac{w_{i,j,k} \cdot \sigma_{i,j,k} \cdot i}{\sqrt{i^2 + j^2 + k^2}} \\ &\text{avec } i \neq 0 \vee j \neq 0 \vee k \neq 0 \\ y' &= \sum_{i=-ns}^{ns} \sum_{j=-ns}^{ns} \sum_{k=-ns}^{ns} \frac{w_{i,j,k} \cdot \sigma_{i,j,k} \cdot j}{\sqrt{i^2 + j^2 + k^2}} \\ &\text{avec } i \neq 0 \vee j \neq 0 \vee k \neq 0 \\ z' &= \sum_{i=-ns}^{ns} \sum_{j=-ns}^{ns} \sum_{k=-ns}^{ns} \frac{w_{i,j,k} \cdot \sigma_{i,j,k} \cdot k}{\sqrt{i^2 + j^2 + k^2}} \\ &\text{avec } i \neq 0 \vee j \neq 0 \vee k \neq 0 \end{aligned}$$

où

$$\sigma_{i,j,k} = \begin{cases} 1 & \text{si le point } q(x+i, y+j, z+k) \text{ appartient à l'objet discret} \\ 0 & \text{sinon} \end{cases}$$

Une pondération des points du voisinage de p peut être faite grâce aux valeurs du coefficient $w_{i,j,k} \in]0, 1]$. Ainsi, plusieurs choix sont proposés par les auteurs : $w_{i,j,k} = 1$ pour une équité dans l'apport de chaque point, ou bien, dans l'idée de donner moins d'importance aux points lointains de p dans le voisinage considéré, $w_{i,j,k} = \frac{1}{i^2+j^2+k^2}$, soit l'inverse de la distance euclidienne au carré entre p et les points du voisinage, ou encore $w_{i,j,k} = \frac{1}{|i|+|j|+|k|}$, soit l'inverse de la distance de Manhattan entre les points.

Une étude plus détaillée sur l'influence de chaque paramètre est fournie dans [TW98]. Il est notamment observé que dans le cas où le bord discret considéré est issu de la discrétisation de surfaces à pente ou à courbure constante, plus ns augmente, meilleure est l'estimation. Au contraire, pour des discrétisations de surface dont la pente varie beaucoup au voisinage de p , l'erreur d'estimation, par rapport à la normale théorique dans le continu, diminue lorsque la taille du voisinage est réduite ; un voisinage étendu ayant tendance à lisser les variations entre les normales. Le choix de ns est donc très dépendant de l'objet manipulé et du contexte lorsqu'il s'agit de minimiser l'erreur et donc il est difficile de fixer ns a priori.

Dans [Thü01], une extension de la méthode est proposée en moyennant les vecteurs normaux dans un voisinage dépendant de la configuration locale de la surface, afin de lisser les petites variations entre les vecteurs normaux tout en conservant celles dues à des discontinuités significatives au bord de l'objet. Pour cela, une première estimation du champ de vecteurs normaux est effectuée grâce aux formules précédentes, avec $ns = 1$. Le vecteur normal $n_p(a, b, c)$ obtenu en un point p du bord de l'objet permet donc d'extraire l'équation de l'estimation du plan tangent $T : ax + by + cz + d = 0$ en p . Étant donné un réel $h \geq 0$, nous pouvons alors créer un « volume tangent » $V_T = \{(i, j, k) \in \mathbb{Z}^3 : -d - h \leq ai + bj + ck \leq -d + h\}$ représentant les points de l'espace discret compris entre deux plans qui correspondent au plan T translaté d'une distance h suivant n_p et $-n_p$. Le vecteur normal « lissé » au point du bord B de l'objet peut alors être évalué grâce à la formule :

$$n_p^l = \frac{\sum_{\forall q \in (N_e(p) \cap B)} w(p, q) n_q}{\left\| \sum_{\forall q \in (N_e(p) \cap B)} w(p, q) n_q \right\|}$$

où n_q représente le vecteur normal estimé en un point q du bord de l'objet appartenant au voisinage $N_e(p)$ de taille e autour de p , et $w(p, q)$ est un coefficient de pondération dépendant de la distance entre p et q . De plus, e doit être le plus grand entier telle que les deux conditions suivantes soient vérifiées :

- (i) $\forall q \in (N_e(p) \cap B) : q \in V_T$
- (ii) $\forall q \in (N_e(p) \cap B) : \alpha(n_p, n_q) < \pi/2$

$\alpha(n_p, n_q)$ étant la valeur de l'angle entre les vecteurs n_p et n_q . Ces conditions permettent d'éviter de considérer des points du voisinage de p qui franchiraient des discontinuités trop importantes (relativement au paramètre h) au niveau des pentes de la surface de l'objet.

Bien que ces méthodes soient utilisables sur des objets discrets bruités, soit la taille du voisinage doit être fixée a priori, soit beaucoup de paramètres et de contraintes sont introduites pour prendre en compte la conformation locale de la surface, rendant les méthodes assez difficiles à ajuster. C'est pourquoi, nous proposons, dans la suite, un estimateur de vecteur normal, adaptatif, reposant sur les morceaux flous de plans discrets où le seul paramètre est l'épaisseur ν .

4.2.2.2 Méthode par voisinage adaptatif : le patch discret d'épaisseur ν

L'idée est la suivante : comme les objets discrets que nous considérons peuvent être bruités, nous ne voulons pas estimer les caractéristiques géométriques dans un voisinage trop petit autour d'un point p du bord de l'objet. Cela conduirait à des estimateurs très sensibles au bruit. Nous voulons donc prendre en compte un voisinage étendu, afin d'extraire des informations sur la conformation locale du bord de l'objet qui soient plus fiables. Comme nous l'avons vu dans la section précédente, se limiter à certaines directions (les tranches de l'objet en l'occurrence) peut conduire à une estimation peu fiable de la géométrie autour de p . Nous désirons donc utiliser l'information apportée par *tous* les points autour de p , en considérant un voisinage étendu uniformément dans toutes les directions. Cela concorde avec l'idée de D. Coeurjolly pour contrôler l'isotropie du voisinage. Néanmoins, afin d'affiner au mieux l'estimation, nous ne voulons pas fixer de limite a priori sur la taille de ce voisinage. Par contre, nous voulons tout de même garder un certain contrôle sur les déformations locales du bord de l'objet, correspondant à un éventuel bruit dans le voisinage considéré. C'est pourquoi nous avons décidé que les points du voisinage devaient être un sous-ensemble d'un plan discret épais. Plus particulièrement, le voisinage doit localement rester un MFPD $_\nu$, le paramètre ν nous permettant de contrôler les variations de la géométrie autour de p . Dans ce but, nous introduisons la notion de patch discret d'épaisseur ν .

Définition 4.8 Soit \mathcal{B}_b le bord d'un objet discret et $p \in \mathcal{B}_b$. Soit T un processus de parcours des points de \mathcal{B}_b dans le voisinage de p et ν un réel. On appelle **patch discret d'épaisseur ν centré en p** , et on note $\Gamma_\nu(p)$, un morceau flou de plan discret d'épaisseur ν reconnu incrémentalement à partir de p en ajoutant les points de \mathcal{B}_b dans le voisinage de p en suivant le processus de parcours T .

Afin de concrétiser cette définition, il nous reste à présenter le processus de parcours T employé, et la condition d'arrêt de l'extension d'un patch discret d'épaisseur ν . La reconnaissance incrémentale de MFPD $_\nu$ est quant à elle assurée par l'algorithme 3.2 proposé à la section 3.4.2.2 page 62.

Le processus de parcours T : Pour étendre le patch discret dans toutes les directions de manière uniforme, la meilleure solution serait d'utiliser un parcours des points du bord dans le voisinage de p en fonction d'une distance géodésique, comme proposé par D. Coeurjolly. Néanmoins, les opérations réalisées pour calculer cette distance géodésique semblent assez lourdes, donc nous avons choisi de parcourir le voisinage relativement à une distance de chanfrein basée sur le masque $\langle 3, 4, 5 \rangle$ [Bor96, TM92]. Le but ici est juste d'obtenir un patch *presque* circulaire, étendu uniformément dans toutes les directions. Pour implémenter ce comportement, nous utilisons une file de priorité Q . Pour le développement d'un patch discret centré en p , nous commençons par insérer p dans la queue avec un poids égal à zéro. Puis, tant que la condition d'arrêt n'est pas atteinte, nous retirons de Q le point v associé au plus petit poids w et nous l'ajoutons au MFPD $_\nu$ en cours de reconnaissance. La file de priorité est ensuite mise à jour en y ajoutant les points du bord de l'objet qui appartiennent au 26-voisinage de v de la manière suivante :

- les 6-voisins de v sont ajoutés dans Q avec un poids $w + 3$;
- les 18-voisins stricts de v sont ajoutés dans Q avec un poids $w + 4$;
- et les 26-voisins stricts de v sont ajoutés dans Q avec un poids $w + 5$.

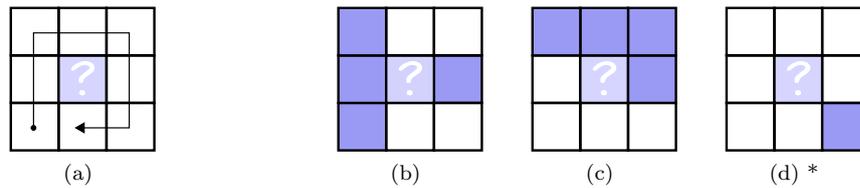


Figure 4.6 – Les points qui appartiennent au patch en cours d’extension sont représentés en bleu. (a) L’ordre de parcours des points du voisinage d’un point à ajouter au patch dans un des plans de projection canonique. (d)-(c) Quelques configurations possibles lorsqu’on désire ajouter le point marqué d’un point d’interrogation : (b) le point ne peut pas être ajouté car il crée un trou ; (c) le point peut être ajouté ; (d) le point ne peut pas être ajouté car il n’est pas 4-connecté au patch (* utilisé dans l’algorithme de segmentation proposé au chapitre suivant).

Une technique efficace pour implémenter une telle file de priorité a été proposée par B.J.H. Verwer et ses coauteurs [VVD89]. Elle repose sur l’utilisation de plusieurs listes (« bucket »), supportant les opérations d’ajout et de suppression en temps constant (une liste doublement chaînée, par exemple), indexées chacune par le poids des points qu’elle contient. L’élément de poids le plus faible est extrait en choisissant le premier élément p de la liste de plus petit indice i , et ses voisins non visités v sont ajoutés dans les listes d’indice $i + poids(p, v)$. Quand un point a été traité, il est marqué *visité*. Comme un point p , en cours de traitement, ne peut faire entrer dans une liste qu’un voisin avec un poids compris entre le poids de p et le poids de p plus le poids maximum entre deux points, le nombre de listes est majoré par le poids maximum entre deux points plus 1. On peut donc « recycler » une liste vide, et ainsi les files peuvent être adressées modulo le poids maximum entre deux points plus 1 (qui dans notre cas vaut 6). Cette idée a déjà été utilisée dans plusieurs travaux considérant un parcours de voisinage sur une surface discrète [Coe02b, Siv04].

À ce parcours se rajoute une condition, celle de rester homéomorphe à un disque topologique. Nous voulons en effet conserver une notion de « contour » unique au niveau d’un patch, car, comme nous le verrons dans la section 4.3.4, l’estimateur de forme utilisera cette propriété. Pour ce faire, nous reprenons la technique utilisée dans [Siv04] page 153, dans une version simplifiée. Le 8-voisinage du point dans les plans de projection canoniques est testé pour vérifier qu’aucun trou n’est introduit. Pour cela, un compteur initialisé à zéro est incrémenté à chaque fois que l’on passe d’un point appartenant au patch à un point n’appartenant pas au patch, et vice-versa, lors du parcours de ce voisinage de la manière illustrée à la figure 4.6(a). À la fin, si la valeur du compteur est supérieure à deux, cela signifie que le point ne peut pas être ajouté sans créer de trou. Si un point ne répond pas à ce critère il n’est pas ajouté au patch.

La condition d’arrêt : Nous construisons un patch discret d’épaisseur ν centré en p grâce à l’algorithme de reconnaissance 3.2. Les points sont ajoutés en suivant le processus de parcours T et dès que la hauteur du morceau flou de plan discret en cours de reconnaissance dépasse ν , l’extension du patch s’arrête.

En appliquant cette méthode sur le bord d’un objet discret, nous obtenons des résultats comme ceux illustrés à la figure 4.7.

Ainsi, un patch discret $\Gamma_\nu(p)$ est, par définition, un morceau flou de plan discret développé autour de p , c’est-à-dire une zone planaire (relativement au paramètre d’épaisseur ν) autour de p . Par conséquent ce patch peut être considéré comme une approximation du plan tangent en p au bord de l’objet discret. Son vecteur normal devient alors une bonne estimation de la normale en p au bord de l’objet discret. Ainsi, en assimilant le vecteur normal de $\Gamma_\nu(p)$ à la normale du bord en p , nous définissons un estimateur de vecteur normal pour tout point du bord d’un objet discret possiblement bruité.

Définition 4.9 Soient \mathcal{B}_b le bord d’un objet discret et un point $p \in \mathcal{B}_b$. On appelle *normale d’épaisseur*

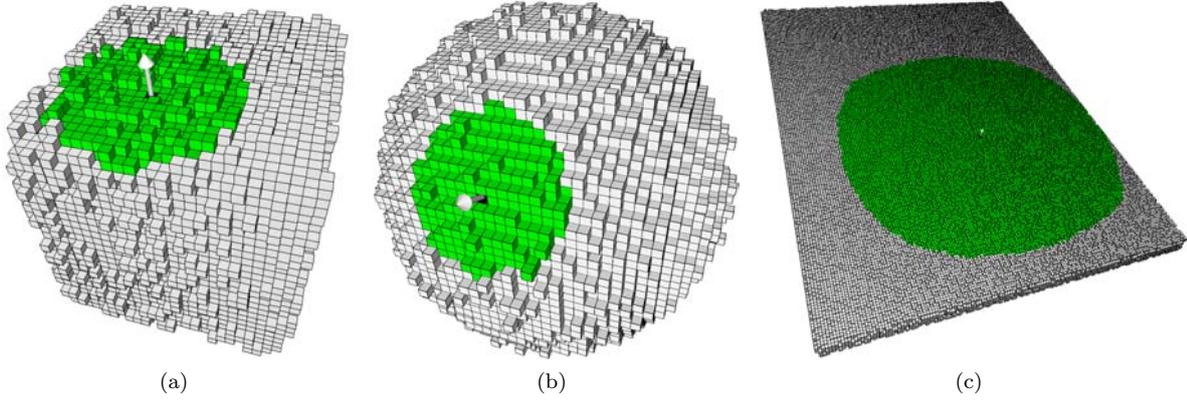


Figure 4.7 – Un exemple de patches discrets d'épaisseur 2 développés au bord de différents objets discrets bruités. (a) Un cube d'arête 25, (b) une sphère de rayon 20 et (c) un pavé de dimensions $150 \times 6 \times 200$.

ν **au point** p , et on note $\vec{n}_\nu(p)$, le vecteur normal

$$\vec{n}_\nu(p) = \vec{n}(\Gamma_\nu(p))$$

où $\vec{n}(\Gamma_\nu(p))$ représente le vecteur normal du MFPD $_\nu$ support du patch $\Gamma_\nu(p)$.

La figure 4.8 montre un résultat visuel de l'estimation, à deux épaisseurs, du champ de vecteurs normaux au bord d'une demi-ellipsoïde creusée bruitée. Comme on peut s'en rendre compte, l'augmentation de la valeur de ν permet d'absorber les perturbations du bord de l'objet et fournit un résultat visuel plus proche des attentes que l'on pourrait avoir.

Dans la section suivante nous proposons une étude quantifiant les différences angulaires des vecteurs issus de cet estimateur avec les normales théoriques, et nous présentons une comparaison entre différents estimateurs de vecteurs normaux.

4.2.3 Résultats

Les résultats de l'estimation du champ de vecteurs normaux au bord de différents objets discrets sont présentés dans cette section. Nous proposons en particulier une étude donnant les différences angulaires entre les vecteurs normaux théoriques et ceux issus de l'estimation effectuée suivant quatre méthodes : l'estimateur proposé par Thürmer et Wüthrich, dans sa version originelle, l'estimateur proposé par Papier et Françon, basé sur les ombrelles d'ordre n et nos deux estimateurs utilisant les tranches de l'objet et les patches discrets. L'estimation sera effectuée sur deux types d'objets dont il est facile de connaître le vecteur normal théorique : des sphères discrètes et des morceaux de plans discrets. Ces objets discrets ont été générés en utilisant leurs définitions arithmétiques, déjà évoquées dans les précédents chapitres, à savoir

$$\left\{ (x, y, z) \in \mathbb{Z}^3 \mid -\|(a-x, b-y, c-z)\|_\infty \leq (x-a)^2 + (y-b)^2 + (z-c)^2 - R^2 < \|(a-x, b-y, c-z)\|_\infty \right\}$$

pour les sphères discrètes de centre (a, b, c) et de rayon R , et

$$\left\{ (x, y, z) \in \mathbb{Z}^3 \mid \mu \leq ax + by + cz < \mu + \omega \right\}$$

pour les plans arithmétiques de normale (a, b, c) , de paramètre de translation μ et d'épaisseur arithmétique ω . Dans le cas d'une sphère de centre c , la direction du vecteur normal théorique en un point p est donnée par le vecteur \vec{cp} .

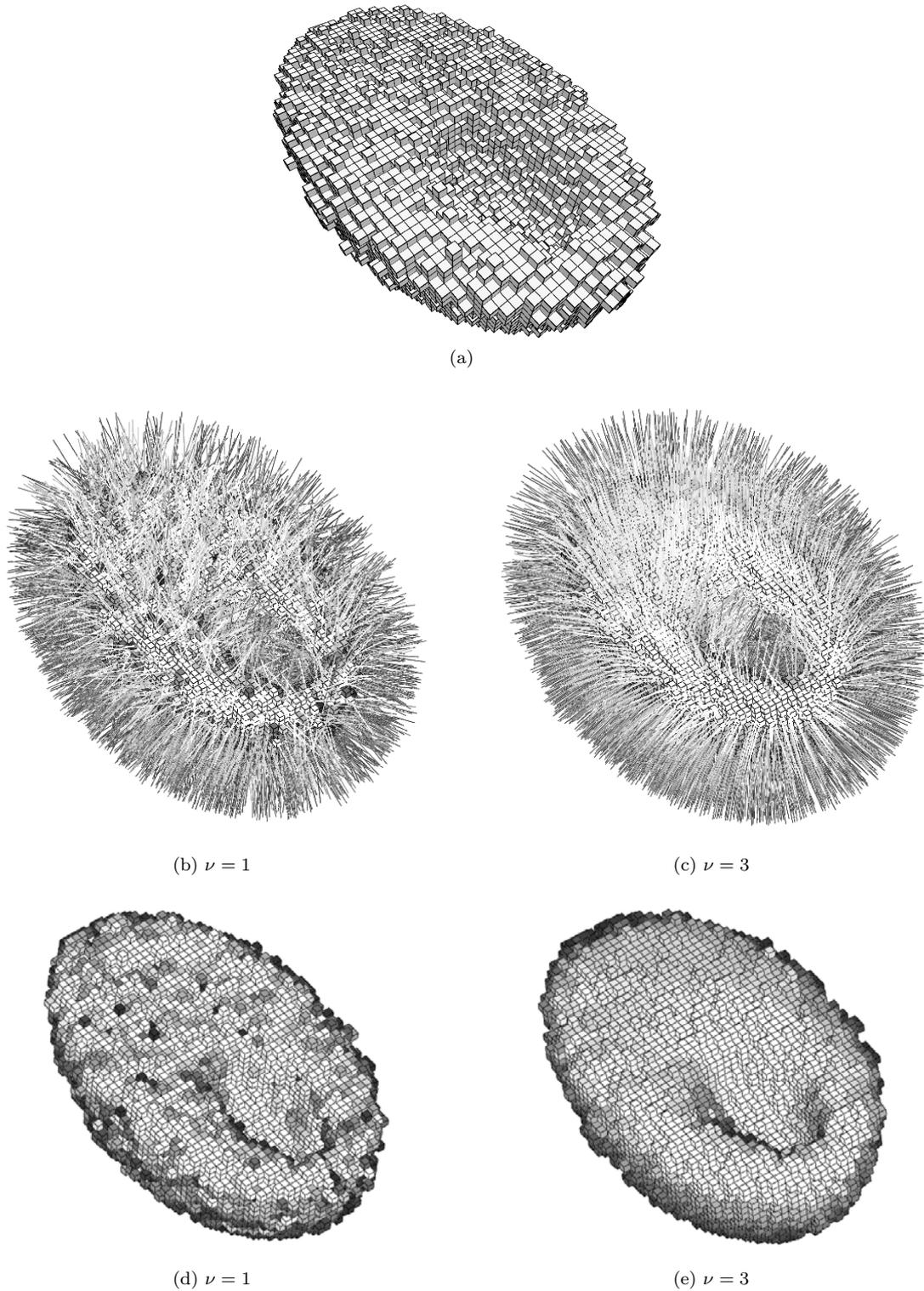


Figure 4.8 – (a) Une demi-ellipsoïde creusée bruitée et l'estimation du champ de vecteurs normaux au bord de cet objet (b) à l'épaisseur 1 et (c) à l'épaisseur 3. Un rendu de l'objet sous OpenGL, en affectant à chaque face d'un voxel le vecteur normal correspondant à la normale estimée au voxel, (d) à l'épaisseur 1 et (e) à l'épaisseur 3.

Approche		Tranches								
Épaisseur ν		1			2			3		
Sphères testées	Rayon	δ_{moy}	δ_{max}	σ	δ_{moy}	δ_{max}	σ	δ_{moy}	δ_{max}	σ
Non bruitée	10	4.72	13.15	2.71	2.49	8.35	1.43	4.54	12.39	2.76
	20	3.69	8.92	1.97	3.12	8.94	1.77	2.01	9.09	1.44
Faiblement bruitée	10	8.34	39.88	4.54	6.67	162.37	11.11	5.59	168.28	11.47
	10 [†]	8.34	39.88	4.54	5.88	24.37	3.20	4.77	73.30	3.28
	20	6.44	72.65	3.96	5.56	179.39	16.26	4.92	179.39	16.44
	20 [†]	6.44	72.65	3.96	3.86	34.80	2.11	3.18	14.93	1.78
Fortement bruitée	10	11.24	173.86	13.53	9.65	175.39	18.52	7.81	178.34	18.42
	10 [†]	10.28	79.01	6.68	7.58	90.00	6.46	5.73	90.00	5.57
	20	9.23	176.79	14.34	7.46	177.37	18.43	6.05	178.00	18.18
	20 [†]	8.07	87.22	6.35	5.39	88.36	6.17	4.06	88.36	5.90

Tableau 4.1 – Différences angulaires δ (en degrés $^\circ$) entre le vecteur normal théorique et celui provenant de l'estimateur utilisant les tranches de l'objet discret, sur le bord de différentes sphères discrètes.

†. Sans tenir compte des outliers (différence angulaire supérieure à 90 degrés).

Nous considérons les objets en version bruitée et non bruitée, la version bruitée étant générée à partir de la méthode décrite en annexe, à la section A.1. Les versions que nous qualifions « *faiblement* » bruitées sont obtenues en appliquant l'algorithme de bruitage une fois, et celles qualifiées « *fortement* » bruitées sont obtenues en effectuant deux passes de l'algorithme.

En ce qui concerne les approches reposant sur des primitives floues, les estimations ont été faites aux épaisseurs 1, 2 et 3. Pour les deux autres approches testées, [TW97] et [PF98], il a fallu fixer une taille de voisinage. Nous avons choisi des tailles de 1, 3 et 6, correspondant à un voisinage contenu dans une boîte englobante de taille similaire à celle contenant des patches développés sur une sphère de rayon 20, aux épaisseurs respectives 1, 2 et 3. Nous obtenons ainsi des chiffres raisonnablement comparables. Nous avons de plus opté pour les versions non pondérées des estimateurs, considérant ainsi une équité dans l'apport d'information de chaque point du voisinage. Dans [TW98] les auteurs avaient d'ailleurs noté que l'influence de la pondération était minime dans le cas d'objets à courbure constante comme les sphères.

Les tableaux 4.1 à 4.4 résument les résultats obtenus. Dans le cas des tableaux 4.2 à 4.4, les différences angulaires δ (exprimées en degrés) entre les vecteurs normaux théoriques et ceux issus de l'estimation étant bien réparties dans l'intervalle $[\delta_{min}, \delta_{max}]$ et la valeur minimale étant généralement très proche de zéro, nous avons choisi de résumer les résultats en recensant pour chaque cas, la différence angulaire moyenne δ_{moy} , la valeur maximale de la différence angulaire δ_{max} et l'écart-type σ des différences angulaires.

Dans le cas du tableau 4.1, décrivant les résultats obtenus en utilisant l'estimateur par tranches de normales à l'épaisseur ν , nous avons recensé les mêmes informations, mais elles ne sont pas aussi pertinentes car un certain nombre de données aberrantes (outliers) perturbe les résultats. Nous notons en effet des différences pouvant aller jusqu'à presque 180°. Pour cet estimateur, nous avons aussi considéré les résultats sans prendre en compte les différences angulaires supérieures à 90°. En éliminant ces outliers des calculs, nous constatons que les chiffres obtenus sont plus raisonnables, et laisseraient envisager une possible utilisation de cet estimateur. Néanmoins, la principale cause d'apparition de ces données aberrantes est la considération des contours des tranches de l'objet bien qu'ils soient issus de perturbations dues à un possible bruit. Ces dégénérescences apparaissent particulièrement dans les parties du bord de l'objet dont la normale est proche d'être orthogonale à l'un des trois plans canoniques de l'espace 3D définissant les tranches. La normale estimée en ces zones n'est donc pas toujours correcte (on peut s'en rendre compte au quatre points cardinaux des cercles de la figure 4.3(c), par exemple). Or, déterminer

Approche		Patches									
Épaisseur ν		1			2			3			
État	Objet testé	δ_{moy}	δ_{max}	σ	δ_{moy}	δ_{max}	σ	δ_{moy}	δ_{max}	σ	
Non bruité	Sphère	$R = 5$	5.02	15.23	3.46	3.13	9.07	3.16	3.62	10.39	2.73
		$R = 10$	3.66	8.21	1.83	2.33	9.92	1.49	1.85	6.49	0.98
		$R = 20$	2.05	6.10	1.07	1.25	3.35	0.61	1.17	3.93	0.69
		$R = 40$	1.04	3.25	0.58	0.77	2.93	0.44	0.70	2.58	0.40
	Plan	$(8, -13, 70)$	0	0	0	0	0	0	0	0	0
		$(3, 4, -5)$	0	0	0	0	0	0	0	0	0
Faiblement bruité	Sphère	$R = 5$	11.59	53.96	7.66	5.57	23.20	3.43	4.70	21.52	2.60
		$R = 10$	10.36	52.53	7.70	3.66	16.86	2.19	2.75	8.90	1.54
		$R = 20$	9.08	91.25	7.50	2.39	10.16	1.35	1.74	7.46	0.92
		$R = 40$	9.04	101.24	7.81	1.46	5.83	0.78	1.02	3.89	0.55
	Plan	$(8, -13, 70)$	11.59	88.93	13.66	0	0	0	0	0	0
		$(3, 4, -5)$	6.49	84.26	11.76	0	0	0	0	0	0
Fortement bruité	Sphère	$R = 5$	16.71	86.20	11.91	7.60	29.05	5.03	4.90	14.96	2.82
		$R = 10$	15.30	95.46	12.46	5.65	46.98	3.77	3.29	14.63	1.90
		$R = 20$	14.10	101.42	11.90	4.57	40.54	3.33	2.37	8.80	1.30
		$R = 40$	14.33	117.79	11.95	4.35	52.57	3.97	1.70	7.32	0.93
	Plan	$(8, -13, 70)$	18.10	88.93	15.83	0.92	64.85	3.87	0	0	0
		$(3, 4, -5)$	12.06	87.30	16.52	0	0	0	0	0	0

Tableau 4.2 – Différences angulaires δ (en degrés $^\circ$) entre le vecteur normal théorique et celui provenant de l'estimateur utilisant les patches discrets, sur le bord de différents plans et sphères discrets.

en pratique si une tranche de l'objet est à éliminer car elle est issue d'un potentiel bruit, semble plutôt compliqué dans un cas général. Donc pour des objets présentant des perturbations importantes, comme celles introduites ici, cet estimateur ne semble pas approprié. Il n'est pas pour autant dénué d'intérêt. En effet, dans le cas d'objets discrets réguliers, les résultats sont plus positifs. Nous pouvons donc penser que, pour des objets discrets comportant peu de perturbations locales n'amenant pas à créer de contour de tranche non fiable, l'utilisation de cet estimateur pourrait être envisageable.

Enfin, si les intuitions perçues à travers les travaux de [KL09] venaient à se confirmer, alors un début de solution pourrait même être apporté pour éliminer les contours non fiables, laissant finalement espérer une utilisation de ce type d'estimateurs sur des objets présentant ce type de défauts.

Néanmoins, pour le moment, les autres approches offrent des résultats plus intéressants, notamment au niveau de la robustesse au bruit. Ainsi, si l'on considère cette fois-ci les tableaux 4.1 à 4.4, une première observation, générale à toutes les approches et correspondant à une lecture horizontale des tableaux, est que les différences angulaires entre les normales estimées et théoriques diminuent en moyenne lorsque la taille du voisinage considéré augmente (directement en incrémentant les paramètres n et ns , ou indirectement en augmentant l'épaisseur ν). En particulier pour les objets bruités, la considération d'un voisinage étendu de points permet d'améliorer la qualité de l'estimation.

Une deuxième observation différencie les approches surfaciques (ombrelles et patches) par rapport à l'approche volumique [TW97], au niveau des sphères (lecture verticale des tableaux). En effet, pour cette approche, il semblerait que pour une taille de voisinage donnée, il existe un rayon de sphère idéal pour l'estimateur. Au deçà ou au delà de cette taille de rayon, les différences angulaires augmentent en moyenne. Ceci est peut-être dû au voisinage cubique considéré, et un voisinage sphérique ou un autre choix pour les coefficients de pondération w_{ijk} pourrait modifier ce comportement. Néanmoins ceci introduirait un paramètre supplémentaire par rapport aux deux autres approches. En utilisant les ombrelles ou les patches discrets, au contraire, plus le rayon de la sphère augmente, moins les normales estimées diffèrent des normales théoriques.

Approche			Thürmer et Wüthrich								
Taille du voisinage (ns)			1			3			6		
État	Objet testé		δ_{moy}	δ_{max}	σ	δ_{moy}	δ_{max}	σ	δ_{moy}	δ_{max}	σ
Non bruité	Sphère	$R = 5$	8.20	17.17	4.80	1.66	2.80	0.76	2.74	3.68	0.86
		$R = 10$	7.08	16.54	3.86	2.56	5.11	1.34	0.70	1.22	0.36
		$R = 20$	7.03	18.41	3.84	3.71	6.20	1.61	2.82	4.74	1.07
		$R = 40$	7.39	22.24	4.19	3.77	7.76	1.54	3.40	5.41	1.44
	Plan	$(8, -13, 70)$	9.52	17.22	4.19	3.04	3.95	0.54	2.99	3.16	0.10
		$(3, 4, -5)$	5.39	7.76	1.82	1.87	2.22	0.28	1.84	1.86	0.02
Faiblement bruité	Sphère	$R = 5$	12.67	36.03	6.19	2.32	6.27	1.24	2.84	5.01	1.04
		$R = 10$	12.90	39.27	6.54	2.76	8.45	1.47	0.94	3.16	0.49
		$R = 20$	12.57	46.70	6.73	3.96	10.18	1.76	2.84	5.75	1.13
		$R = 40$	12.84	54.55	6.87	3.83	10.72	1.88	3.31	6.65	1.51
	Plan	$(8, -13, 70)$	10.45	29.48	4.84	3.08	6.14	0.86	2.99	3.72	0.22
		$(3, 4, -5)$	8.14	27.67	4.39	2.15	5.95	0.92	1.85	2.87	0.29
Fortement bruité	Sphère	$R = 5$	15.10	38.83	7.81	2.81	10.11	1.56	2.70	5.44	1.11
		$R = 10$	14.97	57.69	8.04	2.86	9.13	1.53	1.07	3.26	0.55
		$R = 20$	14.46	63.58	7.85	3.85	11.27	1.88	2.54	5.66	1.07
		$R = 40$	14.63	84.33	8.12	3.93	11.85	1.95	3.15	6.74	1.46
	Plan	$(8, -13, 70)$	11.52	49.75	5.68	3.15	7.72	1.15	3.00	4.06	0.32
		$(3, 4, -5)$	10.54	42.16	5.89	2.47	7.19	1.20	1.87	3.57	0.46

Tableau 4.3 – Différences angulaires δ (en degrés $^\circ$) entre le vecteur normal théorique et celui provenant de l'estimateur proposé dans [TW97], sur le bord de différents plans et sphères discrets.

Approche			Ombrelles d'ordre n								
Ordre de l'ombrelle (n)			1			3			6		
État	Objet testé		δ_{moy}	δ_{max}	σ	δ_{moy}	δ_{max}	σ	δ_{moy}	δ_{max}	σ
Non bruité	Sphère	$R = 5$	6.89	13.26	3.39	3.40	8.13	2.55	2.19	5.16	1.63
		$R = 10$	6.64	16.84	3.67	2.16	6.44	1.35	1.80	4.80	1.08
		$R = 20$	7.26	19.28	3.80	2.25	8.66	1.63	1.07	3.19	0.66
		$R = 40$	7.66	23.10	4.15	2.39	8.15	1.47	0.97	3.66	0.62
	Plan	$(8, -13, 70)$	9.94	20.51	4.78	1.39	3.25	0.85	0.68	1.10	0.27
		$(3, 4, -5)$	5.20	8.13	2.22	2.53	5.36	1.73	0.89	1.51	0.43
Faiblement bruité	Sphère	$R = 5$	13.21	42.03	7.12	3.95	11.18	2.11	3.12	10.40	1.76
		$R = 10$	13.65	48.49	7.35	3.75	12.44	2.11	2.54	8.99	1.32
		$R = 20$	13.16	52.04	7.38	3.51	13.22	1.94	1.45	5.51	0.80
		$R = 40$	13.48	54.30	7.50	3.53	21.61	2.00	1.32	5.67	0.73
	Plan	$(8, -13, 70)$	11.24	35.95	5.54	2.03	7.71	1.12	0.79	2.76	0.40
		$(3, 4, -5)$	8.49	26.41	4.91	2.79	11.17	1.63	1.01	3.72	0.54
Fortement bruité	Sphère	$R = 5$	16.36	43.24	8.32	4.70	17.25	2.68	3.43	11.69	1.93
		$R = 10$	16.01	56.34	8.76	4.34	17.19	2.34	2.77	9.29	1.42
		$R = 20$	15.30	62.93	8.67	4.18	22.50	2.38	1.75	7.51	0.97
		$R = 40$	15.42	73.76	8.84	4.07	20.31	2.28	1.51	6.22	0.84
	Plan	$(8, -13, 70)$	12.68	54.76	6.57	2.50	9.21	1.37	0.96	3.44	0.51
		$(3, 4, -5)$	11.40	54.74	6.43	3.19	13.01	1.79	1.18	4.34	0.63

Tableau 4.4 – Différences angulaires δ (en degrés $^\circ$) entre le vecteur normal théorique et celui provenant de l'estimateur proposé dans [PF98], sur le bord de différents plans et sphères discrets.

Dans le cas des patchs discrets, nous nous apercevons d'ailleurs du caractère adaptatif de l'estimateur. En effet, pour une épaisseur fixée, plus le rayon de la sphère augmente, plus le voisinage relatif au patch s'étend. Les cas où l'objet considéré est un morceau de plan discret viennent d'ailleurs confirmer ce comportement puisque dans le cas où ce dernier est régulier, la normale en chaque point est parfaitement estimée, et nous retrouvons les caractéristiques du plan discret arithmétique initial (pour peu que le morceau généré contienne certains points d'appuis du plan discret). Dans les cas bruités, le comportement est similaire lorsque l'épaisseur est adaptée. Pour des morceaux fortement bruités et une épaisseur trop petite l'estimation n'est pas parfaite mais cela est logique puisque nous considérons une *planarité relative à un facteur ν* , facteur permettant de contrôler l'estimation en fonction des déformations locales du bord de l'objet. Ces résultats sont donc à opposer à ceux obtenus avec les deux autres approches, où la taille de voisinage est fixée a priori et ne permet généralement pas d'obtenir cette estimation exacte.

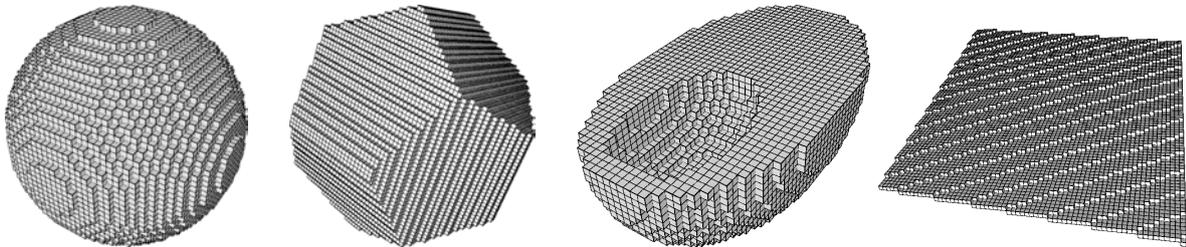
Ces résultats sont à nuancer par le fait que lorsque nous travaillons sur le bord d'un objet discret, les morceaux de plans (caractérisant les facettes planes de l'objet) sont délimités par les arêtes naturelles de l'objet. Or un patch développé en un point d'une telle facette sera perturbé par les points au voisinage de ces arêtes naturelles, puisque quelques-uns d'entre eux seront potentiellement inclus dans le patch, faisant « pencher » le MFPD sous-jacent et perturbant ainsi l'estimation. Ce comportement peut être observé sur certains objets des figures 4.9 et 4.10. Comme l'analyse des statistiques ne permet pas d'appréhender totalement le comportement des estimateurs, nous proposons également des résultats sous forme *visuelle* de différents objets discrets, bruités et non bruités, en affectant à chaque voxel du bord de l'objet sa normale estimée et en effectuant un rendu classique sous OpenGL. Comme nous pouvons le constater, les artefacts de rendu qui apparaissent sur les plans discrets en utilisant un estimateur à voisinage de taille fixe, disparaissent lorsque nous utilisons l'approche par patchs discrets, puisque tous les points du morceau de plan sont considérés pour l'estimation de la normale. Par contre, ce comportement moins local qu'avec les approches par voisinage de taille fixe est plus sensible aux arêtes naturelles de l'objet, comme on peut le constater sur les zones plates de l'ellipsoïde creusé de la figure 4.9(d) et du dodécaèdre bruité à la figure 4.10(d). Certaines normales sont issues d'un patch ayant subi l'influence de points proches des discontinuités naturelles de l'objet, et des défauts apparaissent dans le rendu des parties planes des objets. De manière générale, des perturbations locales apparaissant au bord de l'objet au niveau d'une *partie* d'une zone plate ont potentiellement une influence sur l'estimation des normales de *toute* cette zone, lorsque nous utilisons les patchs discrets. Au contraire, le rendu utilisant des vecteurs normaux issus des estimateurs à voisinage de taille fixée est moins sensible à ces cas de figures.

Chaque estimateur présente donc ses avantages et ses inconvénients et généralement le contexte pratique d'utilisation permettra de faire un choix approprié. Ceci étant, les informations géométriques portées par un patch discret d'épaisseur ν ne se résument pas seulement à un vecteur normal. Nous allons notamment voir, dans la section suivante, comment exploiter l'adaptabilité des patchs et leur sensibilité aux discontinuités naturelles de l'objet pour mettre au point un estimateur de forme distinguant les zones plates, bombées et creuses du bord d'un objet discret.

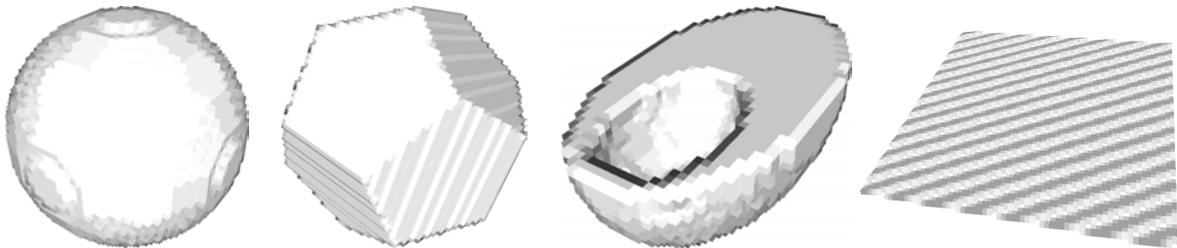
4.3 Estimateur de forme

Nous présentons dans cette section un estimateur de forme permettant de caractériser la conformation locale du bord d'un objet discret en distinguant les zones plates, les creux et les bosses. Je rappelle que l'idée est d'utiliser cet estimateur de forme pour diriger le processus de segmentation du bord d'un objet discret en morceaux flous de plans discrets, en donnant une priorité plus élevée aux zones estimées « plates ».

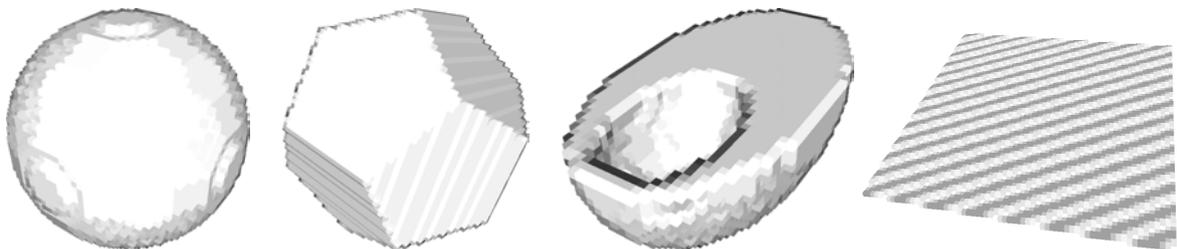
La caractéristique géométrique généralement utilisée pour discriminer ce type de zone est la courbure. Si dans le cas 2D cette notion ne prête pas à confusion, en 3D les choses sont différentes, et plusieurs notions de courbures (courbure moyenne, courbure gaussienne, etc...) existent. Nous rappelons dans un premier temps comment sont définies ces notions dans le cadre de la géométrie continue, puis nous verrons ensuite comment elles ont été adaptées dans le cadre de la géométrie discrète.



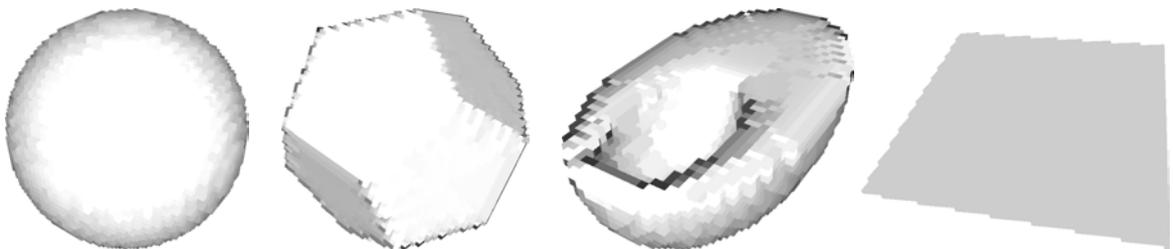
(a) Des objets discrets réguliers : une sphère, un dodécaèdre, un demi-ellipsoïde creusé et un morceau de plan



(b) Estimation des normales en suivant l'approche [TW97] avec un voisinage $ns = 1$

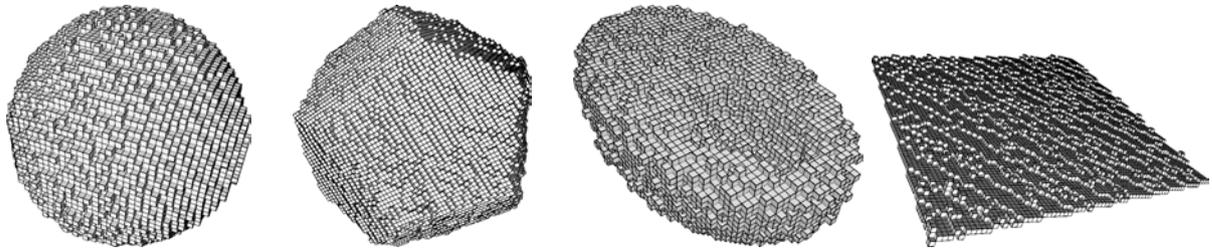


(c) Estimation des normales en suivant l'approche [PF98] avec une ombrelle d'ordre $n = 1$

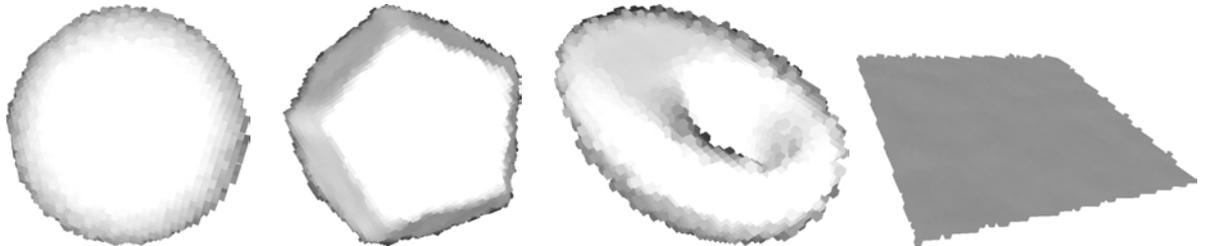


(d) Estimation des normales d'épaisseur 1 en suivant l'approche par patches

Figure 4.9 – Résultats de l'estimation du champ de vecteurs normaux au bord d'objets discrets réguliers (a). Le rendu sous OpenGL pour (b)-(d) est effectué en utilisant, pour chaque voxel du bord, le vecteur normal estimé en ce dernier.



(a) Des objets discrets bruités : une sphère, un dodécaèdre, un demi-ellipsoïde creusé et un morceau de plan



(b) Estimation des normales en suivant l'approche [TW97] avec un voisinage $ns = 6$



(c) Estimation des normales en suivant l'approche [PF98] avec une ombrelle d'ordre $n = 6$



(d) Estimation des normales d'épaisseur 3 en suivant l'approche par patches

Figure 4.10 – Résultats de l'estimation du champ de vecteurs normaux au bord des objets discrets bruités (a). Le rendu sous OpenGL pour (b)-(d) est effectué en utilisant, pour chaque voxel du bord, le vecteur normal estimé en ce dernier.

4.3.1 Notion de courbure en géométrie différentielle

Nous commençons par rappeler les définitions pour une courbe plane de \mathbb{R}^2 . Pour une introduction complète à la géométrie différentielle on pourra consulter [Spi99, GB92].

Soit donc une courbe \mathcal{C} de classe \mathcal{C}^2 , supposée régulière, et paramétrée par son abscisse curviligne s

$$\begin{aligned} \mathbf{M} : I \subset \mathbb{R} &\rightarrow \mathbb{R}^2 \\ s &\mapsto \mathbf{M}(s) \end{aligned}$$

En un point d'abscisse s de la courbe, le vecteur tangent $\mathbf{t}(s)$ est défini par

$$\mathbf{t}(s) = \frac{d\mathbf{M}}{ds}(s)$$

et forme avec un vecteur unitaire perpendiculaire $\mathbf{n}(s)$, appelé vecteur normal, une base orthonormée. On appelle alors *courbure algébrique* au point d'abscisse s la valeur $\gamma(s)$ donnée par

$$\frac{d\mathbf{t}}{ds}(s) = \gamma(s) \cdot \mathbf{n}(s)$$

Une autre définition, équivalente, peut être donnée grâce à la notion de cercle osculateur. Ainsi, si l'on considère un point M_0 sur la courbe, il existe, pour tout point M de la courbe suffisamment proche de M_0 , un unique cercle \mathcal{C}_M tangent à la courbe en M_0 et passant par M . On appelle alors *cercle osculateur à la courbe en M_0* la limite de \mathcal{C}_M lorsque M tend vers M_0 sur la courbe (figure 4.11(a)), et on définit la courbure en M_0 comme étant égale à l'inverse de rayon R_{M_0} du cercle osculateur en M_0 : $\gamma_{M_0} = \frac{1}{R_{M_0}}$. Tout comme dans le cas précédent, cette valeur peut être signée en fonction de l'orientation choisie pour le vecteur normal \mathbf{n} .

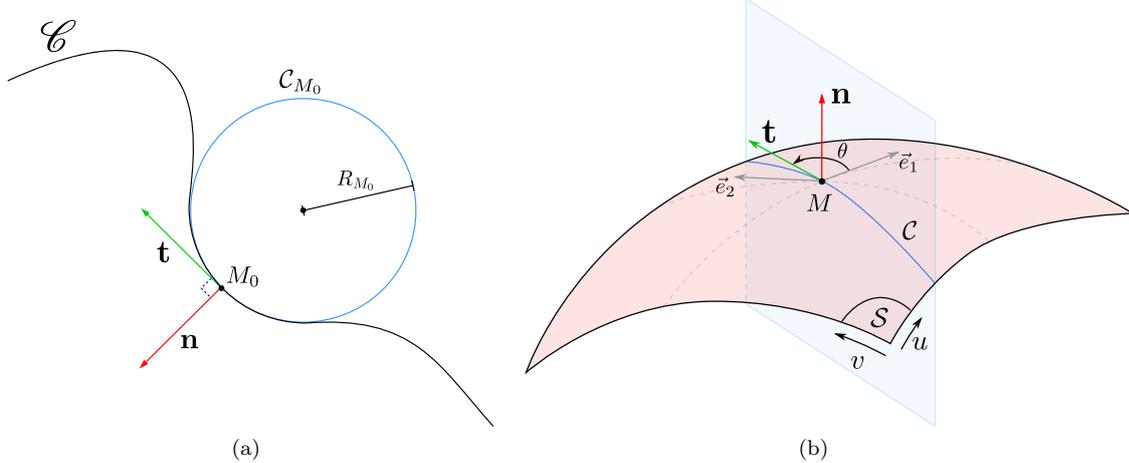


Figure 4.11 – (a) Une courbe et le cercle osculateur à cette courbe en un point M_0 . (b) Courbure normale et directions de courbure principales sur une surface.

Nous allons à présent introduire les différentes notions de courbures au niveau des surfaces de \mathbb{R}^3 . Pour cela, nous allons considérer une surface paramétrée \mathcal{S} de classe \mathcal{C}^2 , supposée régulière et orientée

$$\begin{aligned} \mathbf{M} : U \times V \subset \mathbb{R}^2 &\rightarrow \mathbb{R}^3 \\ (u, v) &\mapsto \mathbf{M}(u, v) \end{aligned}$$

Nous notons $\mathbf{n}(u, v)$ le vecteur normal unitaire à la surface au point $\mathbf{M}(u, v)$. Si l'on considère à présent un plan \mathcal{P} contenant le vecteur normal $\mathbf{n}(u, v)$, alors \mathcal{P} coupe la surface \mathcal{S} en décrivant une courbe plane \mathcal{C} et la courbure algébrique de cette courbe, appelée *courbure normale*, peut être calculée

comme énoncé précédemment. De plus, lorsque le plan \mathcal{P} effectue une révolution autour de l'axe porté par la normale $\mathbf{n}(u, v)$ au point $\mathbf{M}(u, v)$, alors \mathcal{P} coupe la surface \mathcal{S} en un ensemble de courbes planaires. Parmi celles-ci, deux, de vecteurs tangents \vec{e}_1 et \vec{e}_2 , sont associées aux valeurs minimale et maximale de la courbure normale en $\mathbf{M}(u, v)$. Ces valeurs de courbure extrêmes sont notées λ_1 et λ_2 et sont appelées *courbures principales* en $\mathbf{M}(u, v)$. Les vecteurs \vec{e}_1 et \vec{e}_2 représentent les *directions principales de courbure* et ils sont orthogonaux. Ces courbures principales permettent de définir la notion de *courbure gaussienne*, notée K et telle que $K = \lambda_1 \cdot \lambda_2$, ainsi que la notion de *courbure moyenne*, noté H et telle que $H = \frac{\lambda_1 + \lambda_2}{2}$. La connaissance de ces deux courbures permet de caractériser les configurations locales possibles de la surface au point $\mathbf{M}(u, v)$.

De plus, si l'on considère une courbe \mathcal{C} passant par $\mathbf{M}(u, v)$ et dont le vecteur tangent $\mathbf{t}(u, v)$ fait un angle θ avec la première direction de courbure \vec{e}_1 (figure 4.11(b)), alors la courbure normale γ de \mathcal{C} en $\mathbf{M}(u, v)$ peut s'exprimer en fonction des courbures principales grâce à la formule d'Euler :

$$\gamma = \lambda_1 \cdot \cos^2(\theta) + \lambda_2 \cdot \sin^2(\theta) \quad (4.1)$$

En se basant sur ces notions de géométrie différentielle, plusieurs méthodes ont été développées pour permettre d'estimer la courbure dans le cadre de la géométrie discrète.

4.3.2 Estimation de la courbure sur des surfaces discrètes

Une première approche permet d'estimer les courbures principales par des techniques de filtrage (convolution d'une image avec un filtre). On touche ici plus au domaine du traitement de l'image qu'à la géométrie discrète telle que présentée dans le premier chapitre. On pourra donc se référer à [HM95] pour une explication détaillée des notions abordées ci-dessous.

L'objet est représenté à travers une image $I(x, y, z)$ (en niveaux de gris), et l'on considère une surface Σ définie comme une iso-surface de $I(x, y, z)$. Des filtrages de I à l'aide de filtres dérivatifs permettent d'obtenir les dérivées premières I_k , $k \in \{x, y, z\}$ et secondes I_{kl} , $k, l \in \{x, y, z\}$ de $I(x, y, z)$, et le gradient \mathbf{g} ainsi que la hessienne \mathcal{H} peuvent alors être déterminés

$$\mathbf{g} = \begin{pmatrix} I_x \\ I_y \\ I_z \end{pmatrix} \quad \mathcal{H} = \begin{pmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{pmatrix}$$

Soit à présent \mathbf{t} , un vecteur unitaire du plan tangent \mathcal{T}_M au point M de Σ , et $\mathbf{n} = \frac{\mathbf{g}}{\|\mathbf{g}\|}$ la normale à Σ en M . En utilisant les formalismes de géométrie différentielle exposés ci-dessus, on peut montrer que la courbure $\gamma_{\mathbf{t}}$ dans la direction \mathbf{t} est donnée par la formule

$$\gamma_{\mathbf{t}} = -\frac{{}^t\mathbf{t} \mathcal{H} \mathbf{t}}{\|\mathbf{g}\|}$$

Les courbures principales et les directions principales de courbure de Σ au point M peuvent alors être obtenues en cherchant les directions \mathbf{t} pour lesquelles $\gamma_{\mathbf{t}}$ est un extremum. Si (\mathbf{a}, \mathbf{b}) est une base orthonormée du plan tangent \mathcal{T}_M , on peut écrire $\mathbf{t} = \cos(\theta)\mathbf{a} + \sin(\theta)\mathbf{b}$, et dériver $\gamma_{\mathbf{t}}$ par rapport à θ pour avoir

$$-\|\mathbf{g}\| \frac{d\gamma_{\mathbf{t}}}{d\theta} = \sin(2\theta)({}^t\mathbf{b} \mathcal{H} \mathbf{b} - {}^t\mathbf{a} \mathcal{H} \mathbf{a}) + 2 \cos(2\theta)({}^t\mathbf{a} \mathcal{H} \mathbf{b})$$

\mathbf{t} est alors une direction principale de courbure lorsque cette dérivée s'annule, c'est-à-dire lorsque

$$\tan(2\theta) = \frac{{}^t\mathbf{a} \mathcal{H} \mathbf{b}}{{}^t\mathbf{b} \mathcal{H} \mathbf{b} - {}^t\mathbf{a} \mathcal{H} \mathbf{a}}$$

pour peu que le dénominateur ne soit pas nul. Comme ce dernier n'est nul que dans le cas d'un point ombilic, cela ne pose pas de problème puisqu'alors la courbure est identique dans toutes les directions.

Une fois les directions principales de courbure obtenues, les courbures principales associées nous permettent de calculer la courbure gaussienne ainsi que la courbure moyenne.

Évidemment les résultats de cette approche restent dépendants des filtres utilisés pour l'estimation des dérivées et de la manière de les implémenter (voir [HM95] pour une discussion). En général, une étape de lissage est associée à la dérivation (opérateur dérivateur-lisseur), afin de limiter la sensibilité de l'opérateur aux perturbations, rendant cette méthode applicable sur des objets bruités.

Estimateur de courbure gaussienne

Si nous nous intéressons uniquement au calcul de la courbure gaussienne, alors D. Coeurjolly propose dans [Coe02b] un estimateur de courbure utilisant les propriétés de l'application de Gauss. Cette dernière, notée \mathcal{G} , associe à tout point d'une surface \mathcal{S} orientable et \mathcal{C}^1 , son vecteur normal unitaire orienté. En s'inspirant de la relation de géométrie différentielle suivante définissant la courbure gaussienne en un point m de \mathcal{S}

$$K(m) = \lim_{u \rightarrow m} \frac{\text{Aire}(\mathcal{G}(u))}{\text{Aire}(u)}$$

où u est un voisinage ouvert de \mathcal{S} contenant le point m , l'auteur dérive un estimateur utilisant des outils de géométrie discrète. Plus précisément, en un surfel s d'une surface discrète Σ pour laquelle la normale a été estimée en chaque surfel, et un voisinage géodésique V_d de Σ de rayon d autour de s , l'estimateur suivant est proposé

$$K^*(s) = \frac{\text{AIRE_SPHERE}(\mathcal{G}(V_d))}{E_{\mathcal{A}}(V_d)}$$

$E_{\mathcal{A}}(V_d)$ est un estimateur d'aire (décrit plus loin à la section 4.3.4.2) utilisant la normale estimée en chaque surfel pour exprimer l'aire du voisinage V_d de Σ , et AIRE_SPHERE est un procédé permettant d'estimer l'aire du nuage de points obtenu en appliquant \mathcal{G} au surfel de V_d .

Deux options sont considérées pour mettre en œuvre AIRE_SPHERE . La première consiste à estimer l'aire en utilisant un calcul d'enveloppe convexe sphérique – une généralisation de la notion d'enveloppe convexe où les côtés (arêtes) de l'enveloppe sont remplacés par des arcs de grands cercles de la sphère unité – des points de $\mathcal{G}(V_d)$. En supposant que le polygone convexe sphérique obtenu ait n sommets et que la somme des angles au niveau des sommets du polygone soit égale à θ , alors l'aire de ce dernier représente l'estimation de $\mathcal{G}(V_d)$ recherchée, d'où

$$\text{AIRE_SPHERE}(\mathcal{G}(V_d)) = \theta - (n - 2)\pi$$

L'enveloppe convexe sphérique est obtenue en généralisant l'algorithme d'enveloppe convexe de Graham [Gra72] à l'aide d'un tri polaire sphérique et d'un prédicat de type « *left turn* » adapté aux sphères.

L'autre approche utilise la notion d'*ellipse de meilleure approximation*. Dans un premier temps le barycentre \mathbf{b} du nuage de points $\mathcal{G}(V_d)$ sur la sphère unité est calculé, puis l'ensemble de ces points est projeté sur le plan tangent en \mathbf{b} à la sphère unité. Dès lors, les paramètres de l'ellipse de meilleure approximation des projetés $\{(x_i, y_i)\}_{1 \leq i \leq n}$ des points peuvent être calculés à partir des valeurs propres de la matrice \mathcal{M} (appelée matrice d'inertie) définie de la façon suivante

$$\mathcal{M} = \begin{pmatrix} m_{xx} & m_{xy} \\ m_{yx} & m_{yy} \end{pmatrix} \quad \text{avec} \quad \begin{aligned} m_{xx} &= \frac{1}{n} \sum_{k=1}^n (x_k - \mu_x)^2 \\ m_{yy} &= \frac{1}{n} \sum_{k=1}^n (y_k - \mu_y)^2 \\ m_{yx} &= m_{xy} = \frac{1}{n} \sum_{k=1}^n (x_k - \mu_x)(y_k - \mu_y) \end{aligned} \quad \text{et} \quad \begin{aligned} \mu_x &= \frac{1}{n} \sum_{k=1}^n x_k \\ \mu_y &= \frac{1}{n} \sum_{k=1}^n y_k \end{aligned}$$

Ainsi, si l'on note λ_1 et λ_2 les valeurs propres de \mathcal{M} , l'ellipse de meilleure approximation est l'ellipse de centre (μ_x, μ_y) et dont les axes, portés par les vecteurs propres de \mathcal{M} , sont de longueur $\sqrt{\lambda_1}$ et $\sqrt{\lambda_2}$. L'aire de cette ellipse est alors utilisée comme approximation de l'aire de $\mathcal{G}(V_d)$, ce qui nous donne

$$\text{AIRE_SPHERE}(\mathcal{G}(V_d)) = \pi \sqrt{\lambda_1 \lambda_2}$$

Les deux méthodes présentées ci-dessous nécessitent néanmoins, pour être valides, que l'ensemble des points $\mathcal{G}(V_d)$ se situe dans un même hémisphère de la sphère unité.

Les quelques tests proposés dans [Coe02b] donnent des résultats visuels satisfaisants sur des objets discrets mathématiques réguliers (cylindre et cube) mais ne permettent pas de conclusion plus poussée. Néanmoins, il paraît évident qu'une bonne estimation du champ de normales à la surface de l'objet discret est indispensable pour assurer le bon comportement de ces approches. Une autre question concerne le choix de d , la taille du voisinage géodésique V_d considéré. Comment fixer ce paramètre pour effectuer au mieux l'estimation, et particulièrement dans le cas d'objets discrets bruités? Nous pourrions envisager de remplacer ce voisinage géodésique par un patch discret d'épaisseur ν , mais cette idée ne semble pas judicieuse, puisqu'un patch $\Gamma_{\nu,p}$ représente une zone plate (relativement à ν) autour d'un point p . En considérant les normales de cette zone, nous limiterions a priori l'amplitude de la courbure, ce qui n'est pas désirable.

Estimateur de courbure moyenne

Si nous nous intéressons à présent à la notion de courbure moyenne, il existe dans la littérature différentes approches applicables sur des objets discrets voxels.

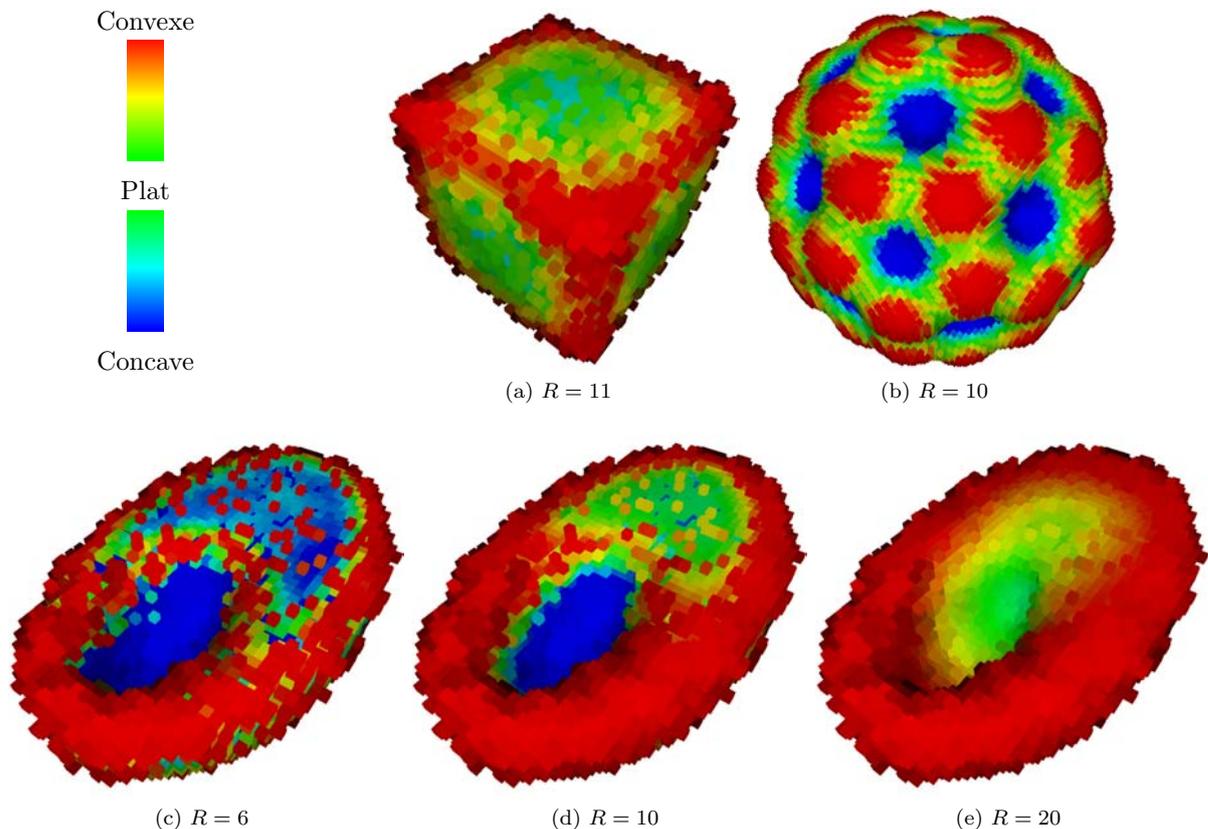


Figure 4.12 – Estimation de la courbure moyenne, en utilisant l'estimateur proposé dans [BGCF95], au bord (a) d'un cube discret bruité et (b) d'une molécule de fullerène, en utilisant des rayons de sphères « qui vont bien » (fixés de manière empirique). (c)-(e) Estimation de la courbure moyenne au bord d'une demi-ellipsoïde creusée bruitée en faisant varier le rayon R de la sphère.

Une première [BGCF95], proposée par J.W. Bullard et ses coauteurs, repose sur l'existence d'une relation de linéarité entre la courbure moyenne et le volume d'une portion de sphère centrée à la surface d'un objet. Plus précisément, nous nous plaçons dans l'espace continu et nous considérons une surface orientée \mathcal{S} , de classe C^2 . Étant donnée une sphère B de rayon R centrée en un point p de \mathcal{S} , nous notons

V le volume inclus dans B qui se situe d'un côté de \mathcal{S} . Les auteurs montrent que la courbure moyenne H en p peut être approximée par la formule

$$H \approx \frac{4}{\pi R^4} V - \frac{8}{3R}$$

Pour une utilisation sur des objets discrets, les auteurs proposent d'approximer la courbure moyenne en un point p du bord de l'objet en utilisant la formule précédente et en choisissant pour V le nombre de voxels de l'objet qui se situent à l'intérieur d'une sphère de rayon R centrée en p . Comme cette méthode est relativement simple à programmer et que la sensibilité au bruit est en quelque sorte contrôlable grâce au rayon de la sphère, j'ai implémenté la formule et quelques résultats sont présentés à la figure 4.12.

Comme nous pouvons l'observer, cet estimateur permet de différencier les parties concaves, convexes et plates d'un objet discret, même bruité, sous réserve que le rayon de la sphère soit bien choisi. En effet, avec un rayon trop petit l'estimation reste très sensible au bruit, et avec un rayon trop grand, l'estimateur a tendance à « lisser » la courbure, du fait du voisinage trop grand considéré. Un autre problème est que la sphère est susceptible d'englober des parties de l'objet non désirées, comme illustré à figure 4.13. Si le cas de la figure 4.13(a) peut être évité en ne considérant que la composante connexe, incluse dans la sphère, qui contient le point où l'on estime la courbure, pour le cas de la figure 4.13(b) c'est plus difficile ; ces problèmes apparaissent surtout à cause du caractère non adaptatif de la sphère, c'est-à-dire du voisinage volumique fixe considéré.

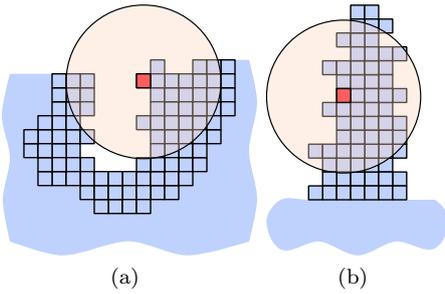


Figure 4.13 – Cas problématiques pouvant apparaître lors de l'estimation de la courbure moyenne au voxel rouge, avec l'estimateur [BGCF95]. Les figures sont 2D, mais sont à interpréter comme une coupe d'un objet 3D projetée sur un plan. L'estimation peut être substantiellement perturbée car dans le cas (a) la sphère rouge contient deux composantes connexes distinctes appartenant à l'objet et dans le cas (b) la sphère « traverse » de part et d'autre de l'objet.

Une seconde approche, exploitant les contours des tranches d'un objet discret, a été proposée par A. Lenoir dans [Len97, Len99]. À partir de la formule d'Euler (équation 4.1) et de trois courbes planaires de la surface de l'objet qui passent par un point p , l'auteur montre que la courbure moyenne H en p peut s'écrire comme une combinaison linéaire des courbures en p de ces trois courbes planaires. En particulier, lorsque les trois courbes planaires C_i sont des contours des tranches d'indice i ($i = 1, 2, 3$) qui contiennent p , alors la courbure moyenne s'écrit

$$H = \sum_{i=1}^3 \frac{(1 - n_i^2)^{\frac{3}{2}}}{2} \cdot c_i \quad (4.2)$$

où n_i représente la i^{e} composante du vecteur normal \mathbf{n} estimé à la surface de l'objet en p et c_i sont les courbures en p des contours des tranches d'indices i de l'objet.

La normale \mathbf{n} en chaque surfel de la surface de l'objet est estimée en utilisant la technique décrite en 4.2.1.1 qui repose sur l'estimation des tangentes en ces surfels. Et conformément aux formules de géométrie différentielle, les courbures normales des contours de tranches sont estimées en appliquant un filtre dérivatif sur les tangentes estimées le long de ces contours. Néanmoins, pour pouvoir utiliser la formule 4.2, trois contours de tranches sont nécessaires pour approximer la courbure moyenne en un surfel s de la surface de l'objet discret. Or, en ce même surfel ne passent que deux contours de tranches. L'idée de l'auteur est alors d'estimer statistiquement la courbure du contour manquant, disons issu d'une tranche d'indice 1, pour fixer les idées, en allant chercher l'information sur des surfels du voisinage de s dans les contours des tranches d'indices 2 et 3. Ainsi, si pour ces derniers nous notons $\pi_1(s)$ la proportion de surfels au voisinage de s intersectés par un contour de tranche d'indice 1, alors, à condition que π_1 reste presque constant

sur les surfels au voisinage de s et que la courbure normale des contours des tranches en ces mêmes surfels reste aussi presque constante, par moyennage de ces dernières nous obtenons la courbure du contour de tranche d'indice i en s . Nous pouvons alors appliquer la formule 4.2 pour calculer la courbure moyenne.

Comme cette technique repose sur des calculs dans les contours des tranches de l'objet, nous avons essayé de l'adapter en utilisant des primitives floues, afin de la tester sur des objets bruités. C'est ce que nous décrivons dans la section suivante.

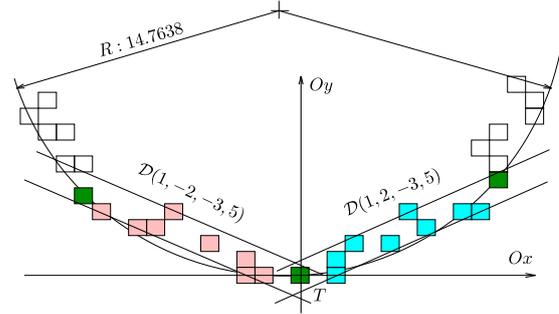
4.3.3 Approche par tranches

Nous allons utiliser une approche similaire à celle de la section 4.2.1.2, en calculant dans les trois contours de tranches de l'objet discret, les données nécessaires à l'application de la formule 4.2. Grâce à l'algorithme 4.1, nous avons déjà une estimation du vecteur normal en tout point du bord de l'objet¹⁰. Il ne nous reste qu'à déterminer la courbure en chacun de ces mêmes points. Pour cela nous utilisons une technique proposée dans [NDR07], qui se base sur le calcul de la couverture tangentielle en segments flous maximaux d'épaisseur ν afin d'estimer le rayon du cercle osculateur en un point d'une courbe discrète. Plus précisément, si nous reprenons les notations de la section 4.2.1.2 à la page 78, nous avons :

Proposition 4.10 ([NDR07]) *Appelons $L(k)$ et $R(k)$ les fonctions qui représentent respectivement les indices des extrémités gauches et droites des segments flous maximaux d'épaisseur ν du côté gauche et du côté droit du point C_k d'une courbe discrète C . Alors nous avons :*

- $\forall k$ tel que $E_{i-1} < k \leq E_i$, alors $L(k) = B_i$
- $\forall k$ tel que $B_i \leq k < B_{i+1}$, alors $R(k) = E_i$

La méthode pour calculer la courbure à l'épaisseur ν au point C_k consiste alors à déterminer $L(k)$ et $R(k)$ et à calculer le rayon r du cercle circonscrit aux points $C_{L(k)}, C_k, C_{R(k)}$. La valeur de la courbure est alors égale à l'inverse de r et son signe est déterminé par l'orientation de la courbe C .



Si nous désirons estimer la courbure en un voxel v du bord d'un objet discret, deux cas peuvent se présenter : soit trois contours de tranche passent par v , soit seulement deux contours de tranche passent par v . Dans le premier cas, nous estimons la normale au voxel ainsi que les courbures des trois contours de tranches et nous disposons de toutes les informations nécessaires pour appliquer la formule 4.2.

Dans l'autre cas, nous recherchons dans le voisinage de v , un voxel par lequel passe un contour de tranche de même indice que celui manquant en v et nous l'utilisons pour extraire la courbure manquante pour appliquer la formule 4.2.

Les résultats obtenus avec cette méthode sont illustrés à la figure 4.14

Le choix du contour de tranche d'indice manquant est ici discutable. En effet, les perturbations du bord de l'objet induisent les mêmes effets que ceux constatés à la section 4.2.1.2, à savoir l'apparition de tranches non fiables. Si la courbure du contour de tranche manquant est estimée en se basant sur un tel contour, alors les résultats risquent d'être faussés. Nous aurions pu effectuer, tout comme dans [Len97], un moyennage des courbures de tous les contours de tranche d'indice manquant, au voisinage de v , pour atténuer les effets des perturbations. Nous avons cependant préféré adopter une méthode utilisant les patches discrets d'épaisseur ν , qui nous a conduit à définir un estimateur de forme, décrit ci-après et utilisé dans le cadre de la décomposition du bord d'un objet discret bruité en $MFPD_\nu$ pour guider le processus de segmentation.

¹⁰. Nous pouvons en fait utiliser n'importe quel estimateur de vecteur normal pour déterminer \mathbf{n} , mais en utilisant celui par tranches à l'épaisseur ν , nous pouvons fusionner l'estimation de la normale et de la courbure, évitant ainsi de calculer plusieurs fois la couverture tangentielle.

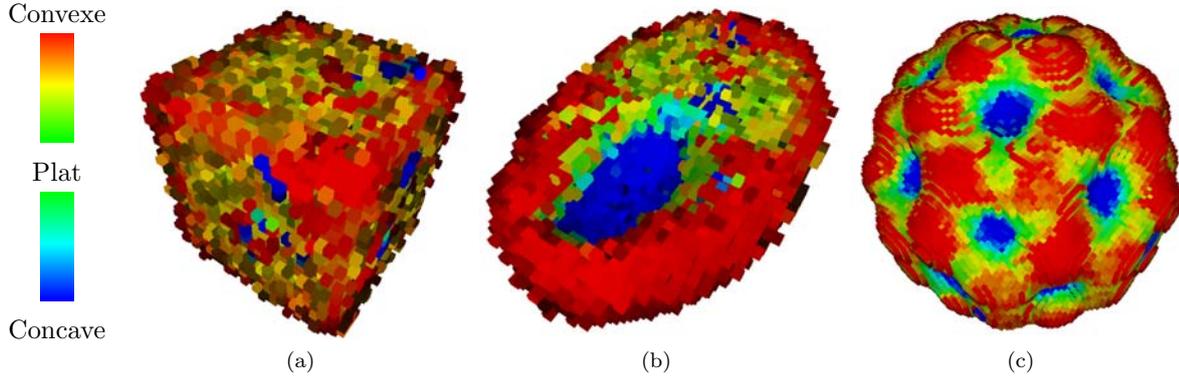


Figure 4.14 – Estimation de la courbure moyenne, en utilisant une approche par tranches à l'épaisseur ν , au bord (a) d'un cube discret bruité, (b) d'un demi-ellipsoïde creusé bruité et (c) d'une molécule de fullerène.

4.3.4 Approche par patchs discrets d'épaisseur ν

Au vu des résultats non satisfaisants obtenus avec l'estimateur exploitant les tranches de l'objet, nous avons proposé un autre estimateur de forme exploitant les informations portées par des patchs discrets développés au bord de l'objet discret. Nous n'utilisons pas ici la notion de courbure, mais nous nous en inspirons en exploitant les variations des directions des vecteurs normaux sur le bord de l'objet discret.

4.3.4.1 Intuition

Un patch $\Gamma_\nu(p)$, comme nous l'avons défini auparavant, permet de caractériser la planarité du bord de l'objet discret autour de p , respectivement à l'épaisseur ν considérée. En effet, plus le patch est étendu, plus la surface autour de p peut être considérée comme plate, puisqu'appartenant à un morceau flou de plan discret. La « taille » de $\Gamma_\nu(p)$ peut donc nous apporter une information utile pour mettre au point notre estimateur.

D'un autre côté, si l'extension de $\Gamma_\nu(p)$ s'est arrêtée, cela signifie que les points restants, au voisinage du patch, n'ont pas pu être ajoutés car ils auraient engendré une perturbation trop importante au niveau de $\Gamma_\nu(p)$ pour que ce dernier soit toujours considéré comme un MFPD_ν , et donc jugé comme étant une zone plate. Ainsi, il semble possible d'extraire des informations des zones autour de $\Gamma_\nu(p)$ pour pouvoir en déduire ensuite une conformation de la surface autour du point p .

Il nous reste à quantifier ces observations. En ce qui concerne la notion de « taille » d'un patch, nous avons choisi de nous reposer sur l'estimation de son aire, et pouvons l'interpréter comme un coefficient de fiabilité associé à l'information « être une zone plane ». Pour les zones avoisinantes de $\Gamma_\nu(p)$, nous avons décidé d'exploiter les informations portées par les patchs centrés en des points du contour de $\Gamma_\nu(p)$. Ces notions et les solutions envisagées sont présentées dans les sections suivantes.

4.3.4.2 Aire d'un patch discret d'épaisseur ν

Afin de présenter la façon dont l'aire du patch est calculée, nous reprenons les notations utilisées dans [CFTT03].

Soit \mathcal{S} une surface euclidienne et $\{\vec{n}\}$ le champ de vecteurs normaux qui lui est associé. Une relation bien connue en mathématiques nous permet de calculer l'aire de la surface en intégrant sur \mathcal{S} le champ de vecteurs normaux,

$$\mathcal{A}(\mathcal{S}) = \int_{\mathcal{S}} \vec{n}(s) ds$$

Si maintenant nous considérons une discrétisation $D(\mathcal{S})$ de \mathcal{S} , nous pouvons remplacer l'intégrale sur \mathcal{S} par une somme finie sur les surfels de $D(\mathcal{S})$, le vecteur $\vec{n}(s)$ par une estimation du vecteur normal au surfel s et ds par le produit scalaire de cette dernière avec le vecteur unitaire orthogonal au surfel s pointant vers l'extérieur. L'idée derrière cette formule, initialement proposée en 3D par A. Lenoir dans [LMR96], est de calculer la contribution de chaque surfel à l'aire global de $D(\mathcal{S})$ en projetant le surfel selon la direction de sa normale estimée. Ainsi, nous avons une version discrète de la formule précédente, à savoir,

$$E_{\mathcal{A}}(D(\mathcal{S})) = \sum_{s \in D(\mathcal{S})} \vec{n}^*(s) \cdot \vec{n}_{el}(s)$$

où $\vec{n}^*(s)$ représente le vecteur normal associé à un élément de surface s dans $D(\mathcal{S})$ et $\vec{n}_{el}(s)$ est le vecteur unitaire orthogonal au surfel s .

Dans notre cas, nous pouvons choisir pour $\vec{n}^*(s)$ un des estimateurs de vecteur normal proposés à la section précédente. Nous avons choisi de retenir le vecteur $\vec{n}_\nu(p)$ (définition 4.9), ce qui nous amène à l'estimateur d'aire pour un patch discret $\Gamma_\nu(p)$:

$$E_{\mathcal{A}}(\Gamma_\nu(p)) = \sum_{s \in \mathcal{S}_{\Gamma_\nu(p)}} \vec{n}_\nu(p) \cdot \vec{n}_{el}(s) = \vec{n}_\nu(p) \cdot \sum_{s \in \mathcal{S}_{\Gamma_\nu(p)}} \vec{n}_{el}(s)$$

4.3.4.3 Contour d'un patch discret d'épaisseur ν

Pour extraire des informations des zones autour d'un patch $\Gamma_\nu(p)$, dans le but de décider ensuite de la conformation de la surface autour du point p , nous avons choisi de nous focaliser sur les patches d'épaisseur ν centrés en des points du contour de $\Gamma_\nu(p)$.

Soit \mathcal{O}_b un objet discret et \mathcal{S}_b l'ensemble des surfels appartenant à la surface de \mathcal{O}_b . Notons $\mathcal{S}_{\Gamma_\nu(p)}$ le sous-ensemble de surfels de \mathcal{S}_b appartenant au patch $\Gamma_\nu(p)$. Un point q appartient au *contour* de $\Gamma_\nu(p)$ si la représentation voxel de q possède un surfel $s \in \mathcal{S}_{\Gamma_\nu(p)}$ et s'il existe un surfel $s' \in \mathcal{S}_b \setminus \mathcal{S}_{\Gamma_\nu(p)}$ tel que s et s' soient adjacents par arête.

Le fait que les patches soient développés de manière à rester homéomorphes à un disque topologique nous assure que nous pouvons bien construire le contour d'un patch.

Un exemple de contour d'un patch discret d'épaisseur 2 est proposé à la figure 4.15.

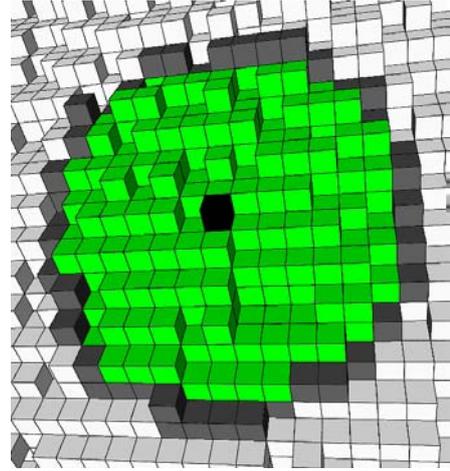


Figure 4.15 – En gris, les points du contour d'un patch d'épaisseur 2, centré au point noir.

4.3.4.4 Estimateur de forme à l'épaisseur ν

À l'aide des différentes notions présentées ci-dessous, nous pouvons finalement présenter notre estimateur de forme permettant de faire la distinction entre les zones plates, les creux et les bosses sur le bord d'un objet discret possiblement bruité.

Pour estimer la conformation de la surface autour d'un point p du bord d'un objet discret, la méthode est la suivante : étant donnée une épaisseur ν , nous développons un patch discret $\Gamma_\nu(p)$ d'épaisseur ν à partir de p et nous calculons son contour C . Ensuite, pour chaque point $q_i, 1 \leq i \leq |C|$ nous développons un patch voisin $\Gamma_\nu(q_i)$. À partir de ces derniers, nous obtenons un champ de vecteurs normaux $\{\vec{n}_\nu(q_i)\}$ sur le contour C . Un exemple d'un tel champ de vecteurs normaux est illustré à la figure 4.16.

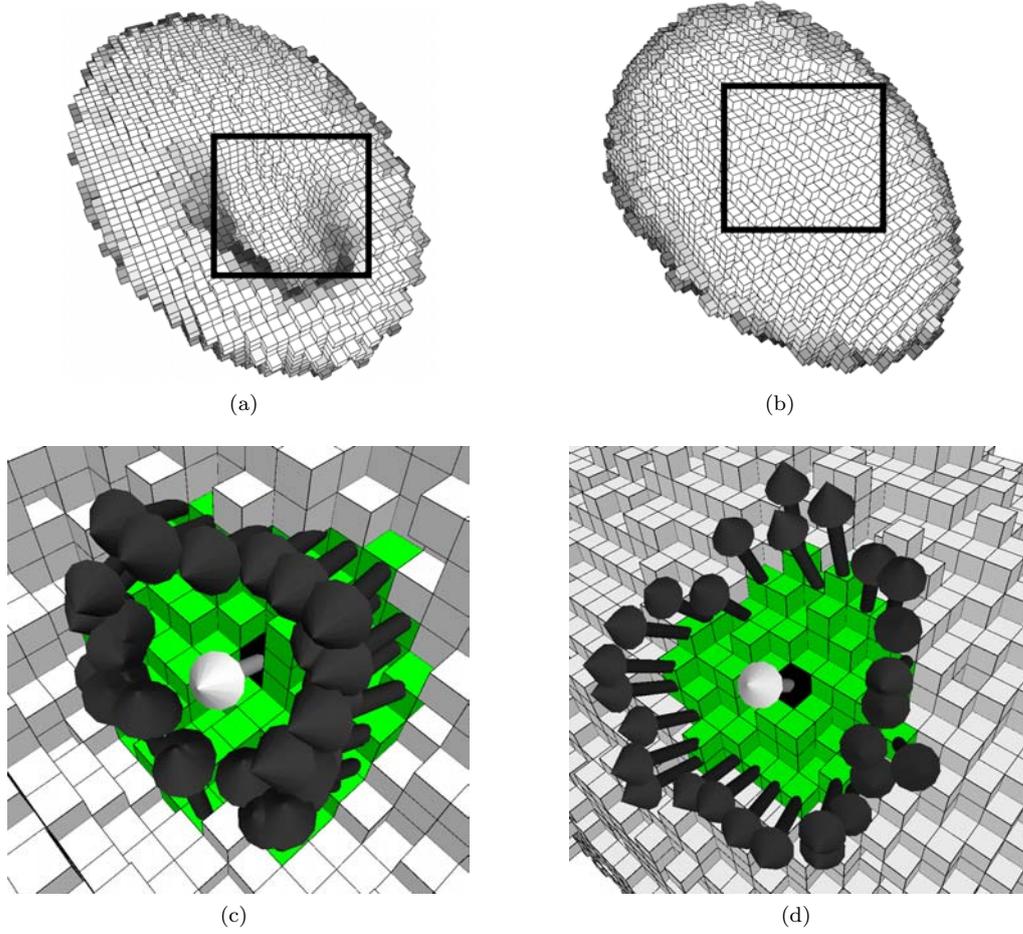


Figure 4.16 – Les champs de vecteurs normaux sur le contour d’un patch discret (a), (c) dans une partie creuse de l’objet et (b), (d) dans une partie bombée de l’objet.

Observations

Dans les zones creuses de l’objet 4.16(c), le champ de vecteurs $\{\vec{n}_\nu(q_i)\}$ pointe vers le vecteur normal $\vec{n}_\nu(p)$ du patch central, alors que dans les zones bombées 4.16(d) ce même champ de vecteurs a la tendance inverse. La valeur de l’angle orienté entre le vecteur $\vec{n}_\nu(p)$ et les vecteurs $\vec{n}_\nu(q_i)$ peut donc apporter des informations permettant de faire la distinction entre ces zones. Notre estimateur de forme exploite cette information :

Définition 4.11 Soit $\Gamma_\nu(p)$ un patch discret d’épaisseur ν développé en un point p du bord d’un objet discret et C l’ensemble des points du contour de $\Gamma_\nu(p)$. On appelle *estimateur de forme à l’épaisseur ν au point p* la valeur

$$\mathcal{F}_\nu(p) = \frac{1}{|C|} \sum_{\forall q \in C} (\widehat{\vec{n}_\nu(p), \vec{n}_\nu(q)}) \cdot \frac{E_A(\Gamma_\nu(q))}{E_A(\Gamma_\nu(p))}$$

où $(\widehat{\vec{n}_\nu(p), \vec{n}_\nu(q)})$ représente la valeur de l’angle orienté entre les deux vecteurs.

L’estimateur $\mathcal{F}_\nu(p)$ est donc une moyenne pondérée des valeurs des angles entre la normal au patch et les vecteurs normaux des patches avoisinants.

Orientation des angles : Le signe de la valeur de l'angle $(\widehat{\vec{n}_\nu(p), \vec{n}_\nu(q)})$ est déterminé comme indiqué sur la figure 4.17. Dans un premier temps les deux produits vectoriels $\vec{n}_\nu(p) \wedge \vec{p}\vec{q}$ et $\vec{n}_\nu(p) \wedge \vec{n}_\nu(q)$ sont calculés, puis le signe est déterminé en fonction du produit scalaire des deux vecteurs résultants. Ainsi, lorsqu'on considère que les vecteurs résultants pointent dans la même direction, c'est-à-dire lorsque le produit scalaire $(\vec{n}_\nu(p) \wedge \vec{p}\vec{q}) \cdot (\vec{n}_\nu(p) \wedge \vec{n}_\nu(q))$ est positif, l'angle entre les vecteurs est compté positivement, sinon, l'angle est compté négativement. Les produits vectoriels ne sont faisables que si les vecteurs ne sont pas colinéaires et le produit scalaire ne doit pas être nul. Bien que dans la plupart des cas ces conditions soient satisfaites, il arrive en pratique, tout particulièrement lorsque les patchs sont très petits, que de telles dégénérescences apparaissent. Comme elles n'apportent pas d'information pertinente sur la conformation de la surface en ces zones, nous pouvons tout bonnement les ignorer et continuer avec le prochain point du contour C .

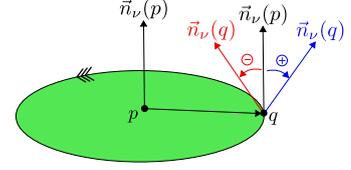


Figure 4.17 – Signe de la valeur angulaire entre les vecteurs.

Coefficients de pondération : Conformément aux interprétations précédentes, un patch $\Gamma_\nu(p)$ exprime le caractère planaire de la surface autour de p , relativement à l'épaisseur ν considérée. Ainsi, les informations apportées par les patchs développés en des points du contour de $\Gamma_\nu(p)$ sont à prendre en considération de façon inversement proportionnelle à la taille de $\Gamma_\nu(p)$, puisque plus ce dernier est grand, moins les patchs avoisinants apportent d'information sur la conformation locale de la surface autour de p . D'où l'utilisation du facteur $\frac{1}{E_A(\Gamma_\nu(p))}$ dans la formule.

D'autre part, lorsque pour des points q_i et q_j , $j \neq i$, du contour de $\Gamma_\nu(p)$, le patch $\Gamma_\nu(q_i)$ est plus étendu que le patch $\Gamma_\nu(q_j)$, cela traduit un caractère plus plat de la surface autour de q_i qu' autour de q_j . Si nous interprétons cette planarité comme le fait que la surface autour du point est moins perturbée, nous pouvons considérer l'information apportée par $\Gamma_\nu(q_i)$ comme plus fiable que celle apportée par $\Gamma_\nu(q_j)$. C'est pourquoi nous avons choisi de considérer l'information apportée par ces patchs proportionnellement à leurs tailles, d'où l'association du facteur $E_A(\Gamma_\nu(q))$ à la valeur angulaire $(\widehat{\vec{n}_\nu(p), \vec{n}_\nu(q)})$ dans la formule.

Interprétation de l'estimateur de forme : Comme résultat, il ressort que le bord d'un objet discret est caractérisé *bombé* autour de p lorsque $\mathcal{F}_\nu(p)$ est positif, et *creux* autour de p lorsque $\mathcal{F}_\nu(p)$ est négatif ; les valeurs croissantes de $|\mathcal{F}_\nu(p)|$ signifiant un caractère plus prononcé de la conformation dans chacun des cas. Lorsque $\mathcal{F}_\nu(p)$ est proche de 0, deux interprétations sont possibles : soit $\Gamma_\nu(p)$ est étendu, et dans ce cas la surface autour de p est considérée comme plate, soit $\Gamma_\nu(p)$ est petit, et dans ce cas on ne peut pas vraiment qualifier la conformation de la surface autour de p . Néanmoins, nous pouvons différencier ces deux cas suivant la taille de $\Gamma_\nu(p)$, c'est-à-dire en utilisant l'estimateur d'aire $E_A(\Gamma_\nu(p))$.

Nous proposons à la figure 4.18 quelques résultats obtenus avec cet estimateur de forme sur différents objets discrets, bruités et non bruités. Le rendu de chaque objet est fait en associant à chaque voxel v de l'objet son vecteur normal $\vec{n}_nu(v)$. De plus ces objets sont colorés suivant l'échelle de couleur indiqué à gauche. Celle-ci est constituée d'un dégradé linéaire de la couleur verte à la couleur rouge (resp. bleue) représentant les valeurs positives (resp. négatives) de $\mathcal{F}_\nu(p)$ entre 0 et le 3^e quartile des valeurs positives (resp. négatives) de $\mathcal{F}_\nu(p)$, les valeurs supérieures (resp. inférieures) étant représentées en rouge (resp. bleu).

Comme nous pouvons le constater, l'estimateur nous permet de distinguer les zones creuses et bombées au bord des objets discrets, sous réserve, ici aussi, de choisir une épaisseur ν adéquate. En fait, tout comme l'estimateur proposé dans [BGCF95], une épaisseur trop petite rend l'estimation très sensible au bruit. Une épaisseur plus importante tend à lisser l'estimation, mais le caractère adaptatif des patchs au regard du bord de l'objet permet de limiter les cas problématiques tels que ceux illustrés à la figure 4.13 (il va de soi qu'avec une épaisseur trop grande nous pouvons difficilement éviter ces problèmes sans la prise en compte d'informations supplémentaires).

Malgré tout, même avec une épaisseur convenable comme à la figure 4.18(c) où $\nu = 3$, nous pouvons noter quelques artefacts au niveau de l'estimation dans les zones où le bruit est « plus présent ». Cela

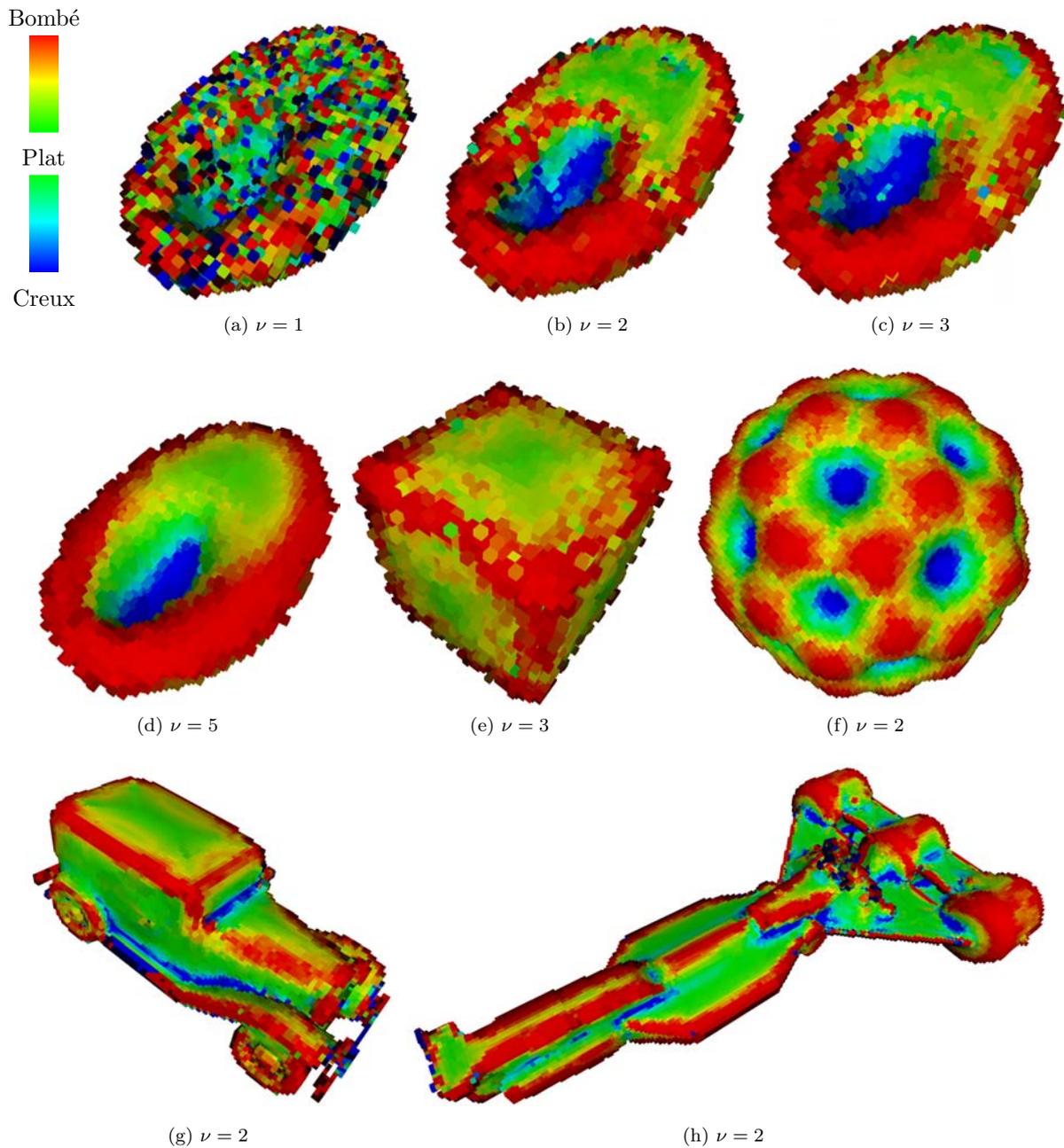


Figure 4.18 – Résultats de l'estimateur de forme sur différents objets discrets. (a) - (d) Un demi-ellipsoïde creusé bruité traité aux épaisseurs 1, 2, 3 et 5. (e) Un cube bruité traité à l'épaisseur 3. (f) Une molécule de fullerène discrétisée, (g) une voiture[†] et (h) un vaisseau spatial[†], tous trois traités à l'épaisseur 2.

[†]. disponibles sur le site du TC18 : http://tc18.org/code_data_set/3D_images.html

provient en fait d'une mauvaise estimation de la normale en ces points. En effet, comme l'estimateur de forme exploite, entre autres, les variations angulaires entre certaines normales, une mauvaise estimation de ces dernières peut perturber les calculs. Comme l'estimateur décrit dans [TW97] a tendance à fournir un champ de vecteurs normaux plus régulier que les autres méthodes (malgré un effet lissant plus prononcé), nous avons calculé $\mathcal{F}_\nu(\cdot)$ en remplaçant dans la formule les normales à l'épaisseur ν , $\vec{n}_\nu(\cdot)$, par des normales estimées à l'aide de méthode [TW97]; le voisinage de points considéré pour l'estimation de l'aire et du contour restant, quant à lui, un patch d'épaisseur ν . Les résultats de ce changement sont illustrés à la figure 4.19. Comme nous pouvons le remarquer (en comparant aux figures 4.18(a) - (c)), la fiabilité de l'estimation des normales a une influence considérable sur l'estimation de la forme, comme cela était prévisible, surtout au niveau des très petits patches ($\nu = 1$) fournissant une mauvaise estimation des normales pour des objets bruités.

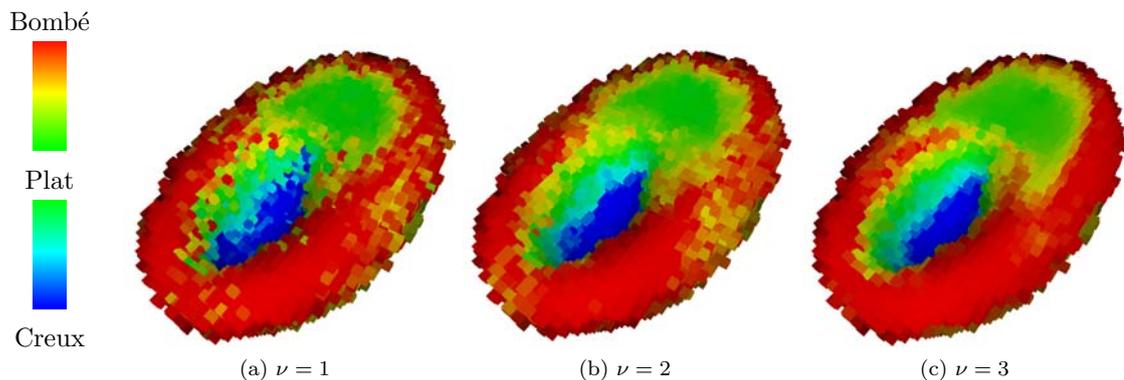


Figure 4.19 – Estimateur de forme en remplaçant l'estimation de la normale d'épaisseur ν par une estimation de la normale avec l'approche [TW97], pour une taille de voisinage $ns = 6$. Les patches utilisés dans l'estimateur de forme sont développés à différentes épaisseurs 1, 2 et 3.

4.4 Conclusion

À travers ce chapitre nous avons proposé différents estimateurs de caractéristiques géométriques pour des objets discrets possiblement bruités. Ces derniers peuvent être classés selon deux catégories : l'une *directionnelle*, exploitant les contours des tranches de l'objet et l'autre *multi-directionnelle*, exploitant un voisinage surfacique autour des points, à travers la notion de *patch discret d'épaisseur ν* .

Pour chaque catégorie, nous avons proposé un estimateur de vecteur normal ainsi qu'un estimateur de forme permettant de caractériser la conformation du bord de l'objet selon les trois cas *creux*, *bombé* et *plat*.

Au vu des résultats obtenus, il semble clair que les approches par tranches ne sont pas convaincantes au regard du type de bruit considéré. En effet, les perturbations synthétisées entraînent la considération de tranches parasites, apparaissant uniquement à cause du bruit. Même si les estimateurs directionnels sont paramétrés par une valeur ν permettant de contrôler les déformations autorisées au sein des primitives floues, ces tranches non fiables faussent complètement les résultats. De plus, comme ces estimateurs reposent sur les tranches de l'objet, les résultats sont très dépendants de l'orientation de ce dernier dans l'espace.

Au contraire, les estimateurs reposant sur les patches discrets apportent plus de stabilité dans les estimations. Comme le patch s'étend dans toutes les directions, ce dernier représente une approximation du plan tangent moins sensible aux variations de l'orientation de l'objet dans l'espace que l'estimation du plan tangent grâce à une base issue de l'estimation des tangentes sur les contours des tranches de l'objet, fiabilisant ainsi l'estimation des normales.

Au niveau de l'estimateur de forme, les résultats obtenus sont satisfaisants et permettent de différencier les zones plates, creuses et bombées de l'objet. Il serait malgré tout intéressant d'étudier plus en détail

les relations qui pourraient exister entre notre estimateur et la notion de courbure généralement utilisée pour caractériser la conformation de la surface d'un objet.

Néanmoins cet estimateur est suffisant pour guider le processus de segmentation du bord d'un objet discret comme nous allons le voir dans le chapitre suivant.

Il est à noter que le contenu de ce chapitre a fait l'objet d'une publication avec présentation orale dans la conférence *Discrete Geometry for Computer Imagery* en 2008 [PDR08a].

Chapitre 5

Segmentation d'objets discrets bruités et applications

Sommaire

5.1	Introduction	109
5.2	Segmentation du bord d'un objet discret bruité	110
5.2.1	Choix de la graine	110
5.2.2	Algorithme de reconnaissance	111
5.2.3	Critères de bonne formation	111
5.2.4	Développement des MFPD $_{\nu}$	111
5.3	Lissage	117
5.4	Vers la polyédrisation d'objets discrets bruités 3D	121
5.4.1	Lissage d'objets discrets bruités non convexes	121
5.4.2	Idées pour des perspectives de polyédrisation	123
5.5	Conclusion	124

5.1 Introduction

Comme nous l'avons déjà annoncé dans l'introduction de ce manuscrit, nous avons choisi d'attaquer le problème de la polyédrisation d'objets discrets en nous reposant sur une segmentation préalable du bord de l'objet en morceaux de plans discrets épais. Plus précisément, il s'agit de *partitionner* le bord de l'objet en *morceaux flous de plans discrets d'épaisseur ν* . Nous avons mis en avant certains problèmes de sur-segmentation apparaissant lorsque nous cherchons à traiter des objets discrets bruités avec des techniques basées sur des primitives discrètes arithmétiques régulières (cf. figure 2.27) et nous allons voir dans la suite que la flexibilité apportée par l'utilisation des MFPD permet d'obtenir des résultats plus satisfaisants sur ce type d'objet.

L'objectif est ici d'essayer de partitionner les objets discrets de façon à obtenir une segmentation cohérente avec les arêtes naturelles des objets. Nous voudrions ainsi avoir autant de segments (c'est-à-dire de MFPD) que de faces. Cela a du sens pour des objets polyédriques comme par exemple un cube pour lequel nous aimerions retrouver six segments correspondant aux six faces naturelles du cube. Pour des objets non linéaires par morceaux, comme les sphères, il est moins évident de définir un but aussi intuitif. Nous essaierons simplement, dans ce cas, d'éviter que les segments ne s'entrelacent.

Dans la première partie de ce chapitre nous allons donc présenter un algorithme de segmentation en morceaux flous de plans discrets adapté aux objets bruités. Puis, nous verrons comment extraire une version lissée (ou débruitée) d'un objet convexe, en nous basant sur cette décomposition en MFPD. Enfin,

dans la dernière partie du chapitre, nous montrerons quelques résultats d'expériences effectuées dans le cadre de la polyédrisation d'objets discrets bruités. Nous ne proposons malheureusement pas de solution concrète à ce problème, mais nous mettrons en avant les difficultés rencontrées et discuterons certaines idées envisagées.

5.2 Segmentation du bord d'un objet discret bruité

Le processus de segmentation que nous présentons ici se place dans un cadre généralement adopté pour les décompositions à bases de primitives discrètes linéaires, à savoir :

1. le *choix d'une graine* pour démarrer un nouveau morceau de plan ;
2. l'extension du morceau de plan grâce à un *algorithme de reconnaissance* d'une telle primitive en respectant certains *critères de bonne formation*.

Nous désirons aussi guider cette segmentation à l'aide de caractéristiques géométriques du bord de l'objet discret. Puisque nous travaillons sur des objets possiblement bruités, nous allons considérer les caractéristiques présentées au chapitre précédent à l'épaisseur ν . Nous appliquons donc, sur chaque objet discret que nous désirons segmenter en MFPD, un pré-calcul visant à fournir ces caractéristiques. Ainsi, pour un réel ν donné, nous développons en chaque point p du bord \mathcal{B}_b de l'objet un patch discret d'épaisseur ν centré en p comme décrit à la section 4.2.2.2. Nous obtenons alors, pour chaque point $p \in \mathcal{B}_b$:

- une estimation du vecteur normal à \mathcal{B}_b en p : $\vec{n}_\nu(p)$,
- un coefficient d'aire : $E_{\mathcal{A}}(\Gamma_\nu(p))$;
- un facteur de forme : $\mathcal{F}_\nu(p)$.

Nous détaillons à présent les différentes étapes du processus de segmentation.

5.2.1 Choix de la graine

Le choix d'un élément de départ, *la graine*, pour débiter un nouveau morceau de plan discret n'est pas anodin et affecte substantiellement le résultat du processus de segmentation. Une solution interactive serait de laisser l'utilisateur choisir l'ensemble des graines et leur ordre de traitement. Néanmoins, il est toujours intéressant de disposer d'une méthode automatique, offrant la possibilité de traiter les images par lots. Le choix le plus simple est alors probablement de maintenir à jour l'ensemble des points du bord de l'objet restant à visiter et d'en choisir un, aléatoirement, lorsqu'il est nécessaire de commencer la reconnaissance d'un nouveau segment. Malheureusement cette approche ne donne aucun contrôle sur la décomposition résultante. En effet, si nous prenons en exemple le cas d'un cube, un mauvais choix serait de prendre pour graines des points du bord se situant près des arêtes du cube. Ce cas pourrait potentiellement conduire à une sur-segmentation telle que celle illustrée à la figure 5.1(a). Cette solution a, de plus, le désavantage de ne pas fournir la même décomposition d'un objet à chaque exécution de l'algorithme (sauf si nous utilisons un générateur pseudo-aléatoire avec une graine fixée).

Une autre possibilité, qui permettrait d'éviter ce côté non reproductible de l'expérience lorsque nous utilisons de l'aléatoire, est de choisir un critère arbitraire comme, par exemple, choisir comme nouvelle graine le dernier point visité qui n'a pas pu être ajouté au dernier morceau de plan en cours de reconnaissance (technique utilisée dans [Siv04]). Pour peu que l'algorithme de reconnaissance soit bien déterministe et que nous démarrions du même point de départ, la décomposition est dans ce cas, pour un objet donné, toujours la même. D'autres critères, dépendant généralement de la technique de décomposition utilisée peuvent être considérés, comme par exemple ceux proposés au chapitre 2, à la page 43, pour la reconstruction analytique inversible. Néanmoins ces critères ne dépendent pas de la géométrie de l'objet testé.

Dans le cas qui nous intéresse, nous avons choisi de guider la segmentation grâce aux caractéristiques géométriques du bord de l'objet discret. Le choix des graines va donc naturellement dépendre de ces dernières. Et puisque nous désirons décomposer les objets de façon à retrouver les faces naturelles de ces

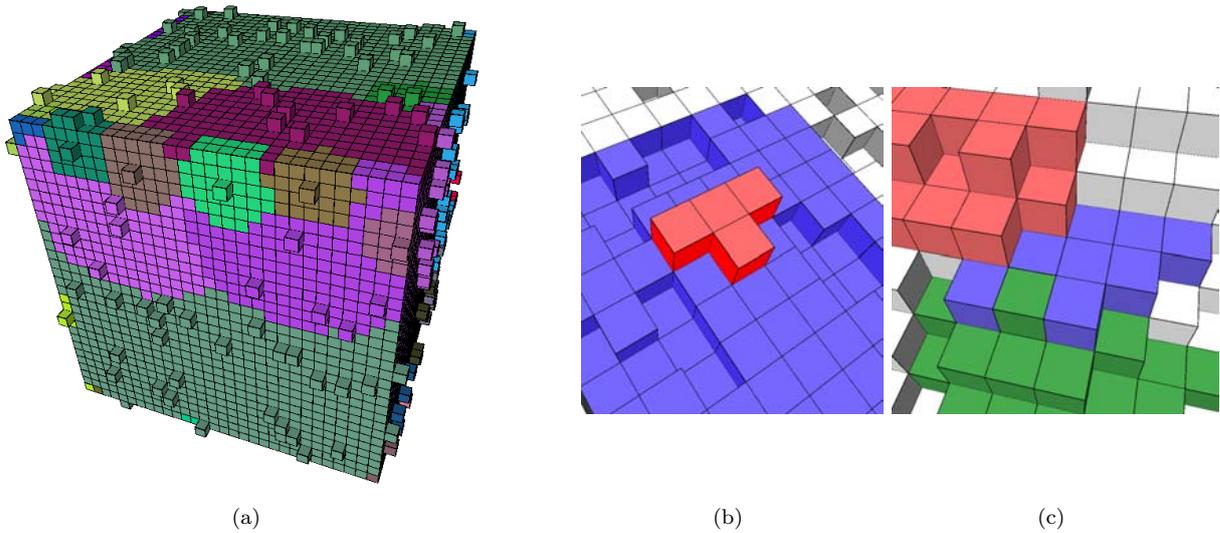


Figure 5.1 – Situations à éviter pour une segmentation en MFPD. (a) Une sur-segmentation due au choix aléatoire des graines.(b) Le MFPD rouge est situé à l'intérieur du MFPD bleu, et (c) le bord du MFPD bleu semble difficilement exploitable pour créer une facette de polyèdre.

derniers (lorsque cela a un sens), il semble logique de choisir les graines en priorité dans les endroits les plus « plats » du bord. Nous nous reposons donc ici sur le coefficient $E_{\mathcal{A}}(\Gamma_{\nu}(p))$ pour effectuer notre choix. L'idée est de prendre comme nouvelle graine le point avec la plus grande valeur de $E_{\mathcal{A}}(\Gamma_{\nu}(p))$, parmi les points encore non traités. Comme ce coefficient reflète le caractère plat du bord autour de p , il semble naturel de l'utiliser.

5.2.2 Algorithme de reconnaissance

Puisque nous désirons apporter une certaine tolérance aux perturbations du bord de l'objet lors de sa segmentation, nous allons tout logiquement utiliser l'algorithme incrémental de reconnaissance de morceaux flous de plans discrets présenté à la page 62.

5.2.3 Critères de bonne formation

Le premier critère à vérifier est que l'épaisseur du morceau flou de plan discret en cours de reconnaissance ne dépasse pas une valeur limite ν , fixée par l'utilisateur. Mais ceci est fait implicitement par l'algorithme de reconnaissance.

L'autre critère que nous avons fixé provient du fait que nous désirons utiliser la segmentation en MFPD comme base pour une méthode de polyédrisation. Comme certains auteurs l'ont fait remarquer dans leurs travaux [Pap99, Siv04], il semble raisonnable de se limiter aux segments homéomorphes à un disque topologique car ces derniers permettent de donner une notion de bord correcte aux faces discrètes reconnues. Nous évitons ainsi les situations illustrées aux figures 5.1(b) et (c) qui peuvent poser problème pour la phase de polyédrisation. Nous imposons ainsi que les projections dans les plans principaux du 26-voisinage du point à ajouter dans le MFPD en cours de reconnaissance respectent les contraintes présentées à la figure 4.6. Nous utilisons pour cela la même technique que celle décrite au chapitre précédent à la page 87.

5.2.4 Développement des MFPD_{ν}

Le développement des MFPD_{ν} dépend très fortement de la manière dont sont visités les points du bord de l'objet à décomposer. Dans [Siv04], I. Sivignon a proposé une étude intéressante montrant comment

les morceaux de plans se développent en fonction de différents parcours (la figure 2.14 reprend quelques exemples). Nous pouvons néanmoins noter que le résultat est très dépendant de l'ordre de parcours du 26-voisinage d'un voxel et ne prend pas en compte les caractéristiques géométriques au bord de l'objet.

5.2.4.1 Stratégie par extension maximale

Nous proposons donc ici une extension permettant d'avoir une décomposition plus intuitive de l'objet en utilisant une stratégie d'extension maximale tenant compte des discontinuités naturelles de l'objet. L'idée est d'étendre le $MFPD_\nu$ en cours de reconnaissance en restant dans les zones les plus plates possibles. Pour cela nous considérons les points q adjacents à son bord qui n'ont pas encore été visités, et nous choisissons celui avec la plus grande valeur de $E_{\mathcal{A}}(\Gamma_\nu(q))$. Encore une fois nous exploitons le fait que les grandes valeurs de $E_{\mathcal{A}}(\Gamma_\nu(q))$ représentent un caractère plat plus prononcé du bord de l'objet.

Fonction $MFPD_ExtensionMaximale(\mathcal{O}_b, g, \nu)$

Entrées : Un objet discret tri-dimensionnel 6-connexe \mathcal{O}_b , un point g – la graine – appartenant au bord \mathcal{B}_b de \mathcal{O}_b et un réel ν

Sortie : Le $MFPD_\nu$ étendu au maximum relativement aux critères de bonne formation

Variables : Q , une file de priorité initialement vide

P , le $MFPD_\nu$ en cours de reconnaissance

Visité, un tableau indicé par les points, contenant les marques (initialement à faux)

```

1 début
  // Initialisation
2  Q.insérer( $g, E_{\mathcal{A}}(\Gamma_\nu(g))$ )
3  Visité[ $g$ ] ← vrai
4  P ← initier_nouveau_MFPD( $\nu$ )
  // Phase d'extension
5  tant que  $Q \neq \emptyset$  faire
6     $v \leftarrow Q.extraire\_max()$ 
7    si  $v$  vérifie les critères de bonne formation alors
8      P.ajouter( $v$ )
9      pour chaque voisin  $w$  du 26-voisinage  $v$  faire
10       si non Visité[ $w$ ] alors
11         Q.insérer( $g, E_{\mathcal{A}}(\Gamma_\nu(g))$ )
12     sinon
13       Visité[ $v$ ] ← faux
14  retourner P
15 fin

```

Pour implémenter cette stratégie nous utilisons une file de priorité Q . Nous commençons par ajouter dans Q le voxel g issu du procédé de choix de la graine, associé à son coefficient $E_{\mathcal{A}}(\Gamma_\nu(g))$ et nous marquons ce voxel comme **visité**. Puis, nous initialisons P , un nouveau procédé de reconnaissance de $MFPD_\nu$. Ensuite, tant que la file Q n'est pas vide, nous y extrayons le voxel v de plus grande priorité, c'est-à-dire celui associé à la plus grande valeur de $E_{\mathcal{A}}(\Gamma_\nu(p))$ et nous l'ajoutons à P s'il satisfait les critères de bonne formation présentés précédemment. Nous mettons à jour Q , qui représente la liste des points possibles à ajouter à P , en y insérant les points q du 26-voisinage de v non encore visités, associés à leur coefficient $E_{\mathcal{A}}(\Gamma_\nu(q))$, et nous les marquons comme **visité**. Si, par contre, un des critères n'est pas satisfait, le point v est marqué comme **non visité** afin de donner l'opportunité à l'algorithme de le revisiter plus tard, lorsque ces critères seront peut-être vérifiés¹¹. De cette manière le développement de

11. Nous pouvons tout de même laisser marqués comme **visité**, les points ne satisfaisant pas la contrainte d'épaisseur des

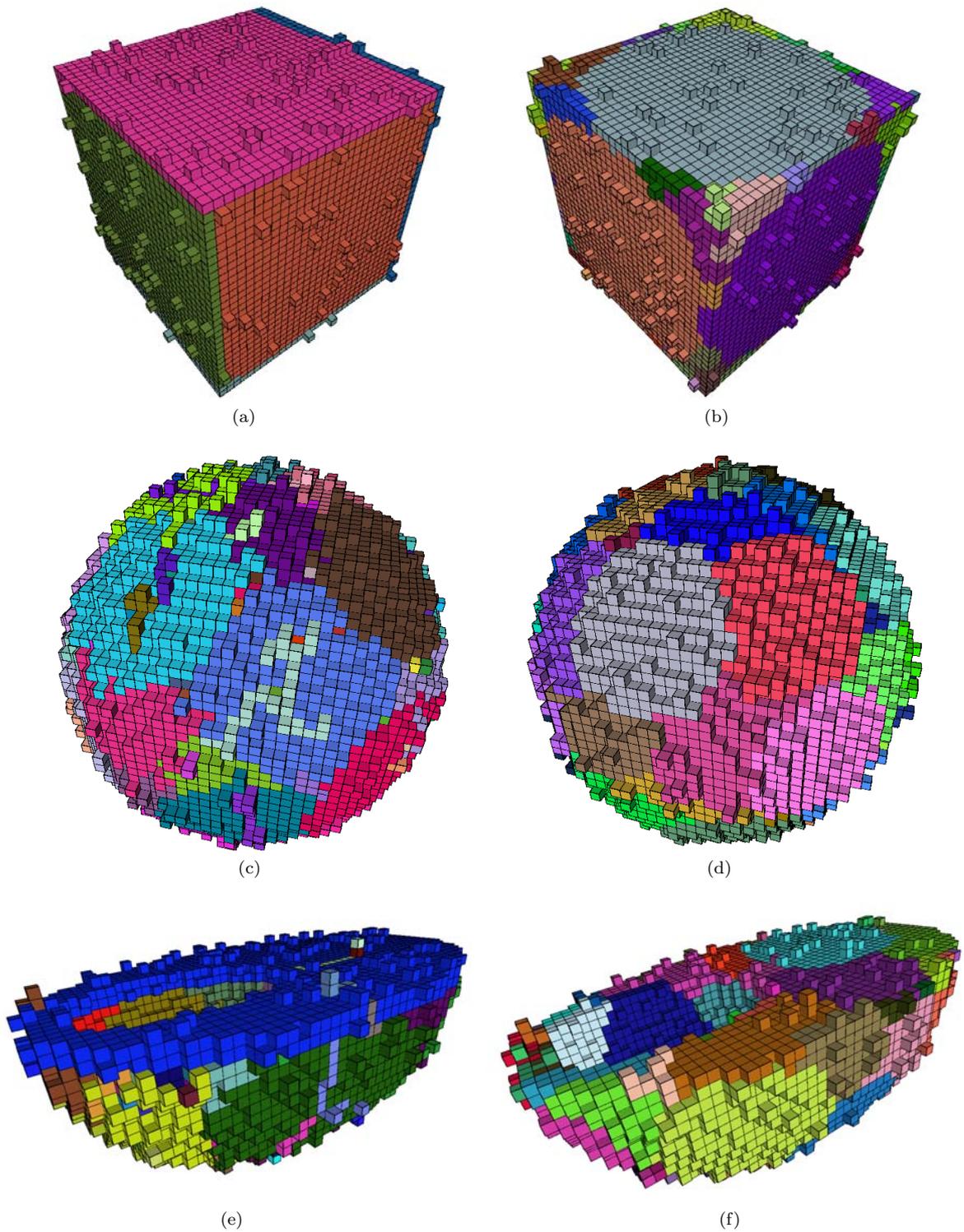


Figure 5.2 – Segmentation à l'épaisseur 2 (a) d'un cube d'arête 30, (c) d'une sphère de rayon 20 et (e) d'un demi-ellipsoïde creusé, en utilisant dans les trois cas la stratégie d'extension maximum pour les MFPD. Les mêmes objets, (b), (d) et (f), mais segmentés en utilisant des patches discrets d'épaisseur 2 uniquement.

P ne s'arrête pas si un point ne peut pas être ajouté, obtenant ainsi une extension maximale, relativement aux critères utilisés. La fonction `MFPD_ExtensionMaximale` résume les différentes étapes de cette stratégie.

Le résultat de cette stratégie sur différents objets discrets peu bruités est illustré sur la partie gauche de la figure 5.2. Nous proposons sur la partie droite de cette même figure, comme base de comparaison, le résultat obtenu sur ces mêmes objets, en utilisant non pas la stratégie de développement maximale mais en développant des patchs discrets d'épaisseur ν .

Une première constatation, positive, est que, dans le cas de l'extension maximale, nous retrouvons au niveau du cube six segments correspondant à ses six faces naturelles. C'est une nette amélioration par rapport à la sur-segmentation de la figure 2.27(b) obtenue en utilisant des primitives arithmétiques régulières, mais aussi par rapport au résultat obtenu avec les patchs discrets épais qui produisaient de nombreux petits segments dans les coins du cube. D'un autre côté, la décomposition obtenue au niveau de la sphère dans le cas de l'extension maximale est perfectible, notamment si nous la comparons au résultat obtenu en développant des patchs discrets épais. Le problème vient ici du caractère courbé du bord des sphères. Les points ne pouvant être ajoutés sans induire un dépassement de l'épaisseur ν autorisée vont créer des obstacles que la stratégie d'extension maximale va, par nature, avoir tendance à « contourner », créant ainsi des segments en forme de Pac-Man 🍷. Il semblerait donc que dans les zones non plates de l'objet, il est plus raisonnable de développer des patchs que d'étendre le plus possible les MFPD. Le cas mixte du demi-ellipsoïde creusé confirme ces tendances, puisque la zone plate formant un grand segment dans le cas d'une extension maximale se trouve sur-segmentée lorsque nous utilisons des patchs discrets, alors que les zones courbées bien segmentées avec les patchs discrets, sont entrecoupées de petits morceaux de plans formant des chemins fins en utilisant l'autre stratégie.

5.2.4.2 Stratégie hybride

Algorithme 5.2 : Segmentation hybride

Données : Un objet discret tri-dimensionnel 6-connexe \mathcal{O}_b , une valeur seuil σ et un réel ν

Résultat : Un partitionnement du bord \mathcal{B}_b de \mathcal{O}_b en MFPD $_\nu$, à travers la liste `MorceauxPlans`

```

1 début
2   ÀVisiter ←  $\mathcal{B}_b$ , le bord de  $\mathcal{O}_b$ 
3   MorceauxPlans ←  $\emptyset$ 
4   tant que ÀVisiter  $\neq \emptyset$  faire
5      $g \leftarrow$  le point  $p \in$  ÀVisiter avec la plus grande valeur  $E_{\mathcal{A}}(\Gamma_\nu(g))$ 
6     si  $|\mathcal{F}_\nu(g)| < \sigma$  alors
7       MP ← MFPD_ExtensionMaximale( $\mathcal{O}_b, g, \nu$ )
8     sinon
9       MP ←  $\Gamma_\nu(g)$ 
10    pour chaque point  $p \in$  MP faire
11      ÀVisiter ← ÀVisiter  $\setminus \{p\}$ 
12    MorceauxPlans ← MorceauxPlans  $\cup$  MP
13  retourner MorceauxPlans
14 fin
```

Ces observations nous ont alors amené à considérer une approche hybride, exploitant les caractéristiques géométriques du bord de l'objet, en développant des MFPD étendus au niveau des zones plates de ce dernier, et des patchs discrets épais dans les parties courbées. Pour différencier ces zones nous utilisons l'estimateur de forme $\mathcal{F}_\nu(p)$. Comme nous l'avons expliqué à la page 105, les valeurs croissantes de $\mathcal{F}_\nu(p)$

MFPD $_\nu$. En effet comme nous ne retirons jamais de points à P , il n'est pas possible qu'un ajout entraînant un dépassement d'épaisseur à une étape i n'en entraîne pas un à une étape $j > i$.

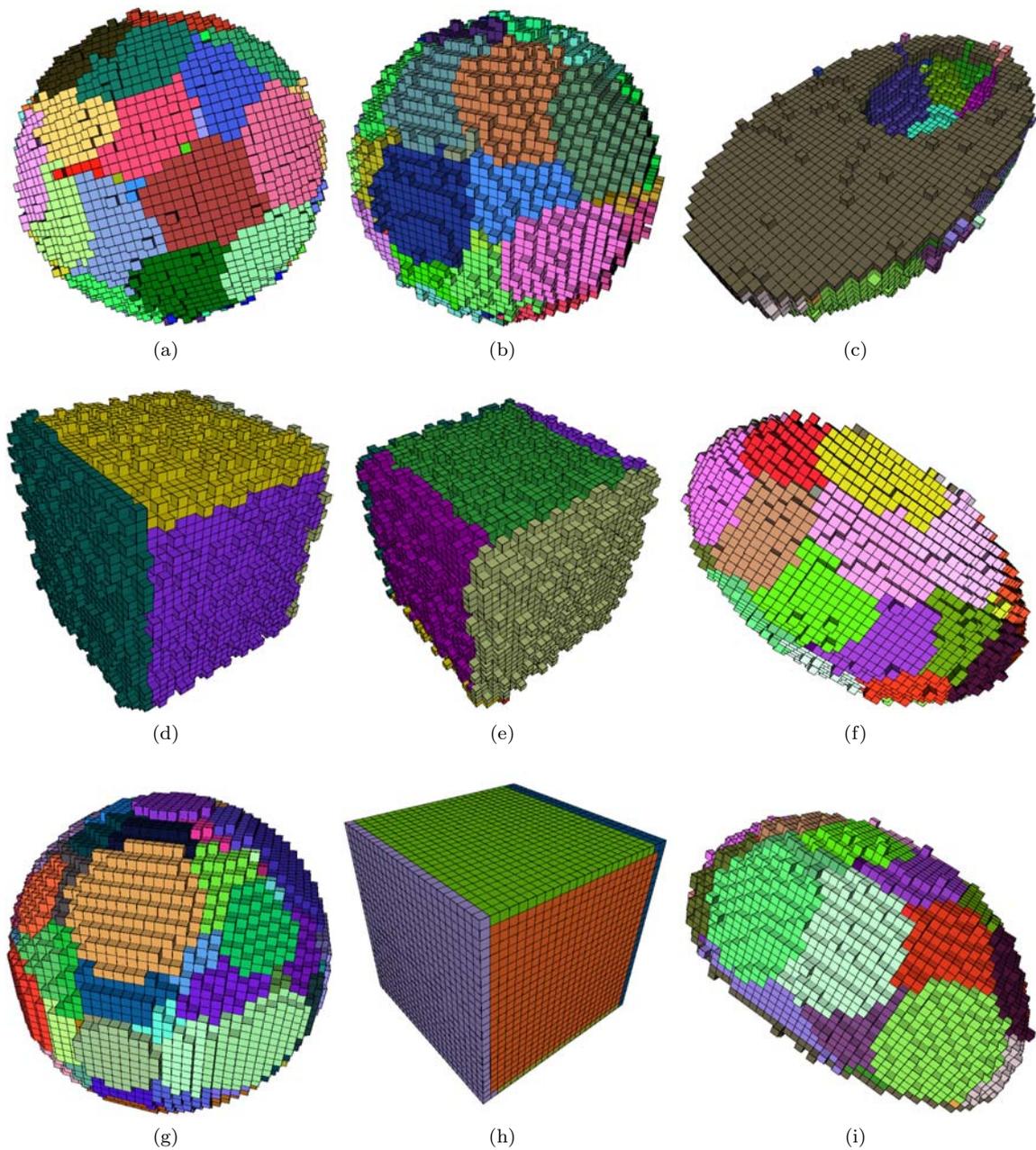


Figure 5.3 – Résultats de la méthode hybride de segmentation sur différents objets discrets synthétisés : (a), (b) une segmentation à l'épaisseur 2 d'une sphère faiblement bruitée de rayon 20 ; (d), (e) une segmentation à l'épaisseur 3 d'un cube fortement bruité d'arête 25 ; (g), (h) une segmentation à l'épaisseur 1 d'objets non bruités ; et (c), (f), (i) une segmentation à l'épaisseur 2 d'un demi-ellipsoïde creusé peu bruité.

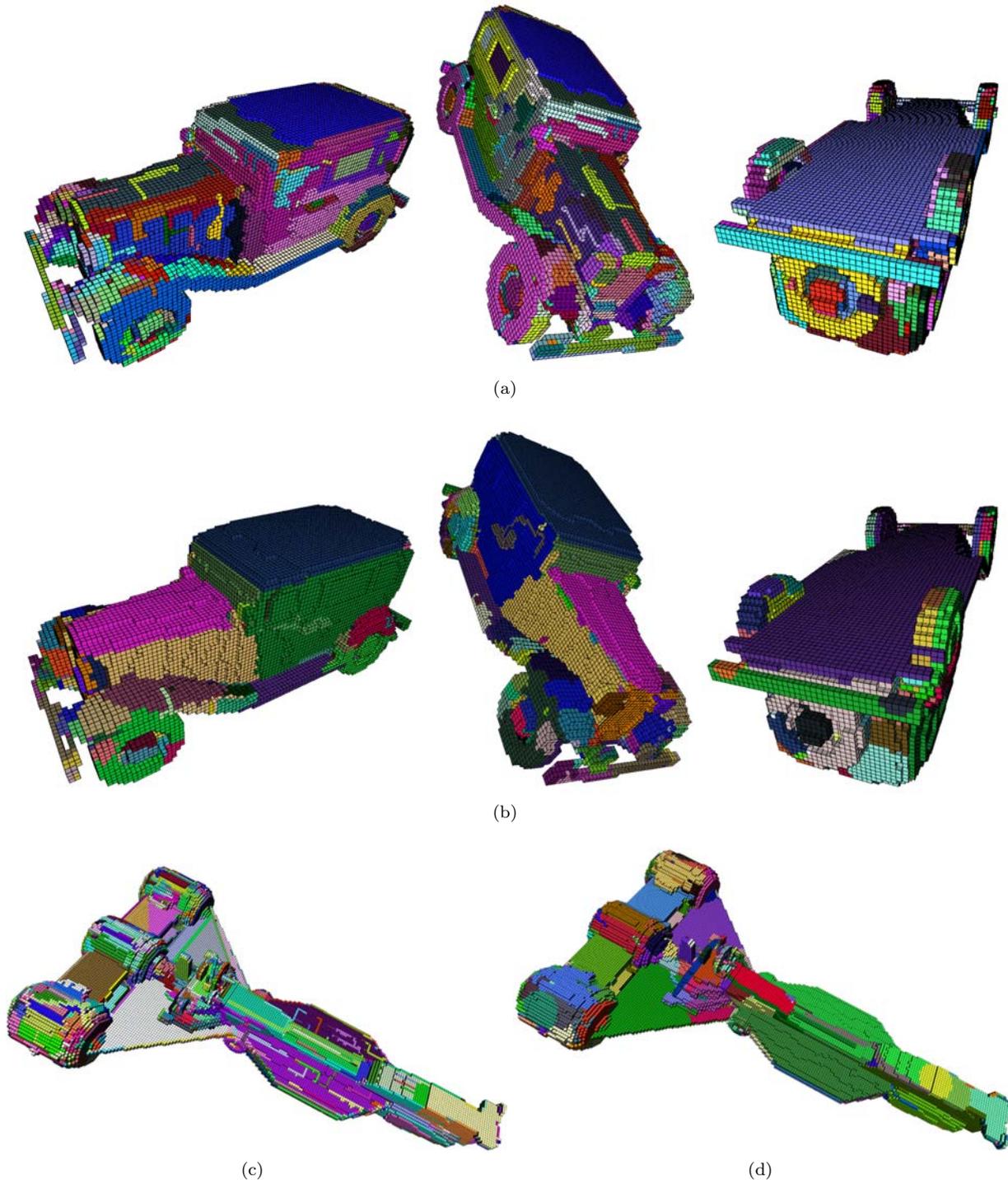


Figure 5.4 – (a) Différentes vues d'une voiture segmentée en un ensemble de morceaux de plans discrets naïfs avec l'algorithme DSD proposé par I. Sivignon. (b) La même voiture segmentée en morceaux flous de plans discrets d'épaisseur 2 avec la méthode hybride. (c) L'algorithme DSD appliqué sur un vaisseau spatial et (d) sa version segmentée avec l'approche hybride à l'épaisseur 2.

indiquent des déformations plus prononcées du bord de l'objet autour du point p . Par conséquent, étant donné une valeur seuil σ et un réel ν , l'idée est de développer, à partir d'une graine g , un MFPD $_{\nu}$ en suivant la stratégie d'extension maximale lorsque $|\mathcal{F}_{\nu}(g)| < \sigma$ et un patch $\Gamma_{\nu}(g)$ dans les autres cas. D'où l'algorithme 5.2.

Les résultats de cette méthode hybride sont présentés sur les figures 5.3 et 5.4. À la figure 5.3 nous proposons des objets discrets synthétisés de différentes formes, bruités et non bruités, et segmentés à des épaisseurs raisonnables. Comme nous pouvons le constater, le comportement désiré est obtenu et nous retrouvons les faces plates naturelles des cubes et de l'ellipsoïde à travers les segments résultants. De plus, dans les parties courbées, l'utilisation des patches discrets permet d'éviter les morceaux de plans très fins venant s'imbriquer dans les autres segments déjà développés. Nous pouvons aussi noter qu'avec de petites épaisseurs ν la méthode reste applicable sur des objets non bruités.

Cette stratégie hybride a aussi été appliquée sur quelques objets réels scannés. Les résultats sur ces derniers sont représentés aux figures 5.4(b) et (d), et les mêmes objets segmentés avec des morceaux de plans discrets naïfs, grâce à l'algorithme DSD¹², sont proposés comme référence.

Nous pouvons observer l'intérêt d'utiliser des MFPD pour simplifier la segmentation au niveau des zones plates des objets (en perdant possiblement la réversibilité) puisque nous retrouvons de grands segments décomposant naturellement le toit, le pare-brise et le capot de la voiture, par exemple. Dans les zones non plates des objets, il est difficile d'être catégorique sur les résultats, n'ayant pas de critère permettant de définir ce qu'est vraiment une « bonne » segmentation en plans discrets dans ces configurations.

Ces résultats ont fait l'objet d'une publication avec présentation orale à la conférence International Workshop on Combinatorial Image Analysis en 2008 [PDR08b].

5.3 Lissage

Nous proposons dans cette section une application de la procédure de segmentation, décrite à la section précédente, au lissage d'objets discrets bruités. Cette première approche, de par sa nature, ne fonctionne que pour des objets discrets convexes mais les résultats sont encourageants et méritent d'être présentés puisque bon nombre d'objets mathématiques possèdent cette propriété. De plus, en géométrie discrète, peu d'études ont été jusqu'à présent menées sur ce type de problématique en 3D.

Nous trouvons néanmoins dans [SdBS00] une approche de lissage, fonctionnant sur tout type d'objets discrets, exploitant la transformée en distance. Le but est de supprimer les protubérances, les cavités et les composantes de taille négligeable de l'image. Pour cela la transformée en distance de l'image 3D, c'est-à-dire de l'objet et de son complémentaire, est calculée. Puis les voxels ayant une distance inférieure à un certain seuil, donné en paramètre à la méthode, sont mis à zéro. Ceci crée une sorte de « zone de vide » autour du bord de l'objet. Le lissage consiste alors à calculer la transformée en distance de cette zone de vide et à définir comme voxels du bord les voxels à égale distance de l'objet et de son complémentaire.

Notre méthode exploite directement les résultats obtenus avec le procédé de segmentation présenté à la section précédente. Les étapes pour lisser un objet discret bruité \mathcal{O}_b peuvent être résumées de la façon suivante :

1. Calculer une segmentation de \mathcal{O}_b en MFPD $_{\nu}$. Soit L_S la liste des segments obtenus ;
2. Pour chaque segment $S_i \in L_S$, ($1 \leq i \leq |L_S|$), choisir un plan euclidien P_i représentant de S_i ;
3. Soit B_i l'ensemble des points entiers qui appartiennent au demi-espace négatif délimité par P_i . Calculer l'objet lissé résultant $\mathcal{O}_{\ell} = \cap_i B_i$.

Le point 3 définit la version lissée de l'objet comme étant la discrétisation d'un ensemble convexe. Ceci explique pourquoi cette approche n'a de sens que si elle est employée sur des objets dont le résultat attendu est un objet convexe.

Par ailleurs, comme nous utilisons des morceaux plans discrets épais, il n'existe pas un unique choix possible pour le plan euclidien représentant de l'étape 2. Dans notre cas, si (a, b, c, μ, ω) représentent les

12. <http://liris.cnrs.fr/isabelle.sivignon/DSD.html>

caractéristiques du morceau flou de plan discret d'épaisseur ν associé à un segment S de la segmentation, nous avons choisi comme représentant euclidien de S le plan d'équation $ax + by + cz = \mu + \frac{\omega}{2}$, c'est-à-dire le plan euclidien se situant à mi-distance des deux plans d'appui de S . Ceci a un sens, puisque les objets que nous traitons dans cette section sont des polyèdres discrets mathématiques synthétisés et bruités avec le modèle de bruit de Kanungo (voir annexe A.2) en prenant des paramètres α et β identiques. Or dans ce cas, le bruit créé est symétrique par rapport au bord de l'objet. Malgré tout, le choix du représentant euclidien reste pour l'instant un problème dans un cas général, comme nous le discuterons dans la section suivante.

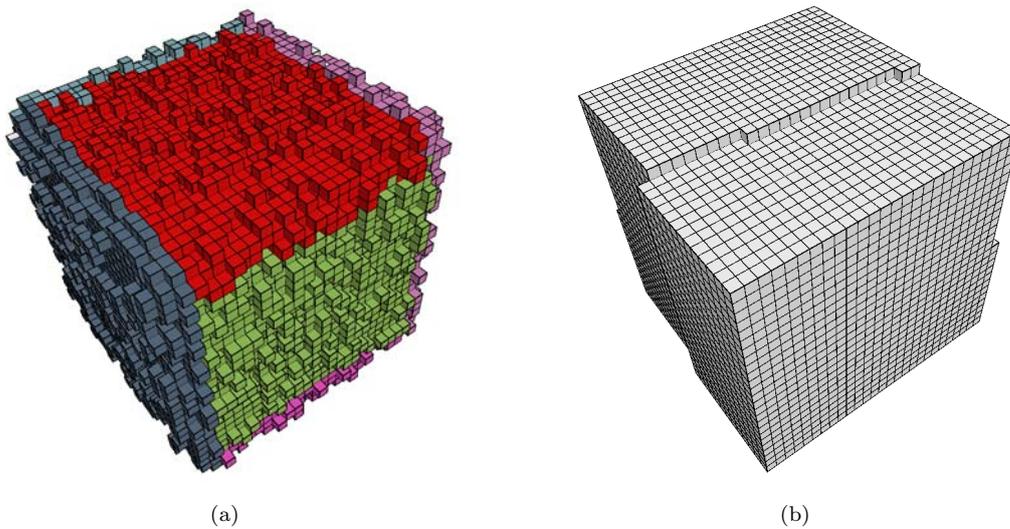


Figure 5.5 – (a) Un exemple d'une segmentation non désirable pour une application au lissage. Des points proches d'une arête du cube ont été ajoutés au segment rouge, du fait de la stratégie d'extension maximale, ce qui a donné naissance à une marche d'escalier au niveau de (b) la version lissée du cube.

La figure 5.5 montre le résultat d'une opération de lissage sur un cube bruité. Le cube a été synthétisé de façon à voir ses faces parallèles aux plans principaux de l'espace, et nous l'avons ensuite bruité selon le modèle de Kanungo. Comme nous pouvons le remarquer, la version lissée n'est pas totalement satisfaisante puisque nous notons une marche d'escalier sur certaines faces du cube. Cela s'explique par la stratégie d'extension maximale utilisée pour développer les $MFPD_\nu$. En effet, rien n'interdit dans ce cas l'ajout de points se trouvant dans les zones de discontinuités des objets (en l'occurrence, au niveau des arêtes du cube). Si tous les points de la zone plate font déjà partie du $MFPD_\nu$ en cours de construction et que les seuls candidats sont les points proches des arêtes de l'objet, alors ces derniers sont ajoutés – s'ils n'impliquent pas un dépassement de l'épaisseur limite ν – et tendent à faire « pencher » le morceau flou de plan discret. Les caractéristiques de ce dernier sont donc moins représentatives de la face du cube qu'elles représentent. C'est par exemple le cas du segment rouge de la figure 5.5(a). Dans le but d'éviter ce genre de scénario, nous devons donc nous assurer que les segments obtenus sont suffisamment fiables pour pouvoir les utiliser dans un cadre de lissage. Il semble donc judicieux de ne pas ajouter les points se situant dans les zones de l'objet considérées comme non plates. Afin de détecter ces points, nous pouvons utiliser l'estimateur de forme $\mathcal{F}_\nu(\cdot)$. Comme les valeurs croissantes de $\mathcal{F}_\nu(p)$ indiquent des déformations plus prononcées du bord de l'objet autour du point p , nous n'ajoutons pas les points dont le facteur de forme est inférieur à un seuil σ fixé. L'estimation de $\mathcal{F}_\nu(\cdot)$ sur le bord d'un dodécaèdre discret bruité est illustrée sur la figure 5.6(c) à travers une échelle de couleur allant du vert dans les zones plates, au rouge dans les zones bombées. Et nous pouvons observer, sur la figure 5.6(d), une décomposition en segments *fiables* obtenue lorsque nous n'ajoutons pas les points se situant au niveau des discontinuités de l'objet,

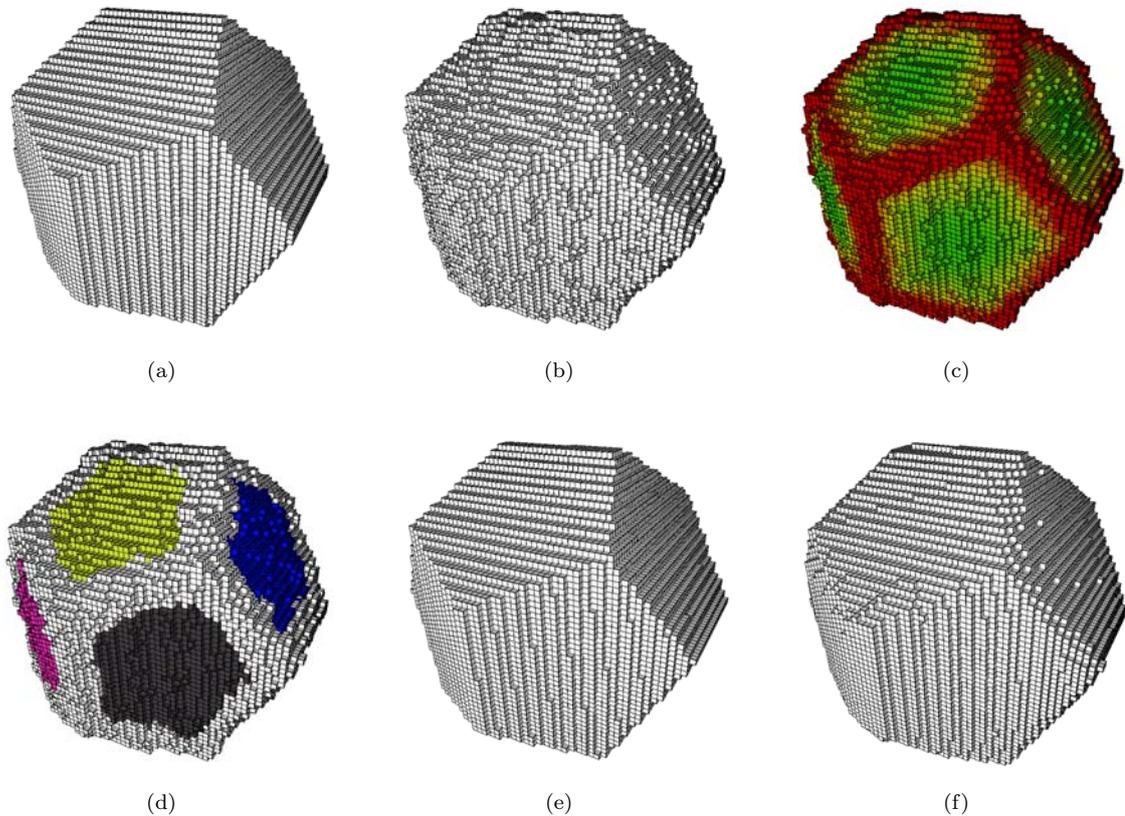


Figure 5.6 – Les différents états d’un objet discret. (a) Un dodécaèdre discret régulier et (b) sa version bruitée selon le modèle de bruit de Kanungo. (c) Une coloration du bord du dodécaèdre bruité représentant l’estimateur de forme calculé sur ce dernier – un dégradé allant du vert, pour les zones plates, au rouge pour les zones bombées. (d) Les segments *fiabiles* de la décomposition en MFPD $_{\nu}$. (e) L’objet lissé résultant et (f) une discrétisation de l’enveloppe convexe de la version bruitée.

c’est-à-dire dans les zones rouges.

La figure 5.6 résume en images les différents états d’un objet discret, de sa version originale en (a) à sa version bruitée en (b), jusqu’à sa version lissée en (e). Nous pouvons voir que nous retrouvons les douze faces canoniques du dodécaèdre et que la version lissée obtenue est très proche de la version originale de l’objet. Nous proposons, comme référence, une discrétisation de l’enveloppe convexe de la version bruitée de l’objet, que nous pouvons aussi considérer comme un lissage.

En comparaison, la version issue de notre procédé est plus régulière. Ceci est tout à fait logique, puisque l’enveloppe convexe est composée d’un plus grand nombre de faces, non coplanaires, du fait de la présence de bruit. De plus la discrétisation de l’enveloppe convexe est un sur-ensemble « par excès » de l’objet discret, donc plus volumineux, ce qui n’est pas le cas de notre approche, grâce au choix plus judicieux du représentant euclidien.

D’autres polyèdres classiques discrets, bruités ont été lissés grâce à cette méthode et les résultats sont présentés à la figure 5.7. Comme nous pouvons le constater, le nombre de segments fiables trouvé correspond au nombre de faces canoniques de chaque objet, nous permettant de retrouver dans chaque cas une version lissée cohérente avec l’objet initial. Pour l’instant le seuil σ est fixé manuellement.

Un article décrivant ces travaux a été publié dans le journal Pattern Recognition en 2009 [PDR09].

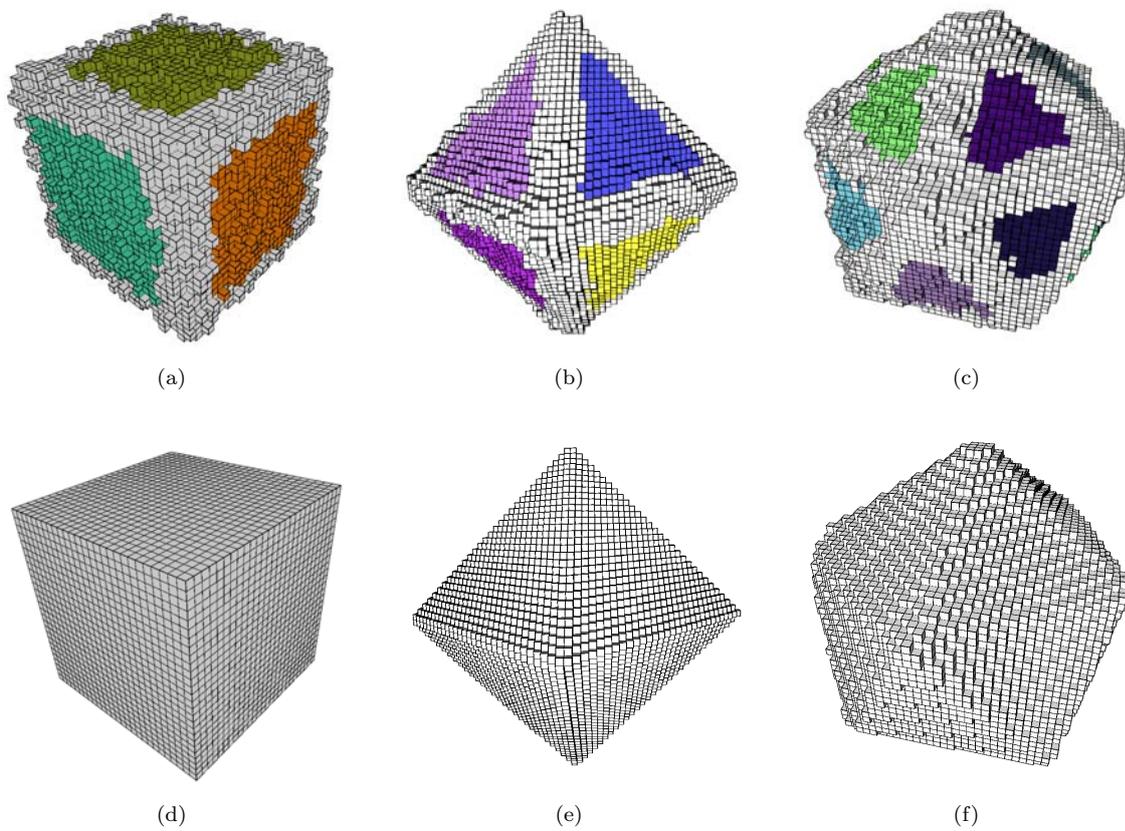


Figure 5.7 – Différents objets discrets convexes, bruités, et les segments fiables de la décomposition de leur bord en $MFPD_\nu$: (a) un cube, (b) un octaèdre, (c) un icosaèdre et (d), (e), (f) leur version lissée respective.

5.4 Vers la polyédrisation d'objets discrets bruités 3D

Le problème de la polyédrisation d'objets discrets bruités ayant été le fil conducteur de cette thèse, nous ne pouvions la terminer sans évoquer les difficultés rencontrées pour le résoudre avec l'approche que nous avons envisagée. Même si nous ne donnons aucune solution concrète au problème de la polyédrisation, nous présentons dans la suite de cette section des tests préliminaires ainsi que des idées pour avancer sur la résolution de cette problématique.

Une première réponse pourrait découler de l'algorithme de lissage décrit précédemment, en l'étendant au cas des objets discrets non convexes. Nous décrivons dans la section suivante une procédure pour concrétiser cette extension, puis nous reviendrons dans la section d'après aux problèmes liés à la polyédrisation.

5.4.1 Lissage d'objets discrets bruités non convexes

Nous proposons ici une idée, pour l'instant non concrétisée, permettant de lisser des objets discrets bruités non convexes. Celle-ci repose sur le point de vue suivant : pour un objet discret non convexe \mathcal{O} , sa version lissée peut être considérée comme l'objet issu de la différence entre la discrétisation lissée de l'enveloppe convexe de \mathcal{O} et les cavités lissées de \mathcal{O} .

Plus précisément, la procédure de lissage d'un objet discret bruité non convexe \mathcal{O} serait la suivante : dans un premier temps, l'enveloppe convexe discrète pleine de \mathcal{O} , que nous noterons $\mathcal{CH}_d(\mathcal{O})$, est calculée. Les cavités de \mathcal{O} sont extraites comme différence entre $\mathcal{CH}_d(\mathcal{O})$ et \mathcal{O} . Si aucune cavité n'est trouvée, l'objet est considéré comme convexe (dans le sens où sa version lissée peut-être obtenue en appliquant la méthode de lissage expliqué précédemment). Sinon, une segmentation de $\mathcal{CH}_d(\mathcal{O})$ est effectuée, grâce à l'algorithme de segmentation hybride 5.2 par exemple, et seules les composantes connexes, formées par les cavités, qui n'appartiennent pas totalement aux segments issus de la décomposition de $\mathcal{CH}_d(\mathcal{O})$ en MFPD_ν sont conservées. Soit EC l'ensemble de ces composantes connexes conservées. Chaque élément de EC est alors considéré comme un nouveau objet discret bruité, et la procédure décrite ci-dessus est appliquée récursivement. Elle s'arrête donc quand les décompositions ne sont formées que d'objets discrets assimilés convexes (toujours dans le sens où nous pouvons le lisser avec la méthode présentée dans la section précédente). Pour un objet discret bruité non convexe \mathcal{O} , dont l'ensemble des composantes EC ont été lissées, nous pouvons alors extraire une version lissée de \mathcal{O} de la manière suivante : $\mathcal{CH}_d(\mathcal{O})$ (qui est un objet convexe) est lissée, toujours grâce à la méthode de lissage de la section 5.3. L'union de toutes les composantes lissées de EC est alors effectuée, et la version lissée de \mathcal{O} est obtenue comme différence entre la version lissée de $\mathcal{CH}_d(\mathcal{O})$ et l'union résultante de l'étape précédente.

Afin de mieux appréhender cette procédure, une illustration sur un objet discret bruité 2D non convexe est proposée à la figure 5.8. Les étapes sont numérotées de la façon suivante :

- | | |
|---|--|
| ① un objet discret initial \mathcal{O} ; | ④ la segmentation de $\mathcal{CH}_d(\mathcal{O})$; |
| ② son enveloppe convexe ; | ⑤ la version lissée de $\mathcal{CH}_d(\mathcal{O})$; |
| ③ la discrétisation $\mathcal{CH}_d(\mathcal{O})$ de son enveloppe convexe pleine ; | ⑥ la version lissée de \mathcal{O} . |

Les lettres accolées au numéro permettent de différencier les composantes de l'objet. Par exemple ②_{ca} représente l'enveloppe convexe de la première (en fait la seule ici) sous-composante de la troisième composante de l'objet initial. La décomposition de l'objet est représentée par le symbole \oplus et l'opération de différence entre les objets par le symbole \ominus .

Dans cette illustration, certaines étapes ③, ④ et ⑤ ne sont pas représentées car les objets sont convexes et tous leurs points appartiennent aux segments flous d'épaisseur 2 calculés lors de leur segmentation. On considère donc que leur version lissée est l'objet lui-même.

Bien sûr, une étude plus poussée de cette idée est nécessaire afin de mieux en cerner le comportement. On peut par exemple se demander si les opérations de différences ne vont pas engendrer des déconnexions de l'objet. Comment la segmentation influence le résultat ? Doit-on conserver une épaisseur ν identique

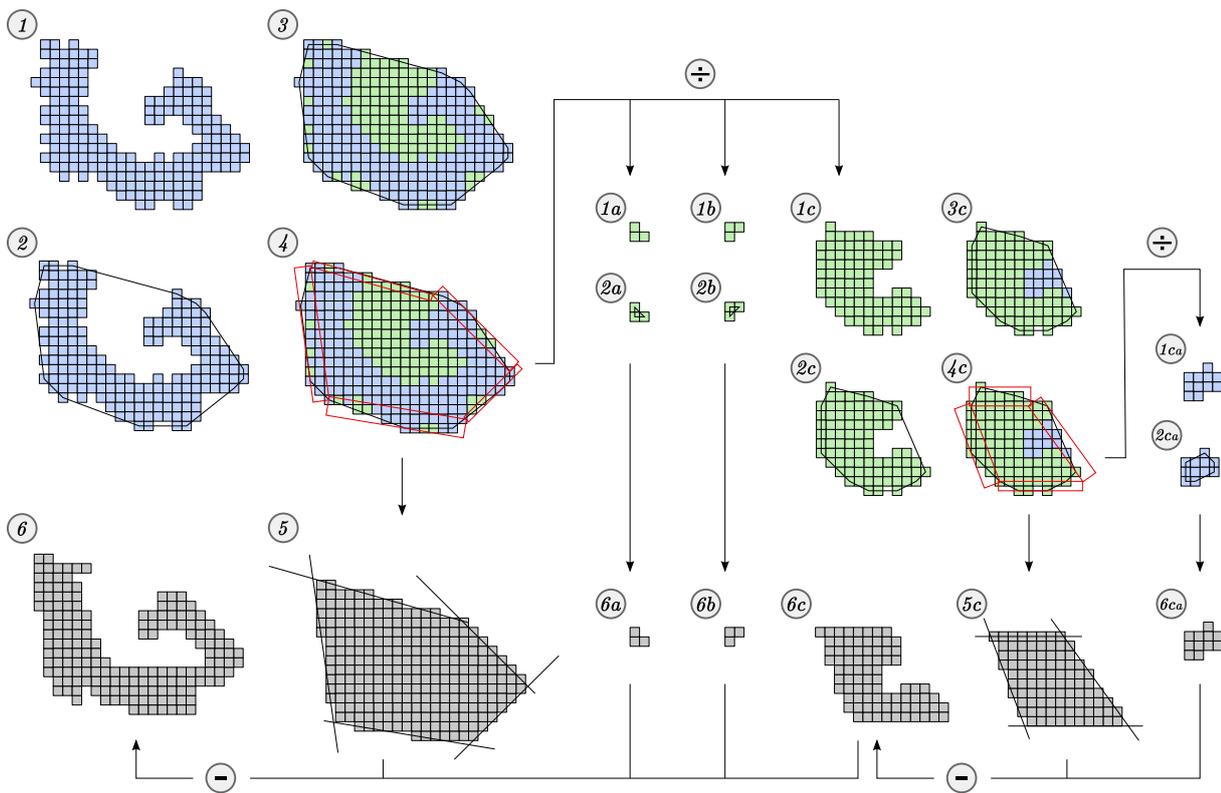


Figure 5.8 – Illustration de l'idée d'une procédure de lissage pour un objet non convexe. L'objet est décomposé récursivement en sous-objets, en fonction de ses cavités, jusqu'à l'obtention d'un objet convexe. La procédure de lissage est alors appliquée et l'objet résultant est défini comme la différence entre la discrétisation de son enveloppe convexe et des sous-objets lissés dont il est composé.

tout au long de la procédure? etc... Néanmoins cette piste semble intéressante à investiguer et pourrait permettre de mettre au point un algorithme d'approximation polyédrique pour des objets discrets bruités.

5.4.2 Idées pour des perspectives de polyédrisation

Si nous examinons la méthode décrite précédemment, nous pouvons remarquer que dans les étapes ⑤ les objets sont convexes et leur lissage est défini comme la discrétisation d'une intersection de demi-espaces. Ces derniers sont délimités par les représentants euclidiens issus de la segmentation. Dans ce cas nous avons bien une approximation polyédrique de l'objet. On pourrait alors appliquer la même démarche, en considérant cette fois-ci les représentations polyédriques des objets et en effectuant des différences de polyèdres.

Naturellement, rien qu'à l'évocation de ce procédé, nous voyons déjà quelques difficultés surgir. Par exemple, dans la phase de calcul de la différence de deux polyèdres, les bords des polyèdres en question ne s'intersectent pas forcément, comme nous pouvons le voir à la figure 5.9(a). Dans ce cas, le polyèdre résultant, au lieu d'être non convexe et composé d'une seule surface, devient un polyèdre avec une cavité interne, ce qui n'est pas le résultat désiré. La zone entourée en rouge dans cette figure ne devrait pas exister. Une solution pourrait être de détecter ces zones sensibles au préalable, dans l'étape ④ de segmentation, comme le montre la figure 5.9(b). En marquant les points des cavités appartenant aux segments issus de la phase décomposition en $MFPD_\nu$, nous pourrions éviter de considérer les (parties de) représentants euclidiens en rapport avec ces points et envisager des intersections comme celle illustrée dans la figure 5.9(b) – ce qui reste une tâche ardue car les intersections entourées en rouges ne sont pas évidentes à déterminer, d'autant plus en 3D.

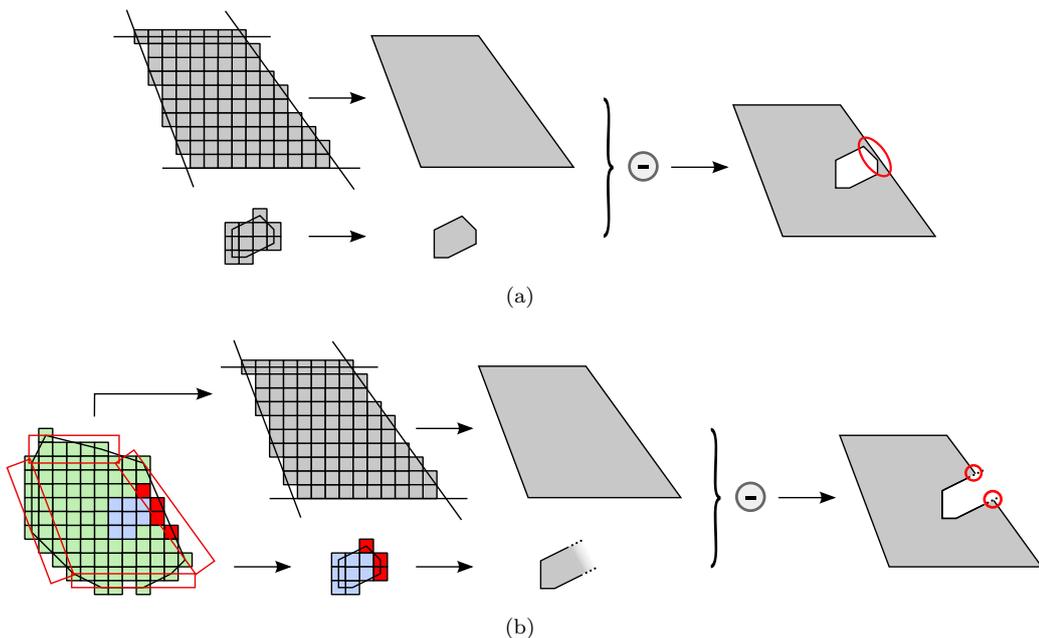


Figure 5.9 – (a) Exemple d'un cas non souhaité lorsqu'on calcule la différence de deux polyèdres. La zone entourée en rouge ne devrait pas exister. (b) Tentative de détection des zones sensibles. Les intersections entourées restent difficile à déterminer.

Une autre possibilité pour attaquer le problème de la polyédrisation serait de revenir à l'approche initiale, sans se reposer sur la notion de *segments fiables*, qui consiste à extraire une représentation polyédrique d'un objet discret à partir d'une segmentation du bord de ce dernier en un ensemble de morceaux flous de plans discrets d'épaisseur ν . L'idée est que chaque $MFPD$ issu de la segmentation va engendrer une face dans la représentation polyédrique de l'objet discret. Il reste alors à « recoudre » les faces entre elles pour obtenir la représentation finale recherchée. Bien que la formulation des étapes

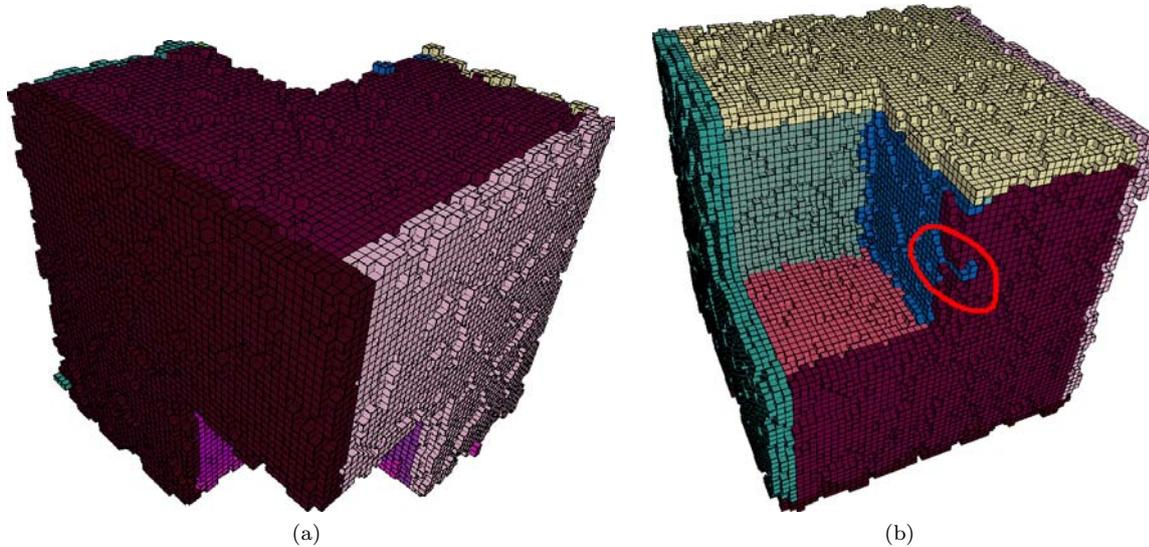


Figure 5.10 – Deux vues d'un objet polyédrique. Dans l'ensemble, le bord de l'objet est segmenté de manière convenable, en respectant les facettes naturelles de celui-ci, néanmoins la partie entourée en rouge dans la vue (b) montre un isthme dû au bruit, rendant irrégulier la limite entre les deux segments.

à suivre reste plutôt basique, il en est tout autre des questions soulevées pour les franchir. En effet, la segmentation préalable du bord de l'objet discret est effectuée en utilisant des morceaux flous de plans discrets d'une certaine épaisseur, ce qui induit des problèmes, notamment au niveau des contours des segments obtenus lors de la décomposition en $MFPD_\nu$. La figure 5.10(b) montre un de ces problèmes à la jointure des segments bleu et bordeaux, où, à cause du bruit, un isthme du premier pénètre dans le second.

Pour tenter « d'absorber » ces perturbations, nous pourrions envisager une approche « flou total » qui consisterait à définir des notions d'*arêtes floues* (basées sur les segments flous 3D par exemple) et de *sommets flous*. Ainsi, deux faces seraient *cousues* entre-elles par une arête floue reconnue sur les points adjacents des contours de ces deux faces. Ces *coutures floues* engendreraient un sorte de « zone floue » tout autour du bord de l'objet discret, mais pour laquelle nous pourrions contrôler l'approximation grâce au paramètre ν . Se pose alors la difficile question du choix des représentants euclidiens pour chacune des entités floues ? Ces représentants seraient à choisir à l'intérieur de la zone floue, mais cette zone étant épaisse, de nombreux choix seraient possibles.

Nous pourrions aussi songer à revenir à des morceaux de plans naïfs après la phase de segmentation, et tenter d'appliquer les algorithmes déjà existants dans le domaine, même si certaines difficultés restent présentes. Nous pouvons par exemple voir sur la figure 5.11(a) un cube discret bruité, segmenté en morceaux flous de plans discrets d'épaisseur 3. La figure 5.11(b) représente ce même cube lorsque nous remplaçons chaque MFPD par son plan englobant optimal épais, et la figure 5.11(c) représente leur homologues naïfs. Nous voyons ici l'exemple d'un choix non satisfaisant (nous avons choisi, arbitrairement, la discrétisation du plan d'appui supérieur des morceaux de plans épais) produisant des trous dans l'objet. Il serait ici plus judicieux de faire un choix dépendant des convexités des intersections entre les morceaux de plans (dans une démarche similaire à [DRDR06], mais en 3D). Ceci ouvre un champ d'étude plutôt vaste sur les intersections de MFPD encore très mal connues et comprises pour l'instant.

5.5 Conclusion

Ce dernier chapitre a été l'occasion, à travers un amalgame des notions introduites dans les chapitres précédents, de proposer un processus de segmentation du bord d'objets discrets bruités, en morceaux

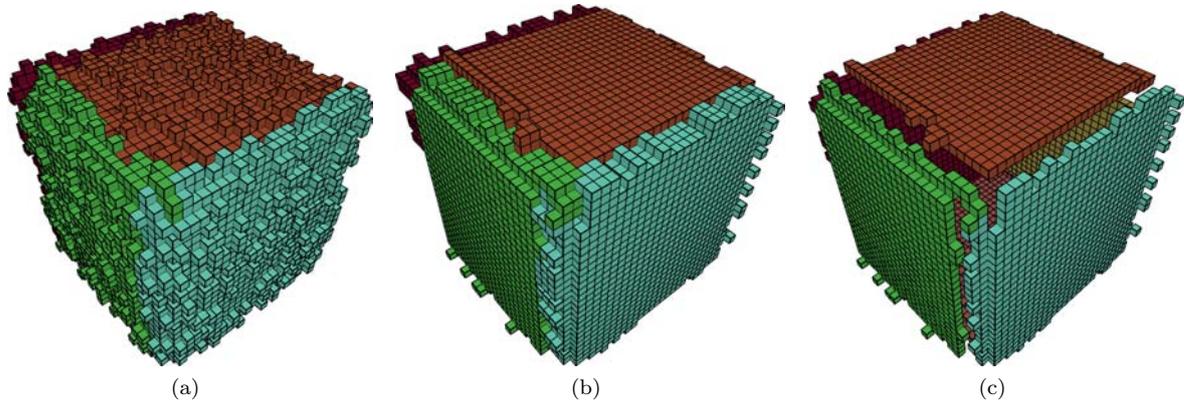


Figure 5.11 – (a) Un cube discret bruité segmenté à l'épaisseur 3, avec (b) la représentation épaisse de ses segments, et (c) une représentation à travers la discrétisation du plan d'appui supérieur des segments.

floos de plans discrets d'épaisseur ν . Cette décomposition a l'avantage d'être robuste à un certain niveau de perturbations, évitant ainsi les sur-segmentations observées en général lorsqu'on utilise des techniques basées sur des primitives discrètes régulières.

De plus, l'utilisation de caractéristiques géométriques, calculées au bord des objets discrets, autorise un guidage de la segmentation qui permet d'obtenir des résultats satisfaisants, aussi bien dans les zones plates que dans les zones courbées de l'objet. Pour les objets discrets polyédriques par exemple, l'ensemble des segments obtenu respecte les discontinuités naturelles des objets et caractérise ainsi relativement bien les faces canoniques de ces derniers.

Nous avons ensuite présenté un procédé de lissage d'objets discrets bruités exploitant la segmentation en MFPD_ν décrite auparavant. À l'aide de l'estimateur de forme, les segments sont réduits à un exemplaire fiable dont est extrait un plan représentant euclidien. L'objet lissé est alors obtenu comme discrétisation de l'intersection des demi-espaces négatifs délimités par ces représentants euclidiens. Pour cette raison, ce procédé a été concrétisé pour des objets discrets pour lesquels le résultat attendu est un objet convexe. Néanmoins, nous avons proposé des idées pour étendre celui-ci au cas d'objets non convexes.

Enfin, sans aborder le problème de la polyédrisation de manière concrète, nous avons exprimé différentes idées et perspectives de travaux pouvant amener à des résultats intéressants, en exposant malgré tout certaines difficultés que nous pourrions rencontrer.

Conclusion et perspectives

Ce manuscrit touche presque à son terme, et il convient à présent de récapituler les contributions apportées par cette thèse et de présenter les perspectives de travail qui nous semble intéressantes à poursuivre.

1 Contributions

Les études effectuées au sein de cette thèse ont été motivées par le constat suivant : l'approche arithmétique de la géométrie discrète conduit à la définition de primitives linéaires discrètes d'une régularité parfaite. Or, les procédés d'acquisitions, eux, ne sont pas parfaits. Que les aléas d'expérimentations soient dus à une partie du capteur défaillante ou aux milieux pas toujours totalement contrôlés dans lesquels elles s'effectuent, les résultats produits sont généralement empreints d'une certaine incertitude conduisant potentiellement à des objets non réguliers. Suivant l'amplitude des irrégularités, traiter ces objets avec des primitives discrètes « parfaites » peut conduire à des résultats non satisfaisants. C'est en particulier le cas pour la segmentation d'objets discrets en morceaux de plans discrets, comme nous l'avons déjà illustrée. Le travail prospectif mené tout au long de cette thèse a donc eu pour but d'essayer d'apporter une réponse à ce problème.

L'apport principal de nos travaux réside donc dans la définition et l'utilisation de primitives floues, destinées à apporter plus de souplesse dans les traitements effectués sur des objets discrets. Nous avons pour cela défini la nouvelle notion de *morceau flou de plan discret d'épaisseur ν* . Cette primitive généralise la notion 2D de segment flou de droite discrète et permet, grâce à son paramètre d'épaisseur ν , de contrôler la tolérance aux perturbations apparaissant sur les objets.

Nous avons aussi mis au point deux algorithmes incrémentaux de reconnaissance de morceaux flous de plans discrets d'épaisseur ν . L'un, utilisant des techniques de programmation linéaire, permet de résoudre le problème en considérant une définition des MFPD $_{\nu}$ suivant la norme infinie. L'autre, considérant cette fois-ci une définition suivant la norme euclidienne, tire parti de la géométrie de l'enveloppe convexe des points à reconnaître pour résoudre le problème. Une étude du comportement « en pratique » du dernier algorithme a été menée afin de déceler les améliorations possibles. Les codes sources de ces algorithmes sont disponibles à l'adresse suivante <http://www.loria.fr/~debled/BlurredPlane/>.

Doté de cette primitive plus souple, nous avons proposé différents estimateurs de caractéristiques géométriques au bord d'objets discrets : des estimateurs de vecteurs normaux, que nous avons comparé à certains autres déjà existants dans le domaine, ainsi qu'un estimateur de forme permettant de discriminer les zones plates, bombées et creuses du bord d'un objet discret. Nous avons pour cela envisagé une approche directionnelle utilisant les contours des tranches des objets discrets, ainsi qu'une approche multi-directionnelle reposant sur la notion de patch d'épaisseur ν , décrivant un voisinage adaptatif autour des points du bord d'objets discrets.

Nous avons finalement proposé une technique de segmentation du bord d'un objet discret en morceaux flous de plans discrets. En exploitant l'estimateur de forme établi précédemment, nous avons mis au point une stratégie hybride permettant d'obtenir des résultats convenables même sur les zones courbées

du bord des objets. L'exploitation de cette segmentation nous a permis de tenter une première approche du problème de lissage d'objets discrets bruités, convenant pour l'instant aux objets convexes. Nous avons, pour terminer, discuté des difficultés rencontrées pour élargir cette application au problème de la polyédricisation d'objets discrets, laissant entrevoir un champ d'investigations fertile.

Des perspectives à ce travail sont d'ailleurs présentées dans la section suivante.

2 Perspectives

De mon point de vue, les travaux effectués durant ma thèse ont été prospectifs et nous ont permis d'entrevoir les difficultés qui se présentent lorsqu'on emploie des primitives épaisses pour traiter des problèmes comme la segmentation ou la polyédricisation d'objets discrets bruités. Avant même de continuer plus l'exploration de ces problématiques, j'aimerais revenir sur certains travaux et les améliorer.

Le premier concerne la reconnaissance des morceaux flous de plans discrets. Cette étape est le cœur de tous les travaux que nous avons menés. Il est par conséquent primordial qu'elle soit parfaitement comprise et contrôlée, tout en restant la plus efficace possible. Or de nombreuses améliorations sont possibles à l'heure actuelle. Nous avons en effet proposé, à la section 3.5.4, quelques idées pour optimiser la reconnaissance de MFPD_ν et il me tient à cœur de les concrétiser.

De plus, bien que nous ayons proposé une définition des MFPD_ν relative au choix d'une norme, nous avons uniquement utilisé l'algorithme qui considère la norme euclidienne dans la suite de nos travaux. Il serait intéressant de comparer les résultats obtenus en choisissant différentes normes, afin de voir les impacts sur les méthodes qui emploient une reconnaissance de MFPD .

Une autre perspective concerne l'estimation de caractéristiques géométriques au bord d'objets discrets bruités. Nous avons, entre autres, proposé un estimateur de forme permettant de distinguer les zones plates, bombées et creuses du bord d'un objet discret, en exploitant les variations angulaires entre les vecteurs normaux de patches « voisins ». Or, en général, la notion de courbure permet de faire cette distinction. Il serait donc intéressant de faire ressortir le lien, s'il existe, entre notre estimateur et une notion de courbure sur une surface discrète.

D'autre part, pour estimer le champ de vecteurs normaux au bord d'un objet discret, nous avons considéré une notion de plan tangent en développant en chaque point du bord un patch discret d'épaisseur ν . Chaque point du bord fait alors partie de plusieurs patches. Est-ce que ceci ne pourrait pas correspondre à la généralisation aux surfaces de la notion de couverture tangentielle développée en 2D ? Il serait intéressant d'étudier plus en détail cette question car la couverture tangentielle en 2D est un outil important pour l'étude des courbes discrètes et intervient dans le processus de segmentation de ces dernières.

Plus généralement, les estimateurs mis au point devront être testés et validés sur un plus grand panel d'objets discrets. Une démarche expérimentale soignée devra alors être mise en place et des critères d'évaluation devront être définis. Une possibilité serait de partir d'une forme continue pour laquelle nous connaissons l'expression analytique de la caractéristique à évaluer. Après discrétisation et perturbation de cette forme, l'application des estimateurs proposés serait effectuée et comparée aux résultats théoriques. Une étude de la convergence asymptotique pourra aussi être menée, en considérant une discrétisation de l'objet sur une grille de plus en plus fine. Néanmoins ces évaluations ne peuvent pas se faire sans une caractérisation formelle de la notion de bruit sous-jacente à l'étape de perturbation des objets. Nous avons dans cette thèse considéré des perturbations aléatoires, ou encore issues d'un modèle conçu pour la 2D, mais ceci reste discutable. Une démarche plus audacieuse serait de définir un modèle de bruit ayant une vraie réalité. Ceci nécessiterait probablement de s'intéresser aux méthodes (voire aux appareils) d'acquisition d'images 3D.

Comme nous l'avons vu, la sensibilité au bruit est contrôlée grâce au paramètre ν associé aux primitives floues. Or, dans nos expérimentations, ce paramètre est fixé manuellement. Néanmoins, ne pourrait-on pas trouver un moyen de le fixer automatiquement ? Pour le moment, cette question est restée en sus-

pens, et dans l'absolu, il semble difficile d'en donner une réponse. Néanmoins, avec une connaissance du contexte, et notamment celui lié au bruit, on pourrait extraire des informations utiles pour que ce choix automatique de ν puisse être fait.

Une autre étude semble aussi incontournable dès lors que nous désirons aborder la problématique de la polyédrisation d'objets discrets bruités avec plus de facilité (et dans un formalisme peut-être plus adapté qu'il ne l'est à présent), c'est celle de l'intersection de deux MFPD. En effet, avec l'approche employée qui consiste à baser la polyédrisation sur une segmentation préalable du bord de l'objet discret en MFPD, comprendre comment s'intersectent les segments issus de cette décomposition permettrait probablement de faire un choix judicieux pour leur représentant euclidien. Une vaste voie d'étude s'ouvre alors, comprenant, entre autres :

- l'étude de l'intersection de plans discrets épais, puisque c'est sur la base de ces derniers (à travers la notion de plan englobant optimal) que sont définis les MFPD ;
- l'étude de l'influence du contour des morceaux dans ces intersections ;
- le contrôle des distances par rapport à l'objet discret initial ;

et bien d'autres sujets encore, particulièrement si l'on veut définir une notion d'arête flou à partir de ces intersections. De plus, dans le but de respecter les concavités et convexités des objets, un guidage utilisant l'estimateur de forme pourrait aussi être envisagé pour le calcul des intersections.

Annexe A

Perturbation d'objets discrets

Dans le cadre de cette thèse, nous avons toujours pris soin de valider les algorithmes proposés en les programmant et en les testant sur différents objets discrets. Nous avons été amenés à effectuer de nombreuses expérimentations, aussi bien sur des objets synthétisés, que sur des objets scannés issus du monde réel. Nous avons notamment utilisé des objets mathématiques, qui, par définition, étaient très réguliers (cubes, sphères, polyèdres réguliers, etc...). Afin de tester la flexibilité annoncée par nos méthodes, il a donc fallu perturber la surface de ces objets discrets. Pour cela, différentes méthodes ont été envisagées, et nous présentons ces dernières dans les sections suivantes.

A.1 Perturbations aléatoires

Nous considérons ici que nous avons en entrée un objet discret 6-connexe dont la surface en surfels est une variété combinatoire de dimension 2. L'idée est de générer de petites perturbations locales en parcourant chaque surfel s de la surface de l'objet et en choisissant (suivant un coefficient de probabilité donné) de rajouter ou non un nouveau voxel représentant un déplacement du voxel qui contient s , suivant une translation de vecteur \vec{n} , \vec{n} étant le vecteur normal élémentaire à s , comme indiqué à la figure A.1.

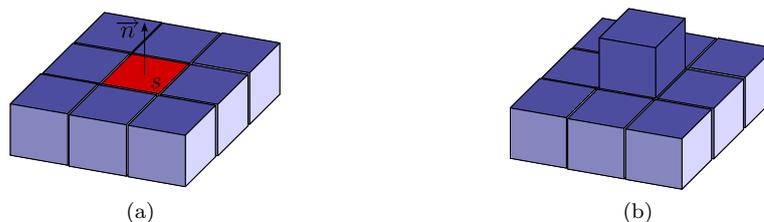


Figure A.1 – Configuration (a) avant et (b) après l'ajout du voxel en fonction du surfel rouge. Les surfels de la surface sont indiqués en bleu foncé.

Si nous désirons conserver une surface qui est une variété combinatoire de dimension 2, alors nous devons vérifier que les pointels et les lignels du nouveau voxel suivent bien les règles définies pour appartenir à une 2-variété combinatoire sans bord, à savoir :

- chaque lignel doit être incident à exactement deux surfels de la surface ;
- chaque pointel doit être incident à exactement une ombrelle de surfels de la surface.

Les seules configurations possibles d'ombrelles pour une 2-variété combinatoire sont rappelées à la figure A.2, conformément à [Fra95]. Si une configuration interdite est rencontrée, on choisit simplement de revenir à l'état précédent de l'objet.

Cette phase de perturbation peut alors être appliquée plusieurs fois sur l'objet suivant la « quantité d'irrégularités » désirée. Néanmoins, la méthode ne perturbe la surface qu'en déplaçant les voxels vers

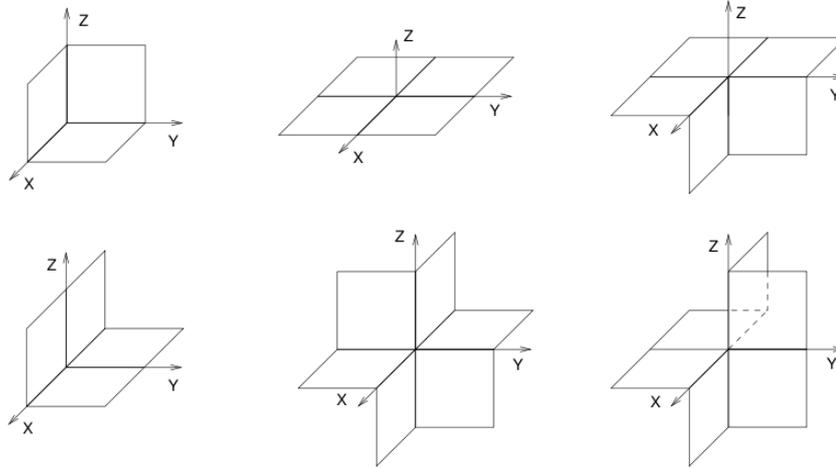


Figure A.2 – Configurations possibles d'ombrelles pouvant appartenir à une variété combinatoire de dimension 2 (aux symétries près).

l'extérieur de l'objet. Nous pourrions aussi choisir de déplacer les voxels vers l'intérieur ce qui produirait un bruit plus « symétrique ».

Néanmoins, afin d'avoir un algorithme de bruitage plus général, nous avons programmé une extension aux objets 3D d'une méthode de bruitage d'images discrètes, développée par T. Kanungo et ses coauteurs [KHP94], dans le contexte de l'OCR. Bien que ce modèle ait été validé en 2D dans ce contexte, son emploi en 3D reste discutable. Malgré tout, l'extension 3D ayant déjà été utilisée dans [SdBS00] et fournissant des résultats exploitables, nous la présentons dans la section suivante.

A.2 Modèle de Kanungo

La technique présentée ici repose sur un modèle de bruit défini par T. Kanungo et destiné à simuler les distorsions locales qui peuvent se produire lors des procédés de photocopie ou de scan. Le modèle proposé tient compte des inversions de pixels (dues aux fluctuations de la lumière, à la sensibilité des pixels et à la valeur de seuillage choisie) et du rendu flou qui apparaît à cause de la fonction d'étalement du point des systèmes optiques des scanners. Ce modèle a donc été initialement défini pour des images 2D, mais comme nous allons le voir, les concepts utilisés peuvent facilement s'étendre en 3D.

On considère ici des images binaires sans aucune contrainte sur l'objet représenté dans l'image. L'idée est de parcourir tous les voxels de l'image (objet plus fond) et de les inverser en suivant les règles suivantes : la probabilité d'inverser un *voxel fond* en *voxel objet* (et vice versa) est modélisée à l'aide d'une fonction exponentielle de sa distance par rapport au plus proche voxel objet (resp. voxel fond). Ainsi, la probabilité $P(o|b, d, \alpha_0, \alpha)$ d'inverser un voxel fond en un voxel objet est égale à $\alpha_0 e^{-\alpha d^2} + \eta$. Le paramètre α_0 représente la valeur initiale de l'exponentielle, et la croissance de l'exponentielle est contrôlée par le paramètre α . De manière similaire, la probabilité $P(b|o, d, \beta_0, \beta) = \beta_0 e^{-\beta d^2} + \eta$ d'inverser un voxel objet en un voxel fond est contrôlée par les paramètres β_0 et β . Le paramètre η représente quant à lui une probabilité constante d'inversion pour tous les voxels, quels qu'ils soient.

La distance d d'un voxel objet au plus proche voxel fond (ou vice versa) peut être calculée à l'aide de l'algorithme de transformée en distance (*DT*) proposé dans [Bor96]. Néanmoins, une version de cet algorithme permettant de calculer simultanément la transformée en distance des voxels objets et des voxels fonds a été proposée dans [SdBS00].

Enfin, pour prendre en compte les effets de la fonction d'étalement du point des systèmes optiques, une opération de fermeture morphologique est appliquée à l'objet [Ser82, NT08].

Ainsi, on peut résumer les différentes phases de l'algorithme de synthétisation d'objets bruités à partir d'un objet discret binaire initial de la façon suivante :

1. Calculer la distance d au bord de l'objet pour chaque voxel en appliquant un algorithme de transformée en distance ;
2. Inverser les voxels fonds avec la probabilité $P(o|b, d, \alpha_0, \alpha) = \alpha_0 e^{-\alpha d^2} + \eta$;
3. Inverser les voxels objets avec la probabilité $P(b|o, d, \beta_0, \beta) = \beta_0 e^{-\beta d^2} + \eta$;
4. Appliquer une opération de fermeture morphologique à l'objet.

Des résultats sont proposés dans les figures suivantes pour un demi-ellipsoïde creusé. Nous avons synthétisé des homologues bruités en faisant varier les différents paramètres de la méthode. La figure A.3 présente des versions bruitées du demi-ellipsoïde discret, synthétisées en utilisant différentes valeurs de α et β . $\alpha_0 = \beta_0 = 1$, $\eta = 0$ et nous avons choisi de fixer $\alpha = \beta$ afin de générer un bruit symétrique. Les figures A.4 et A.3(b) permettent de voir l'influence de la transformée en distance. Et enfin, la figure A.5 montre les résultats obtenus en utilisant différents éléments structurants, pour effectuer l'opération de fermeture morphologique.

Il est à noter qu'avec cette méthode la topologie des objets n'est pas contrôlée et l'image résultante peut contenir de nombreuses composantes connexes, tout spécialement avec de très petites valeurs pour α et β . Néanmoins, nous pouvons toujours extraire la composante connexe maximale et la considérer comme une version bruitée de l'objet initial.

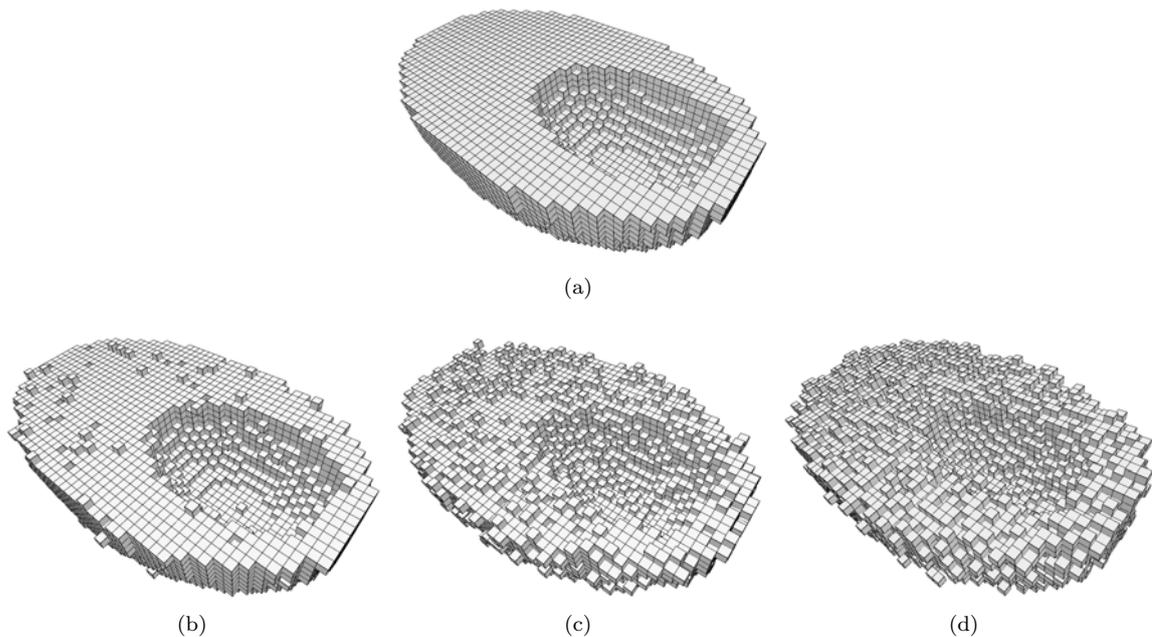


Figure A.3 – Plusieurs demi-ellipsoïdes creusés bruités générés à partir de l'objet (a) en utilisant le modèle de bruit de Kanungo avec une $DT = \langle 3, 4, 5 \rangle$ et (b) $\alpha = \beta = 0.4$, (c) $\alpha = \beta = 0.2$ et (d) $\alpha = \beta = 0.1$. L'élément structurant pour l'opération de fermeture morphologique est composé d'un voxel et son 6-voisinage.

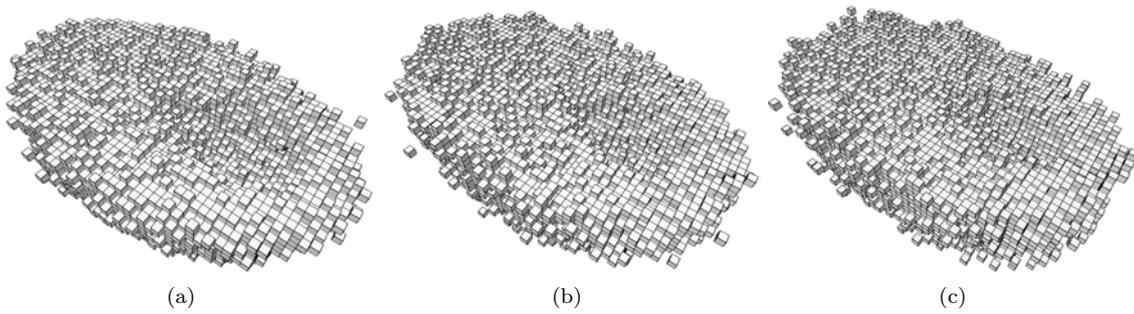


Figure A.4 – Plusieurs demi-ellipsoïdes creusés bruités générés à partir de l'objet de la figure A.3(a) en utilisant le modèle de bruit de Kanungo avec $\alpha = \beta = 0.4$ et (a) $DT^{26} = \langle 1, 1, 1 \rangle$, (b) $DT^{18} = \langle 1, 1, \infty \rangle$ et (c) $DT^6 = \langle 1, \infty, \infty \rangle$. L'élément structurant pour l'opération de fermeture morphologique est composé d'un voxel et son 6-voisinage.

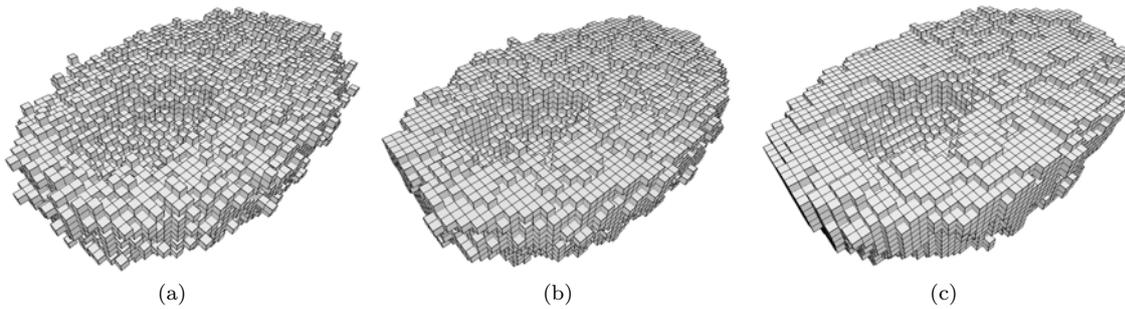


Figure A.5 – Plusieurs demi-ellipsoïdes creusés bruités générés à partir de l'objet de la figure A.3(a) en utilisant le modèle de bruit de Kanungo avec une $DT = \langle 3, 4, 5 \rangle$, $\alpha = \beta = 0.05$ et un élément structurant pour l'opération de fermeture morphologique composé d'un voxel et (a) son 6-voisinage, (b) son 18-voisinage et (c) son 26-voisinage.

Bibliographie

- [AAS97] E. ANDRES, R. ACHARYA et C. SIBATA : Discrete analytical hyperplanes. *CVGIP : Graphical Model and Image Processing*, 59(5):302–309, 1997.
- [ABP02] Z. AKTOUF, G. BERTRAND et L. PERROTON : A three-dimensional holes closing algorithm. *Pattern Recognition Letters*, 23(5):523–531, 2002.
- [And93] E. ANDRES : Le plan discret. In *Colloque de géométrie discrète en imagerie : fondements et applications*, Strasbourg, France, septembre 1993.
- [And02] Eric ANDRES : Defining discrete objects for polygonalization : The standard model. In *10th International Conference on Discrete Geometry for Computer Imagery*, volume 2301 de *LNCS*, pages 313–325, 2002.
- [BGCF95] J.W. BULLARD, E.J. GARBOCZI, W.C. CARTER et E.R. FULLER : Numerical methods for computing interfacial mean curvature. *Computational Materials Science*, 4:103–116, 1995.
- [BM94] G. BERTRAND et G. MALANDAIN : A new characterization of three-dimensional simple points. *Pattern Recognition Letters*, 15(2):169–175, 1994.
- [BM03] J. BURGUET et R. MALGOUYRES : Strong thinning and polyhedral approximation of the surface of a voxel object. *Discrete Applied Mathematics*, 125(1):93–114, 2003.
- [Bol07] Dobrina BOLTCHIEVA : *Modélisation géométrique et topologique des images discrètes*. Thèse, Université Louis Pasteur, Strasbourg, octobre 2007. Disponible en ligne : http://lsiit.u-strasbg.fr/Publications/2007/Bol07/these_boltcheva.pdf.
- [Bor96] G. BORGEFORS : On digital distance transforms in three dimensions. *Computer Vision and Image Understanding*, 64(3):368–376, 1996.
- [Bur02] Jasmine BURGUET : *Surfaces discrètes multi-échelle, squelettisation, polyédricisation et opérations ensemblistes sur des polyèdres*. Thèse de doctorat, Université de Caen, décembre 2002.
- [Buz05] L. BUZER : An elementary algorithm for digital line recognition in the general case. In *12th International Conference on Discrete Geometry for Computer Imagery*, volume 3429 de *LNCS*, pages 299–310, Poitiers, France, avril 2005.
- [CFTT03] D. COEURJOLLY, F. FLIN, O. TEYTAUD et L. TOUGNE : Multigrid convergence and surface area estimation. In *Theoretical Foundations of Computer Vision "Geometry, Morphology, and Computational Imaging"*, volume 2616 de *LNCS*, pages 101–119, 2003.
- [CGS04] D. COEURJOLLY, A. GUILLAUME et I. SIVIGNON : Reversible discrete volume polyhedrization using marching cubes. In *SPIE Vision Geometry XII*, volume 5300 de *LNCS*, pages 1–11, San Jose, USA, 2004.
- [CHRU85] L.-S. CHEN, G.T. HERMAN, R.A. REYNOLDS et K.K. UDUPA : Surface shading in the cuberille environment. *IEEE Computer Graphics and Applications*, 5(12):33–43, décembre 1985.
- [Chv83] Vašek CHVÁTAL : *Linear Programming*. Freeman, New York, 1983.
- [CM91] J.-M. CHASSERY et A. MONTANVERT : *Géométrie discrète en analyse d'images*. Traité des Nouvelles Technologies, Hermès, Paris, France, 1991.

- [CMC07] D. COEURJOLLY, A. MONTANVERT et J.-M. CHASSERY : *Géométrie discrète et images numérique*. Traité IC2 – série Signal et Image, Hermès – Lavoisier, Paris, France, 2007.
- [Coe02a] D. COEURJOLLY : Visibility in discrete geometry : An application to discrete geodesic paths. *In 10th International Conference on Discrete Geometry for Computer Imagery*, volume 2301 de *LNCS*, pages 326–337, Bordeaux, France, avril 2002.
- [Coe02b] David COEURJOLLY : *Algorithmique et géométrie discrète pour la caractérisation des courbes et des surfaces*. Thèse, Université Lumière Lyon 2, décembre 2002.
- [CZ06] D. COEURJOLLY et L. ZERARGA : Supercover model, digital straight line recognition and curve reconstruction on the irregular isothetic grids. *Computers & Graphics*, 30(1):46–53, 2006.
- [DA05] M. DEXET et E. ANDRES : Architecture d'un modeleur géométrique à base topologique d'objets discrets. *In Journées de l'Association Française d'Informatique Graphique*, IRCAD, Strasbourg, novembre 2005.
- [DA09] M. DEXET et E. ANDRES : A generalized preimage for the digital analytical hyperplane recognition. *Discrete Applied Mathematics*, 157(3):476–489, 2009.
- [dBvKOS00] M. de BERG, M. van KREVELD, M. OVERMARS et O. SCHWARZKOPF : *Computational Geometry : Algorithms and Applications*. Springer-Verlag, Heidelberg, 2nd édition, 2000.
- [DCA06] M. DEXET, D. COEURJOLLY et E. ANDRES : Invertible polygonalization of 3d planar digital curves and application to volume data reconstruction. *In Second International Symposium on Visual Computing*, volume 4292 de *LNCS*, pages 514–523, Lake Tahoe, NV, USA, novembre 2006.
- [Dex06] Martine DEXET : *Architecture d'un modeleur géométrique à base topologique d'objets discrets et méthodes de reconstruction en dimensions 2 et 3*. Thèse, Université de Poitiers, 2006. Disponible en ligne : <http://www.sic.sp2mi.univ-poitiers.fr/publications/files/publi1824.pdf>.
- [DR95] Isabelle DEBLED-RENNESON : *Étude et reconnaissance des droites et plans discrets*. Thèse, Université Louis Pasteur, Strasbourg, décembre 1995. Disponible en ligne : <http://www.loria.fr/~debled/TheseDebledRenneson.pdf>.
- [DR07] Isabelle DEBLED-RENNESON : *Éléments de géométrie discrète – Vers une étude des structures discrètes bruitées*. Habilitation à diriger des recherches, Université Henri Poincaré, Nancy, décembre 2007. Disponible en ligne : http://www.loria.fr/~debled/HDR_IDR_07.pdf.
- [DRDR06] H. DÖRKSEN-REITER et I. DEBLED-RENNESON : A linear algorithm for polygonal representations of digital sets. *In 11th International Workshop on Combinatorial Image Analysis*, volume 4040 de *LNCS*, pages 307–319, 2006.
- [DRFRD05] I. DEBLED-RENNESON, F. FESCHET et J. ROUYER-DEGLI : Optimal blurred segments decomposition in linear time. *In 12th International Conference on Discrete Geometry for Computer Imagery*, volume 3429 de *LNCS*, pages 371–382, Poitiers, France, avril 2005.
- [DRFRD06] I. DEBLED-RENNESON, F. FESCHET et J. ROUYER-DEGLI : Optimal blurred segments decomposition of noisy shapes in linear time. *Computers & Graphics*, 30(1):30–36, 2006.
- [DRR95] I. DEBLED-RENNESON et J.-P. REVEILLÈS : A linear algorithm for segmentation of digital curves. *International Journal of Pattern Recognition and Artificial Intelligence*, 9(6):635–662, 1995.
- [DRRRD03] I. DEBLED-RENNESON, J.-L. RÉMY et J. ROUYER-DEGLI : Segmentation of discrete curves into fuzzy segments. *In 9th International Workshop on Combinatorial Image Analysis*, volume 12 de *Electronic Notes in Discrete Mathematics*, Palerme, Italie, mai 2003.
- [DS84] L. DORST et A.W.M. SMEULDERS : Discrete representation of straight lines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(4):450–463, 1984.
- [FBF09] A. FAURE, L. BUZER et F. FESCHET : Tangential cover for thick digital curves. *Pattern Recognition*, 42(10):2279–2287, 2009.

- [FF08] A. FAURE et F. FESCHET : Tangential cover for thick digital curves. *In 14th International Conference on Discrete Geometry for Computer Imagery*, volume 4992 de *LNCS*, pages 358–369, Lyon, France, avril 2008.
- [Fou26] J.B.J. FOURIER : Solution d’une question particulière du calcul des inégalités. *Oeuvres II*, pages 317–328, 1826.
- [FP99] J. FRANÇON et L. PAPIER : Polyhedrization of the boundary of a voxel object. *In 8th International Conference on Discrete Geometry for Computer Imagery*, volume 1568 de *Lecture Notes in Computer Science*, pages 425–434, Marne-la-Vallée, France, mars 1999.
- [Fra95] Jean FRANÇON : Discrete combinatorial surfaces. *Graphical Models and Image Processing*, 57(1):20–26, 1995.
- [FST96] J. FRANÇON, J.-M. SCHRAMM et M. TAJINE : Recognizing arithmetic straight lines and planes. *In 6th International Workshop on Discrete Geometry for Computer Imagery*, volume 1176 de *LNCS*, pages 141–150, Lyon, France, novembre 1996.
- [FT99] F. FESCHET et L. TOUGNE : Optimal time computation of the tangent of a discrete curve : application to the curvature. *In 8th International Conference on Discrete Geometry for Computer Imagery*, volume 1568 de *LNCS*, pages 31–40, Marne-la-Vallée, France, mars 1999.
- [FT06] C. FIORIO et J.-L. TOUTANT : Arithmetic discrete hyperspheres and separatingness. *In 13th International Conference on Discrete Geometry for Computer Imagery*, volume 4245 de *LNCS*, pages 425–436, Szeged, Hungary, octobre 2006.
- [GB92] B. GOSTIAUX et M. BERGER : *Géométrie différentielle, variétés, courbes et surfaces*, 2^e édition. PUF, 1992.
- [GDRZ05] Y. GÉRARD, I. DEBLED-RENNESON et P. ZIMMERMANN : An elementary digital plane recognition algorithm. *Discrete Applied Mathematics*, 151:169–183, 2005.
- [Gér03] Y. GÉRARD : A fast and elementary algorithm for digital plane recognition. *In 9th International Workshop on Combinatorial Image Analysis*, volume 12 de *Electronic Notes in Discrete Mathematics*, pages 1–7, Palerme, Italie, mai 2003.
- [GH01] B. GÄRTNER et T. HERRMANN : Computing the width of a point set in 3-space. *J. Exp. Algorithmics*, 4:3, 2001.
- [Gra72] R.L. GRAHAM : An efficient algorithm for determining the convex hull of a finite planar set. *Information Processing Letters*, pages 132–133, 1972.
- [Gé99] Yan GÉRARD : *Contribution à la géométrie discrète*. Thèse, Université d’Auvergne, 1999.
- [Her92] G.T. HERMAN : Discrete multidimensional jordan surfaces. *CVGIP : Graphical Model and Image Processing*, 54(6):507–515, 1992.
- [Her98] G.T. HERMAN : *Geometry of Digital Spaces*. Birkhauser, Boston, 1998.
- [HK03] S. HERMANN et R. KLETTE : Multigrid analysis of curvature estimators. *In Proceedings of Image and Vision Computing New Zealand*, pages 108–112, 2003.
- [HM95] R. HORAUD et O. MONGA : *Vision par ordinateur : outils fondamentaux*, 2^e édition. Éditions Hermès, Paris, 1995. Disponible en ligne : <http://perception.inrialpes.fr/Publications/1995/HM95>.
- [HT88] M.E. HOULE et G.T. TOUSSAINT : Computing the width of a set. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(5):761–765, 1988.
- [Ima] Imagene, Generic digital Image library. Disponible en ligne : <http://gforge.liris.cnrs.fr/projects/imagene>.
- [Kar84] N. KARMARKAR : A new polynomial-time algorithm for linear programming. *Combinatorica*, 4(4):373–395, décembre 1984.
- [KHP94] T. KANUNGO, R. M. HARALICK et I. PHILLIPS : Nonlinear local and global document degradation models. *International Journal of Imaging Systems and Technology*, 5(3):220–230, 1994.

- [KKM90] E. KHALIMSKY, R. KOPPERMAN et P.R. MEYER : Computer graphics and connected topologies on finite ordered sets. *Topology and its Applications*, 36(1):1–17, juillet 1990.
- [KL09] B. KERAUTRET et J.-O. LACHAUD : Multi-scale analysis of discrete contours for unsupervised noise detection. In *13th International Workshop on Combinatorial Image Analysis*, volume 5852 de *LNCS*, pages 187–200, Mexico, novembre 2009.
- [Kov89] V.A. KOVALEVSKY : Finite topology as applied to image analysis. *Computer Vision, Graphics and Image Processing*, 46(2):141–161, 1989.
- [Kov01] V.A. KOVALEVSKY : Algorithms and data structures for computer topology. In *Digital and Image Geometry*, volume 2243 de *LNCS*, pages 38–58, Dagstuhl, Germany, décembre 2001.
- [Kov08] Vladimir A. KOVALEVSKY : *Geometry of Locally Finite Spaces*. Dr. Baerbel Kovalevski, Berlin, 2008.
- [KR04] R. KLETTE et A. ROSENFELD : *Digital Geometry : Geometric Methods For Digital Picture Analysis*. Morgan Kaufmann, Elsevier, 2004.
- [KT96] A.D. KALVIN et R.H. TAYLOR : Surfaces : Polygonal mesh simplification with bounded error. *IEEE Computer Graphics and Applications*, 16(3):64–77, 1996.
- [Lac96] Jacques-Olivier LACHAUD : Topologically defined iso-surfaces. In *6th International Workshop on Discrete Geometry for Computer Imagery*, volume 1176 de *LNCS*, pages 245–256, Lyon, France, novembre 1996.
- [Lac03] J.-O. LACHAUD : Coding cells of digital spaces : a framework to write generic digital topology algorithms. *Electronic Notes in Discrete Mathematics*, 12:337–348, 2003.
- [LC87] W.E. LORENSEN et H.E. CLINE : Marching cubes : a high resolution 3d surface construction algorithm. *Computer Graphics*, 21:163–169, 1987.
- [Len97] A. LENOIR : Fast estimation of mean curvature on the surface of a 3d discrete object. In *7th International Workshop on Discrete Geometry for Computer Imagery*, volume 1347 de *LNCS*, pages 175–186, Montpellier, France, décembre 1997.
- [Len99] Alexandre LENOIR : *Des outils pour les surfaces discrètes : estimation d'invariants géométriques, préservation de la topologie, tracé géodésiques, visualisation*. Thèse, Université de Caen, 1999. Disponible en ligne : <http://www.greyc.ensicaen.fr/EquipeImage/publications/Pub/PostScript/tlenoir.ps.gz>.
- [Lev44] K. LEVENBERG : A method for the solution of certain non-linear problems in least squares. *Quarterly Journal of Applied Mathematics*, II(2):164–168, 1944.
- [Lie91] P. LIENHARDT : Topological models for boundary representation : a comparison with n-dimensional generalized maps. *Computer-Aided Design*, 23(1):59–82, 1991.
- [Lie94] P. LIENHARDT : N-dimensional generalized combinatorial maps and cellular quasi-manifolds. *International Journal on Computational Geometry and Applications*, 4(3):275–324, 1994.
- [LM00] J.-O. LACHAUD et A. MONTANVERT : Continuous analogs of digital boundaries : a topological approach to iso-surfaces. *Graphical Models and Image Processing*, 63:129–164, 2000.
- [LMR96] A. LENOIR, R. MALGOUYRES et M. REVENU : Fast computation of the normal vector field of the surface of a 3-d discrete object. In *6th International Workshop on Discrete Geometry for Computer Imagery*, volume 1176 de *LNCS*, pages 101–112, Lyon, France, novembre 1996.
- [LVdV05] J.-O. LACHAUD, A. VIALARD et F. de VIEILLEVILLE : Analysis and comparative evaluation of discrete tangent estimators. In *12th International Conference on Discrete Geometry for Computer Imagery*, volume 3429 de *LNCS*, pages 240–251, 2005.
- [LVdV07] J.-O. LACHAUD, A. VIALARD et F. de VIEILLEVILLE : Fast, accurate and convergent tangent estimation on digital contours. *Image and Vision Computing*, 25(10):1572–1587, 2007.
- [Mal97] R. MALGOUYRES : A definition of surfaces of \mathbb{Z}^3 : A new 3D discrete Jordan theorem. *Theoretical Computer Science*, 186(1-2):1–41, 1997.

- [Mar63] D.W. MARQUARDT : An algorithm for least-squares estimation of nonlinear parameters. *SIAM Journal on Applied Mathematics*, 11(2):431–441, 1963.
- [MBA93] G. MALANDAIN, G. BERTRAND et N. AYACHE : Topological segmentation of discrete surfaces. *International Journal of Computer Vision*, 10(2):183–197, 1993.
- [McI85] M.D. MCILROY : A note on discrete representation of lines. *AT&T Bell Laboratories Technical Journal*, 64(2):481–490, 1985.
- [Mel87] A. MELKMAN : On-line construction of the convex hull of a simple polygon. *Information Processing Letters*, 25:11–12, 1987.
- [Mot36] T.S. MOTZKIN : *Beitrage zur Theorie der Linearen Ungleichungen*. Thèse de doctorat, Universität Zürich, 1936.
- [MR81] D.G. MORGENTHALER et A. ROSENFELD : Surfaces in three-dimensional digital images. *Information and Control*, 51(3):227–247, 1981.
- [NDR07] T.P. NGUYEN et I. DEBLED-RENNESON : Curvature estimation in noisy curves. In *12th International Conference on Computer Analysis of Images and Pattern*, volume 4673 de *LNCS*, pages 474–481, Vienne, Austria, août 2007.
- [Nie04] G.M. NIELSON : Dual marching cubes. In *15th IEEE Visualization 2004 Conference*, pages 489–496, Austin, TX, USA, octobre 2004.
- [NT08] L. NAJMAN et H. TALBOT : *Morphologie mathématique 1 : approches déterministes*. Traité IC2 – série Signal et Image, Hermès – Lavoisier, Paris, France, 2008.
- [OvL81] M. H. OVERMARS et J. van LEEUWEN : Maintenance of configurations in the plane. *Journal of Computer and System Sciences*, 23(2):166–204, 1981.
- [Pap99] Laurent PAPIER : *Polyédrisation et visualisation d’objets discrets tridimensionnels*. Thèse, Université Louis Pasteur, Strasbourg, 1999.
- [PBDR06] L. PROVOT, L. BUZER et I. DEBLED-RENNESON : Recognition of blurred pieces of discrete planes. In *13th International Conference on Discrete Geometry for Computer Imagery*, volume 4245 de *LNCS*, pages 65–76, Szeged, Hungary, octobre 2006.
- [PDR08a] L. PROVOT et I. DEBLED-RENNESON : Geometric feature estimators for noisy discrete surfaces. In *14th International Conference on Discrete Geometry for Computer Imagery*, volume 4992 de *LNCS*, pages 275–286, Lyon, France, avril 2008.
- [PDR08b] L. PROVOT et I. DEBLED-RENNESON : Segmentation of noisy discrete surfaces. In *12th International Workshop on Combinatorial Image Analysis*, volume 4958, pages 160–171, Buffalo, NY, USA, avril 2008.
- [PDR09] L. PROVOT et I. DEBLED-RENNESON : 3d noisy discrete objects : Segmentation and application to smoothing. *Pattern Recognition*, 42(8):1626–1636, 2009.
- [Per94] Laurent PERROTON : *Segmentation parallèle d’images volumiques*. Thèse, École Normale Supérieure de Lyon, 1994. Disponible en ligne : <ftp://ftp.ens-lyon.fr/pub/LIP/Rapports/PhD/PhD1994/PhD1994-09.ps.Z>.
- [PF98] L. PAPIER et J. FRANÇON : Évaluation de la normale au bord d’un objet discret 3d. *Revue internationale de CFAO et d’Infographie*, 13(2), pages 205–226, 1998.
- [Pro05] Laurent PROVOT : Reconnaissance de morceaux de plans discrets bruités. Mémoire de D.E.A., Ecole Doctorale IAEM Lorraine – Université Paul Verlaine, Metz, juin 2005. Disponible en ligne : http://www.loria.fr/~provot/doc/memoire_dea.pdf.
- [Rev91] J.P. REVEILLÈS : *Géométrie discrète, calculs en nombre entiers et algorithmique*. Thèse d’état, Université Louis Pasteur, Strasbourg, 1991.
- [SdBS00] G. Sanniti di BAJA et S. SVENSSON : Editing 3d binary images using distance transforms. *International Conference on Pattern Recognition*, 2:6030, 2000.
- [SDC04a] I. SIVIGNON, F. DUPONT et J.-M. CHASSERY : Decomposition of a 3d discrete object surface into discrete plane pieces. *Algorithmica*, 38(1):25–43, 2004.

- [SDC04b] I. SIVIGNON, F. DUPONT et J.-M. CHASSERY : Discrete surface segmentation into discrete planes. *In 10th International Workshop on Combinatorial Image Analysis*, volume 3322 de *LNCS*, pages 458–473, Auckland, New Zealand, décembre 2004.
- [SDC05] I. SIVIGNON, F. DUPONT et J.-M. CHASSERY : Reversible polygonalization of a 3d planar discrete curve : Application on discrete surfaces. *In 12th International Conference on Discrete Geometry for Computer Imagery*, volume 3429 de *LNCS*, pages 347–358, Poitiers, France, avril 2005.
- [Ser82] Jean SERRA : *Image Analysis and Mathematical Morphology*, volume 1. Academic Press, Londres, 1982.
- [Siv04] Isabelle SIVIGNON : *De la caractérisation des primitives à la reconstruction polyédrique de surfaces en géométrie discrète*. Thèse, Institut National Polytechnique de Grenoble, 2004. Disponible en ligne : http://www.lis.inpg.fr/stages_dea_theses/theses/manuscript/sivignon.pdf.
- [Spi99] Michael SPIVAK : *A Comprehensive Introduction to Differential Geometry*, 3rd edition, volume 1-5. Publish or Perish, janvier 1999.
- [TDR99] P. TELLIER et I. DEBLED-RENNESON : 3d discrete normal vectors. *In 8th International Conference on Discrete Geometry for Computer Imagery*, volume 1568 de *LNCS*, pages 447–457, Marne-la-Vallée, France, mars 1999.
- [Thü01] G. THÜRMER : Smoothing normal vectors on discrete surfaces while preserving slope discontinuities. *Computer Graphics Forum*, 20(2):103–114, 2001.
- [TM92] E. THIEL et A. MONTANVERT : Chamfer masks : discrete distance functions, geometrical properties and optimization. *In 11th ICPR*, volume 3, pages 244–247, The Hague, The Netherlands, 1992.
- [Tou83] G.T. TOUSSAINT : Solving geometric problems with the rotating calipers. *In Proceedings of IEEE MELECON'83*, Athens, Greece, mai 1983.
- [TW97] G. THÜRMER et C. A. WÜTHRICH : Normal computation for discrete surfaces in 3d space. *Computer Graphics Forum*, 16(3):15–26, 1997.
- [TW98] G. THÜRMER et C.A. WÜTHRICH : Varying neighborhood parameters for the computation of normals on surfaces in discrete space. *In Computer Graphics International*, pages 616–625, Hannover, Germany, juin 1998. IEEE Computer Society.
- [Udu94] J. K. UDUPA : Multidimensional digital boundaries. *CVGIP : Graphical Model and Image Processing*, 56(4):311–323, 1994.
- [USH82] J.K. UDUPA, S.N. SRIHARI et G.T. HERMAN : Boundary detection in multidimensions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 4(1):41–50, 1982.
- [VC00] J. VITTONÉ et J.M. CHASSERY : Recognition of digital naive planes and polyhedrization. *In 9th International Conference on Discrete Geometry for Computer Imagery*, volume 1953 de *Lecture Notes in Computer Science*, pages 296–307, Uppsala, Sweden, décembre 2000.
- [Vit99] Joëlle VITTONÉ : *Caractérisation et reconnaissance de droites et de plans en géométrie discrète*. Thèse, Université Joseph Fourier, Grenoble, 1999. Disponible en ligne : http://www.lis.inpg.fr/fileadmin/stages_dea_theses/theses/manuscript/vittone.pdf.
- [VVD89] B.J.H. VERWER, P.W. VERBEEK et S.T. DEKKER : An efficient uniform cost algorithm applied to distance transforms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(4):425–429, 1989.
- [Wri97] Stephen J. WRIGHT : *Primal-Dual interior-moint methods*. SIAM, 1997.
- [YCK92] R. YAGEL, D. COHEN et A. E. KAUFMAN : Normal estimation in 3d discrete space. *The Visual Computer*, 8(5&6):278–291, 1992.

Résumé

Les travaux de cette thèse s'inscrivent dans le cadre de la géométrie discrète, une discipline ayant pour objectif de définir un cadre théorique pour transposer dans \mathbb{Z}^n les bases de la géométrie euclidienne – les notions discrètes définies étant le plus proche possible des notions continues que nous connaissons (telles que distance, droite, convexité, ...).

De nombreuses études ont déjà été menées au sein de cette discipline, pour en définir l'espace de travail ainsi que les objets fondamentaux manipulés et en saisir leurs propriétés. Des algorithmes de reconnaissance pour ces primitives discrètes ont été développés et utilisés dans des problèmes comme la reconnaissance de formes, l'extraction de caractéristiques géométriques et bien d'autres encore. Néanmoins, la majorité des études ont été effectuées en se reposant sur la régularité des structures fondamentales de l'espace discret, souvent issues de définitions arithmétiques, et ces critères de régularité sont généralement essentiels aux différents algorithmes développés. Or, en pratique, les objets manipulés sont très souvent bruités par les méthodes d'acquisition (scanners, IRM, ...) qui suppriment ce caractère régulier des objets.

Dans cette thèse, nous nous intéressons aux objets discrets 3D et proposons une nouvelle primitive discrète, le morceau flou de plan discret, destinée à apporter plus de flexibilité dans les traitements, afin de proposer des algorithmes capables de fournir des résultats satisfaisants aussi bien sur des objets réguliers que non réguliers. Avec l'emploi de cette nouvelle primitive discrète, nous définissons différents estimateurs de caractéristiques géométriques au bord d'objets discrets et montrons comment les utiliser dans des problèmes de segmentation et de polyédrisation d'objets discrets possiblement bruités.

Mots-clés: Géométrie discrète, objets discrets bruités, morceaux flous de plans discrets, algorithme de reconnaissance, segmentation, polyédrisation

Abstract

The context of this thesis is the digital geometry. An aim of this field is to develop a theoretical framework to implement in \mathbb{Z}^n the basis of the Euclidean geometry. The discrete concepts are thus defined as close as possible to their continuous counterparts (such as distance, straight line, convexity, ...).

Many studies have already been led within this research area to define the workspace, the fundamental objects and figure their properties out. Recognition algorithms have been designed for these discrete primitives and used in problems such as pattern recognition, geometric features estimation and many more. However, most of these studies rely on the regularity of fundamental structures of the digital space, that usually comes from their arithmetical definitions. But in practice, objects may be corrupted by the acquisition devices (like scanners or MRI) which eliminate this regularity.

This thesis is focused on 3D digital objects. We introduce a new discrete primitive: the blurred piece of a discrete plane. Its aim is to bring some flexibility in treatments in order to design algorithms that yield satisfying results on both noisy and non-noisy digital objects. Relying on this new primitive, we define different geometric feature estimators on the border of such objects and show how to use them to guide the segmentation and polyhedrization processes of this border.

Keywords: Digital geometry, noisy discrete objects, blurred pieces of discrete planes, recognition algorithms, segmentation, polyhedrization