HAL open science

# Reconnaissance de morceaux de plans discrets bruités

Laurent Provot

**To cite this version:**

Laurent Provot. Reconnaissance de morceaux de plans discrets bruités. 2005. hal-00013869

**HAL Id: hal-00013869**

**https://hal.science/hal-00013869v1**

Preprint submitted on 15 Nov 2005

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

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

Université Paul Verlaine – Metz

# Reconnaissance de morceaux de plans discrets bruités

## MÉMOIRE

soutenu le 28 juin 2005

pour l'obtention du

**DEA Informatique de Lorraine - École Doctorale IAEM Lorraine**

par

Laurent Provot

**Composition du jury**

| | |
|---|---|
| | Noëlle Carbonell |
| | Dominique Méry |
| | Didier Galmiche |
| *Responsable de filière :* | Dieter Kratsch |
| *Encadrant :* | Isabelle Debled-Rennesson |

**Laboratoire Lorrain de Recherche en Informatique et ses Applications — UMR 7503**

# Remerciements

Je tenais à remercier toutes les personnes qui ont, directement ou indirectement, contribué au bon déroulement de mon stage de DEA au sein du Loria et en particulier toutes les personnes de l'équipe ADAGE.

Ma plus grande gratitude va à Isabelle Debled-Rennesson pour m'avoir fait découvrir la géométrie discrète et m'avoir proposé un sujet aussi passionnant que l'étude des plans discrets, pour son aide, sa disponibilité et sa bonne humeur tout au long de ce stage et pour ses nombreux conseils et encouragements.

Je remercie Grégory Kucherov pour m'avoir accueilli au sein de l'équipe ADAGE.

Merci à Laurent, Fabrice et Christophe pour la bonne ambiance dans notre bureau et pour avoir pris le temps de répondre à toutes mes questions... et pour les bonbons à la menthe.

Je voudrais aussi remercier Jean-Luc Rémy et Mathilde pour leur relecture durant la rédaction de ce mémoire.

Enfin, merci à Monique Teillaud et Sylvain Pion pour avoir répondu à mes questions concernant l'utilisation de CGAL.

# Résumé

La reconnaissance d'objets discrets est un sujet important en géométrie discrète et de nombreux travaux concernant les droites et les plans ont été réalisés. Nous nous intéressons à la notion d'objets discrets « flous », correspondant à des objets discrets bruités, et à leur détection. Ces objets sont définis analytiquement et les algorithmes développés à partir de ceux-ci s'adaptent aux données bruitées par les méthodes d'acquisition, par exemple les données médicales issues de scanners, d'échographie, d'IRM...

Dans ce mémoire nous nous intéresserons plus particulièrement aux plans discrets bruités. En introduisant la définition d'un morceau de plan discret flou, nous montrerons que le problème de le reconnaissance d'un tel objet est équivalent à celui du calcul de l'épaisseur d'un ensemble de points en dimension 3. Après une étude des méthodes existantes résolvant le calcul de l'épaisseur, un algorithme connu sera alors adapté, puis programmé pour résoudre le problème de la reconnaissance incrémentale des morceaux de plans discrets flous.



**Mots-clés:** géométrie discrète, morceaux de plans discrets flous, calcul de l'épaisseur

# Table des matières

**Introduction générale** **1**

**1 Notions de base en géométrie discrète** **3**

1.1 Voisinage et connexité . . . . . 3

1.2 Droites et plans discrets . . . . . 4

1.2.1 Droites discrètes . . . . . 4

1.2.2 Plans discrets . . . . . 5

1.3 Reconnaissance des droites et plans discrets . . . . . 7

**2 Les objets discrets flous** **9**

2.1 Segments flous . . . . . 9

2.2 Une nouvelle définition : les morceaux de plans discrets flous . . . . . 11

2.3 Générateur de morceaux de plans discrets flous . . . . . 12

**3 Calcul de l'épaisseur d'un ensemble de points** **15**

3.1 Méthode de Houle et Toussaint . . . . . 16

3.2 Méthode de Gärtner et Herrmann . . . . . 17

3.3 Comparaison des deux méthodes . . . . . 18

3.4 Une autre approche . . . . . 18

**4 Étude expérimentale** **20**

4.1 Approche initiale . . . . . 21

4.1.1 Parcours choisi . . . . . 21

4.1.2 Premier algorithme de reconnaissance . . . . . 21

4.2 Premières observations et nouvelle version . . . . . 22

4.2.1 Calcul incrémental de l'enveloppe convexe . . . . . 23

4.2.2 Algorithme résultant . . . . . 24

4.3 Tentative d'amélioration de l'algorithme incrémental . . . . . 25

4.4 Algorithme incrémental retenu . . . . . 27

**Conclusion et perspectives** **30**

**Bibliographie** **31**

**Annexes** **33**

**A Liste des tricubes du premier 48$^{e}$ d'espace** **33**

**B Interface de visualisation 3D** **36**

# Table des figures


1 Intersection de droites discrètes . . . . . 1

1.1 Les trois types de connexité définies dans $\mathbb{Z}^3$ . . . . . 4
1.2 Droites d'appui . . . . . 5
1.3 Plans discrets . . . . . 6
1.4 Points d'appui d'un plan discret naïf . . . . . 7
1.5 Représentation par restes du plan $\mathcal{P}(6, 13, 27, 0, 27)$ . . . . . 7

2.1 Distance verticale et droites support . . . . . 10
2.2 Cas de figure possibles lors de l'ajout d'un point au segment flou . . . . . 10
2.3 Reconnaissance incrémentale d'un segment flou . . . . . 11
2.4 Quelques morceaux de plans discrets flous générés à partir de l'algorithme 1 . . . . . 14

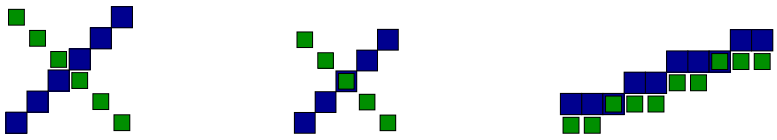
3.1 Paires antipodales en dimension 3 . . . . . 16
3.2 La transformation géométrique de Brown . . . . . 17

4.1 Parcours pseudo-géodésique . . . . . 21
4.2 Enveloppe convexe incrémentale . . . . . 23
4.3 Le point ajouté devient point d'appui . . . . . 25
4.4 Contre-exemple de l'observation 3 . . . . . 26

# Introduction générale

Avec le développement de l'imagerie informatique et son utilisation dans des contextes de plus en plus diversifiés, mais notamment dans le milieu médical avec par exemple, les scanners, l'IRM ou la tomographie, la géométrie discrète a connu un regain d'intérêt important ces dernières années. Cette discipline de l'informatique, introduite il y a plus de quarante ans par les recherches de scientifiques bien connus comme A. Rosenfeld, G. Herman ou encore E. Khalimsky, vise à étudier des objets discrets, c'est-à-dire composés d'un ensemble dénombrable de points, par opposition aux objets de la géométrie euclidienne qui, eux, sont généralement composés d'un ensemble indénombrable de points. Elle joue un rôle essentiel dans des domaines comme l'infographie, la reconnaissance de forme, la vision par ordinateur ou encore l'analyse de documents, tout simplement parce que les données dans l'ordinateur sont intrinsèquement discrètes.

Les objets discrets ont souvent peu de propriétés communes avec leurs homologues continus. Un exemple très révélateur est que dans le plan discret $\mathbb{Z}^2$ deux droites discrètes non parallèles peuvent n'avoir aucun point d'intersection (cf. figure 1(a)). En effet les résultats les plus élémentaires de la géométrie euclidienne ne sont pas vérifiés dans les espaces discrets : des notions fondamentales telles que la continuité sont un peu bousculées – que veut dire continu dans un espace où tout est a priori discontinu ? Une géométrie discrète théorique s'avère donc nécessaire, afin de développer des théorèmes et des méthodes spécifiques à ce type d'espaces et d'objets, et tenter de transposer dans ce contexte les notions familières de la géométrie euclidienne.



(a) (b) (c)

Figure 1 – Différents type d'intersection entre les droites discrètes : (a) l'intersection de deux droites sécantes est vide ; (b) l'intersection de deux droites sécantes est composée d'un point unique ; (c) l'intersection de ces deux droites parallèles est non vide.

Un problème fondamental en géométrie discrète, qui trouve de nombreuses applications dans la reconnaissance de forme ou encore l'analyse d'image, est celui de la reconnaissance d'objets discrets. L'introduction de l'arithmétique discrète avec la définition des droites [Rev91] et des plans [And93] discrets au début des années 90 a offert de nouveaux moyens de développement pour une étude entièrement discrète des données. Ainsi, en s'appuyant sur les définitions précédentes, de nombreux chercheurs se sont intéressés au problème de reconnaissance des segments de droites discrètes, aussi bien en dimension 2 qu'en dimension 3, ainsi qu'à la reconnaissance des morceaux de plans discrets, et des algorithmes efficaces ont été mis au point [DRR96, VC00]. Ceux-ci sont d'ailleurs la base de travaux plus récents sur la polyédrisation d'objets discrets, où l'on cherche à passer d'une représentation d'un objet discret par un ensemble de points en une représentation par un ensemble de sommets, d'arêtes et de faces [Pap99, Siv04].

Néanmoins, toutes ces études ont été effectuées sur des objets exacts, issus des définitions arithmé-

tiques des droites et des plans discrets, et présentant de ce fait des critères de régularité essentiels aux différents algorithmes developpés. Or, en pratique, les objets manipulés sont très souvent bruités par les méthodes d'acquisition (scanners, IRM...) qui suppriment le caractère régulier des objets et donc les algorithmes développés ci-dessus ne fonctionnent plus, ou alors produisent des résultats décevants par rapport à ce que nous pourrions espérer. Il semble donc intéressant de poser de nouvelles définitions pour ces objets non réguliers, destinées à prendre en compte le bruit, et ainsi proposer de nouveaux algorithmes efficaces pour travailler avec ce type d'objets. C'est dans ce cadre que se place mon stage de DEA.

Même si l'intérêt de tels objets semble évident, leur étude selon l'approche arithmétique n'a commencé que très récemment. En 2003, I. Debled-Rennesson et al. introduisent la notion d'objets discrets *flous*, qui correspond à l'étude d'objets discrets bruités [DRRRD03]. Leurs travaux portent sur les segments de droites discrètes flous et leur reconnaissance. Dans [DRFRD05] ils démontrent l'équivalence entre le problème de reconnaissance d'un segment flou et celui du calcul de la distance verticale de l'enveloppe convexe des points de ce segment. En s'appuyant sur les caractéristiques de l'enveloppe convexe, ils developpent un algorithme incrémental linéaire de reconnaissance de segments flous. L'idée est d'ajouter les points les uns après les autres en suivant une certaine direction, de mettre à jour l'enveloppe convexe de ces points à chaque étape et de recalculer la distance verticale de cette dernière. Nous en déduisons alors les nouvelles caractéristiques du segment de droite reconnu. Cette méthode a été utilisée dans le cadre de la segmentation de courbes discrètes bruitées. Si l'épaisseur du segment en cours de reconnaissance dépasse une limite fixée auparavant permettant de contrôler le bruit maximum autorisé, alors la reconnaissance s'arrête en gardant le segment précédant et une nouvelle reconnaissance est lancée à partir du point courant.

Les résultats sont encourageants et cette nouvelle approche plus géométrique du problème de reconnaissance d'objets discrets ouvre de nouvelles portes pour l'étude des objets discrets flous. En effet, aucune recherche n'a encore été effectuée au niveau des plans discrets non réguliers.

Le but de mon stage est donc d'essayer de généraliser la méthode précédente aux morceaux de plans discrets non réguliers afin de proposer un algorithme de reconnaissance incrémental. Le côté incrémental est important pour des travaux ultérieurs dans le cadre de la polyédrisation d'objets discrets bruités. Nous pourrons ainsi envisager d'utiliser une approche identique à celle utilisée pour la segmentation de courbes discrètes bruitées, mais adaptée aux plans.

Dans la première partie de ce mémoire j'introduirai quelques notions de base en géométrie discrète. Puis, dans la seconde partie, je rappellerai la définition d'un segment flou introduite dans [DRFRD05] et j'expliquerai plus en détail l'approche utilisée pour leur reconnaissance. Je généraliserai le concept en introduisant une nouvelle définition : celle d'un *morceau de plan flou*, et je proposerai un algorithme permettant de générer de tels objets. Je ferai aussi le lien entre la reconnaissance d'un morceau de plan flou et le calcul de l'épaisseur d'un ensemble de points en dimension 3. Ceci m'amènera à présenter, dans la troisième partie du mémoire, plusieurs méthodes connues pour résoudre ce dernier problème. Dans la partie suivante j'expliquerai ma démarche pour les adapter au problème de reconnaissance de morceaux de plans discrets flous, et je donnerai un algorithme de reconnaissance incrémental. Enfin je terminerai en proposant quelques perspectives de travaux possibles.

# Chapitre 1

# Notions de base en géométrie discrète

## Sommaire


**1.1 Voisinage et connexité** . . . . . **3**
**1.2 Droites et plans discrets** . . . . . **4**
1.2.1 Droites discrètes . . . . . 4
1.2.2 Plans discrets . . . . . 5
**1.3 Reconnaissance des droites et plans discrets** . . . . . **7**


## Introduction

Je vais présenter dans cette partie des notions de base utiles à la compréhension du reste du mémoire. Pour des compléments d'information on peut se référer aux documents suivants [Rev91, Coe02, Siv04].

En géométrie discrète, tous les points ont des coordonnées entières, nous travaillons donc dans $\mathbb{Z}^n$ en général. Cependant pour le problème qui nous intéresse nous nous limitons à $\mathbb{Z}^3$. Nous travaillons alors dans un espace discret 3D, défini par un maillage cubique unitaire centré sur les points de coordonnées entières. Un cube unitaire de ce maillage est appelé *voxel* [1].

De plus cette espace discret peut être divisé en 48 parties. En effet, chacune des trois coordonnées $x$, $y$, $z$ peut être soit positive soit négative. Nous pouvons donc partitionner l'espace en $2^3 = 8$. Une fois les signes fixés, les 3! permutations des coordonnées nous permettent de les ordonner de 6 manières différentes, ce qui nous donne 6 façons de découper les 8 parties. Ce qui nous fait un total de $8 \times 6 = 48$ parties. On appellera *premier 48$^e$ d'espace* la partie de l'espace définie par $0 \leqslant x \leqslant y \leqslant z$, $z > 0$. Ainsi, sans perdre en généralité, nous pourrons limiter notre étude aux plans dont la normale est contenue dans le premier 48$^e$ d'espace.

Afin de pouvoir qualifier les objets avec lesquels nous allons travailler, il est intéressant d'introduire la notion de connexité entre les voxels. Cette notion repose directement sur le type de voisinage entre les voxels.

## 1.1 Voisinage et connexité

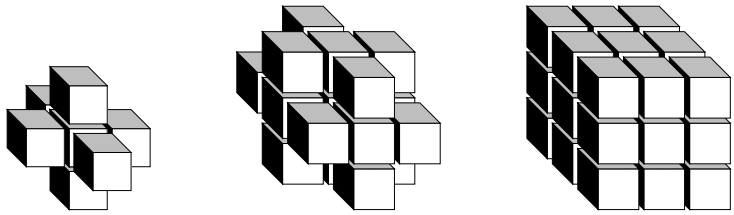
Dans $\mathbb{Z}^3$ on définit généralement trois types de voisinage pour les points (voxels).

**Définition 1.1.1.** Soient $P_1(x_1, y_1, z_1)$ et $P_2(x_2, y_2, z_2)$ deux points appartenant à $\mathbb{Z}^3$ tels que $|x_1 - x_2| \leqslant 1$, $|y_1 - y_2| \leqslant 1$ et $|z_1 - z_2| \leqslant 1$. Soit $d = |x_1 - x_2| + |y_1 - y_2| + |z_1 - z_2|$.

[1] abréviation de ***vo***lume pi***xel***

– si $d = 1$, alors $P_1$ et $P_2$ sont dits *6-voisins* ou encore *6-connexes*,
– si $d \leqslant 2$, alors $P_1$ et $P_2$ sont dits *18-voisins* ou encore *18-connexes*,
– si $d \leqslant 3$, alors $P_1$ et $P_2$ sont dits *26-voisins* ou encore *26-connexes*.

De façon plus concrète,
– si deux voxels sont incidents par face, alors ils sont *6-connexes*,
– si deux voxels sont incidents par arête, alors ils sont *18-connexes*,
– si deux voxels sont incidents par sommet, alors ils sont *26-connexes*.
Les différents voisinages, par rapport au voxel central, sont illustrés à la figure 1.1.



(a) 6-connexité (b) 18-connexité (c) 26-connexité

Figure 1.1 – Les trois types de connexité définies dans $\mathbb{Z}^3$

**Définition 1.1.2.** Soit $(P_i)_{1 \leqslant i \leqslant n}$ une suite de points de $\mathbb{Z}^3$. Si pour tout $i$ tel que $1 \leqslant i < n$, $P_i$ et $P_{i+1}$ sont 6-connexes (resp. 18-connexes, 26-connexes), alors $(P_1, \ldots, P_n)$ est appelé un *6-chemin* (resp. *18-chemin*, *26-chemin*).

**Définition 1.1.3.** Soit $E$ un ensemble de points de $\mathbb{Z}^3$. On dit que $E$ est *6-connexe* (resp. *18-connexe*, *26-connexe*), si pour tout points $P$ et $Q$ appartenant à E, il existe un 6-chemin (resp. 18-chemin, 26-chemin) inclus dans $E$ allant de $P$ à $Q$.

Maintenant que nous avons présenté notre espace de travail, nous pouvons définir les objets utiles à notre étude.

## 1.2 Droites et plans discrets

On peut voir les droites et les plans discrets comme les objets de base de la géométrie discrète. La définition arithmétique de la droite discrète a été introduite en 1991 par J.-P. Reveillès [Rev91] et a été généralisée aux plans discrets par E. Andrès [And93].

### 1.2.1 Droites discrètes

**Définition 1.2.1.** Soient $a$, $b$, $\mu$ et $\omega$ quatre entiers tels que $\text{pgcd}(a, b) = 1$ et $\omega > 0$. On appelle *droite discrète* de pente $\frac{a}{b}$, $b \neq 0$, de borne inférieure $\mu$ et d'épaisseur arithmétique $\omega$, l'ensemble des points $P(x, y) \in \mathbb{Z}^2$ satisfaisant la double inéquation diophantienne

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

On note alors $\mathcal{D}(a, b, \mu, \omega)$, ou tout simplement $\mathcal{D}$ s'il n'y a pas d'ambiguïté possible.
– si $\omega = \max(|a|, |b|)$, alors la droite $\mathcal{D}$ est appelé *droite discrète naïve*.
– si $\omega = |a| + |b|$, alors la droite $\mathcal{D}$ est appelé *droite discrète standard*.

Une droite discrète naïve est représentée à la figure 1.2.

**Définition 1.2.2.** Soit $\mathcal{D}(a, b, \mu, \omega)$ une droite discrète. Les droites réelles d'équations $ax - by = \mu$ et $ax - by = \mu + \omega - 1$ sont appelés les *droites d'appui de la droite* $\mathcal{D}$. Un point entier de ces droites sera nommé *point d'appui*.

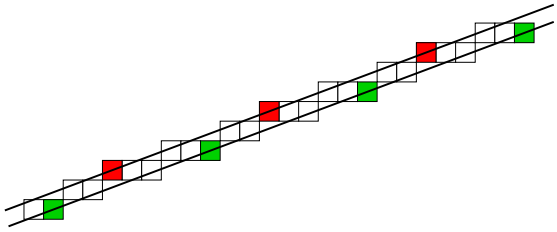


Figure 1.2 – Segment de la droite discrète naïve $\mathcal{D}(3, 8, -4, 8)$. Les droites d'appui sont les deux droites réelles et les points d'appui sont représentés en couleur.

### 1.2.2 Plans discrets

On peut généraliser cette définition aux plans discrets :

**Définition 1.2.3.** Soient $a$, $b$, $c$, $\mu$ et $\omega$ cinq entiers tels que $\text{pgcd}(a, b, c) = 1$ et $\omega > 0$. On appelle *plan discret* de vecteur normal $(a, b, c)$, de borne inférieure $\mu$ et d'épaisseur arithmétique $\omega$, l'ensemble des points $P(x, y, z) \in \mathbb{Z}^3$ satisfaisant la double inéquation diophantienne

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

On note alors $\mathcal{P}(a, b, c, \mu, \omega)$, ou tout simplement $\mathcal{P}$ s'il n'y a pas d'ambiguïté possible.

Des exemples de morceaux de plans discrets sont représentés à la figure 1.3.

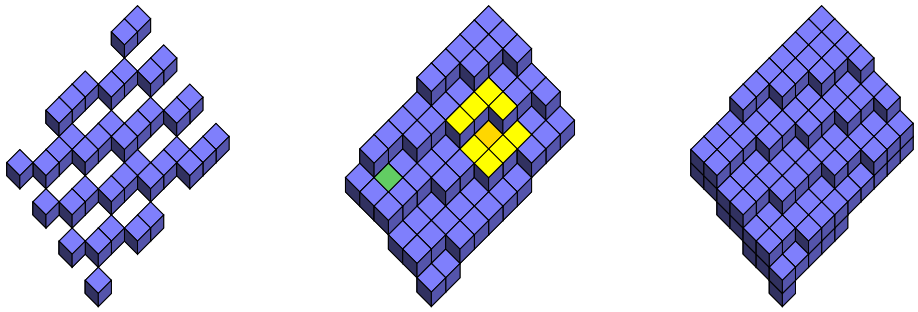
**Définition 1.2.4.** Soient $P_1(x_1, y_1, z_1)$ et $P_2(x_2, y_2, z_2)$ deux points 6-connexes (resp. 18-connexes, 26-connexes) de $\mathbb{Z}^3$. Soit $\mathcal{P}(a, b, c, \mu, \omega)$ un plan discret. On dit que $(P_1, P_2)$ est un 6-trou (resp. 18-trou, 26-trou) du plan $\mathcal{P}$ si $ax_1 + by_1 + cz_1 < \mu$ et $\mu + \omega \leqslant ax_2 + by_2 + cz_2$.

De manière informelle, le plan $\mathcal{P}$ a un trou s'il existe deux voxels voisins dont l'un est « au-dessus » de $\mathcal{P}$ et l'autre « en-dessous » de $\mathcal{P}$.

Dans [And93], E. Andrès a mené une étude sur la connexité des plans discrets en fonction de leur épaisseur arithmétique $\omega$. Les relations suivantes sont mises en évidence :
- si $\omega < \max(|a|, |b|, |c|)$, alors le plan $\mathcal{P}$ est déconnecté ou bien 26-connexe ou 18-connexe avec des 6-trous,
- si $\omega = \max(|a|, |b|, |c|)$, alors le plan $\mathcal{P}$ est appelé *plan discret naïf*. Les plans discrets naïfs sont des plans 18-connexes sans 6-trous, d'épaisseur minimale.
- si $\omega = |a| + |b| + |c|$, alors le plan $\mathcal{P}$ est appelé *plan discret standard*. Les plans discrets standards sont des plans 6-connexes sans 26-trous, d'épaisseur minimale.
- si $\omega > |a| + |b| + |c|$, alors le plan $\mathcal{P}$ est appelé *plan discret épais*.

Dans cette classification, le plan discret naïf est d'un intérêt particulier et c'est celui qui a été le plus étudié. On peut en effet le voir comme le plan minimal permettant de représenter la discrétisation d'un plan réel quelconque. De plus, si on se place dans le premier 48$^{\text{e}}$ d'espace, il a la particularité d'être fonctionnel en $(x, y)$. En effet, si l'on considère la fonction $f_{(a,b,c,\mu)}(x, y) = -\left\lfloor \frac{ax+by-\mu}{c} \right\rfloor$, alors le plan $\mathcal{P}(a, b, c, \mu, c)$ est défini comme étant l'ensemble

$$\{(x, y, z) \in \mathbb{Z}^3 \mid \mu \leqslant ax + by + cz < \mu + c\} \Longleftrightarrow \left\{\left(x, y, z = f_{(a,b,c,\mu)}(x, y)\right) \text{ avec } (x, y) \in \mathbb{Z}^2\right\}$$



(a) (b) (c)

Figure 1.3 – Représentation d'un morceau du plan $\mathcal{P}(6, 13, 27, 0, \omega)$ pour différentes épaisseurs : (a) $\omega = 15$ un plan avec des trous, (b) $\omega = 27$ un plan naïf, (c) $\omega = 46$ un plan standard.

Pour tout $(x, y)$ il existe donc un unique $z$ tel que le voxel $(x, y, z) \in \mathcal{P}$.

Une autre manière de voir le plan discret naïf est non pas comme un ensemble de voxels, mais plutôt comme un ensemble de configurations locales appelées *tricubes*.

**Définition 1.2.5.** Dans un plan discret naïf $\mathcal{P}$, un point $M$ possède exactement 8 voisins appartenant à ce même plan. L'ensemble de ces 9 voxels est appelé *tricube de $M$ dans $\mathcal{P}$*.

*Remarque.* On peut observer un tricube dans le plan naïf de la figure 1.3(b) en jaune.

Dans un plan discret naïf on trouve au maximum 9 configurations de tricubes différentes. Dans [DR95], un dénombrement des tricubes du premier 48$^{\text{e}}$ d'espace a été effectué. On trouve ainsi 40 configurations différentes de tricubes dont la liste est donnée à l'annexe A. Dans sa thèse [Vit99], J. Vittone a utilisé ces tricubes pour développer un algorithme de reconnaissance de plans discrets naïfs. Nous utiliserons ces tricubes pour générer des morceaux de plans flous dans la partie 2.3.

Une autre notion importante est celle du *reste* d'un point en fonction des caractéristiques d'un plan discret.

**Définition 1.2.6.** On appelle *fonction reste par rapport au plan* $\mathcal{P}(a, b, c, \mu, \omega)$ la fonction $r_{(a,b,c,\mu)}$ définie par :

$$r_{(a,b,c,\mu)}(x, y, z) = ax + by + cz - \mu$$

*Remarque.* Il découle de cette définition que seuls les voxels $(x, y, z)$ tels que $0 \leqslant r_{(a,b,c,\mu)}(x, y, z) < \omega$ appartiennent au plan $\mathcal{P}(a, b, c, \mu, \omega)$.

De plus les valeurs 0 et $\omega - 1$ de la fonction $r$ sont particulières puisqu'elles représentent les valeurs limites pour un voxel du plan $\mathcal{P}$. Cela permet d'introduire la notion de *plan d'appui*, dont l'importance pour la reconnaissance des plans discrets naïfs à été mise en évidence dans [DR95].

**Définition 1.2.7.** Soit $\mathcal{P}(a, b, c, \mu, \omega)$ un plan discret. Les plans réels d'équations $ax + by + cz = \mu$ et $ax + by + cz = \mu + \omega - 1$ sont appelés les *plans d'appui du plan $\mathcal{P}$*. Un point appartenant à la fois au plan $\mathcal{P}$ et à un de ses plans d'appui est appelé un *point d'appui*.

*Remarque.* Si on se place dans le premier 48$^{\text{e}}$ d'espace, alors on appellera le plan $ax + by + cz = \mu$ (resp. $ax + by + cz = \mu + \omega - 1$) le *plan d'appui inférieur*(resp. *plan d'appui supérieur*) de $\mathcal{P}$.

*Remarque.* Les points d'appui inférieurs (resp. supérieurs) du plan $\mathcal{P}(a, b, c, \mu, \omega)$ sont les points dont le reste vaut 0 (resp. $\omega - 1$). Ces notions sont illustrés à la figure 1.4.

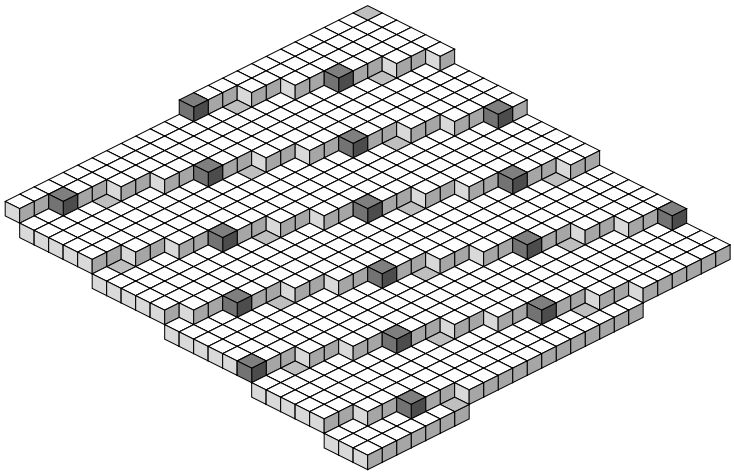


Figure 1.4 – Points d'appui de $\mathcal{P}(3, 7, 37, 0, 37)$ ; en gris foncé les points d'appui supérieurs et en gris clair les points d'appui inférieurs.

Nous avons vu qu'un plan discret naïf $\mathcal{P}$ du premier 48$^{\text{e}}$ d'espace était fonctionnel en $(x, y)$. Ceci veut dire que chaque voxel de $\mathcal{P}$ a une unique projection dans le plan $Oxy$. Ceci nous permet de représenter $\mathcal{P}$ dans le plan $Oxy$ sans perte d'information. En pondérant chaque pixel $(x, y)$ de cette représentation par le reste par rapport à $\mathcal{P}$ du voxel $(x, y, f(x, y))$, nous obtenons ce qu'on appelle la *représentation par restes* du plan $\mathcal{P}$. Cette représentation est illustrée à la figure 1.5.

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 24 | 3 | 9 | 15 | 21 | 0 | 6 | 12 | 18 | 24 |
| 11 | 17 | 23 | 2 | 8 | 14 | 20 | 26 | 5 | 11 |
| 25 | 4 | 10 | 16 | 22 | 1 | 7 | 13 | 19 | 25 |
| 12 | 18 | 24 | 3 | 9 | 15 | 21 | 0 | 6 | 12 |
| 26 | 5 | 11 | 17 | 23 | 2 | 8 | 14 | 20 | 26 |
| 13 | 19 | 25 | 4 | 10 | 16 | 22 | 1 | 7 | 13 |
| 0 | 6 | 12 | 18 | 24 | 3 | 9 | 15 | 21 | 0 |

y
x

Figure 1.5 – Représentation par restes du plan $\mathcal{P}(6, 13, 27, 0, 27)$ pour $(x, y) \in [0, 9] \times [0, 6]$. Les points d'appui sont représentés en couleur.

## 1.3 Reconnaissance des droites et plans discrets

Le problème de la reconnaissance de droites ou de plans discrets est d'un intérêt particulier en géométrie discrète puisqu'il a des applications dans de nombreux domaines (analyse et compression d'images, reconnaissance de forme, polygonalisation et polyédrisation d'objets discrets). Il peut être formulé de la façon suivante : étant donné un ensemble $P$ de points entiers, existe-t-il un objet discret (droite ou plan) qui contient tous les points de $P$ ?

De nombreuses études ont été menées sur les segments de droites et les morceaux de plans discrets qui sont les éléments de base de cette géométrie. Ainsi, I. Debled-Rennesson fut la première en 1995 à proposer un algorithme linéaire incrémental de reconnaissance de segments de droites discrètes naïves, aussi bien en deux qu'en trois dimensions, reposant sur les propriétés arithmétiques de ces dernières [DR95]. Elle l'utilisa pour polygonaliser des courbes discrètes et généralisa son idée à la reconnaissance de morceaux rectangulaires de plans discrets naïfs [DRR96].

En 1999, dans le cadre de la polyédrisation d'objets voxels [Pap99, FP99], L. Papier proposa un algorithme de reconnaissance de plans discrets standards basé sur la résolution d'un système linéaire d'inéquations en utilisant l'algorithme de Fourier-Motzkin.

Parallèlement dans [Vit99, VC00], J. Vittone propose une autre méthode utilisant l'espace dual de paramètres $(\alpha, \beta)$ où une droite d'équation $x_0\alpha + \beta - y_0 = 0$ représente un point de coordonnées $(x_0, y_0)$ de l'espace cartésien et un point de coordonnées $(\frac{a}{b}, \frac{r}{b})$ représente une droite d'équation $ax - by + r = 0$ dans l'espace cartésien. L'idée est que chaque pixel se transforme en une droite de l'espace dual, laquelle délimite un demi-plan. L'intersection des demi-plans nous donne alors un ensemble de points dont la représentation dans l'espace primal est un ensemble de droites contenant tous les pixels. En utilisant les suites de Farey[2] et l'algorithme de Grabiner [Gra92], elle propose un algorithme de reconnaissance de droites discrètes, dont la complexité au pire des cas est $\mathcal{O}(n^3 \log n)$, mais qui est amélioré ensuite à $\mathcal{O}(n \log^2 n)$. En introduisant les suites de Farey bidimensionnelles, elle généralise son approche aux morceaux de plans discrets naïfs.

En 2002, L. Buzer présente dans [Buz02] un algorithme incrémental et linéaire permettant de tester la linéarité d'un ensemble de pixels. Il propose pour cela une version incrémentale de l'algorithme de Megiddo [Meg84] très efficace pour de petites dimensions. En donnant un test de coplanarité d'un ensemble de voxels reposant sur la même approche, il généralise son algorithme aux plans discrets, en gardant toujours une complexité linéaire. Mais il n'existe pour l'instant aucune implémentation de cet algorithme en dimension 3.

Plus récemment [Gér03], Y. Gérard a proposé une nouvelle approche basée sur les cordes, c'est-à-dire s'appuyant sur la construction de l'ensemble des différences entre les points discrets d'un ensemble de voxels. Il montre alors que calculer les paramètres d'un plan discret naïf contenant cet ensemble de voxels revient à chercher et à vérifier la position de la face la plus haute sur l'enveloppe convexe de cet ensemble des différences. La complexité théorique du pire cas de cet algorithme est très élevée, $\mathcal{O}(n^7)$, mais des expérimentations[3] montrent qu'il se comporte bien (quasi linéairement) en pratique.

Néanmoins la majorité des méthodes présentées dans cette partie ne proposent que des algorithmes de reconnaissance d'objets réguliers, principalement les droites et plans discrets naïfs. C'est d'ailleurs souvent les propriétés tirées du caractère régulier de ces objets qui font que ces algorithmes sont efficaces. Mais en pratique les données traitées proviennent souvent d'appareils de mesure qui, aussi sophistiqués soient-ils, produisent des données soumises à certaines erreurs. Or ces erreurs détruisent le caractère régulier des objets et donc les algorithmes de reconnaissance connus ne sont plus efficaces.

Une solution à ce problème est de définir analytiquement de nouveaux objets destinés à prendre en compte le bruit induit par les méthodes d'acquisition et de développer de nouveaux algorithmes reposant sur ces nouvelles définitions. La partie qui suit est destinée à présenter ces nouveaux objets.

[2] 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 $n$.

[3] Une implémentation en C de l'algorithme est proposée à l'adresse suivante : `http://www.loria.fr/~debled/plane/index.html`

# Chapitre 2

# Les objets discrets flous

## Sommaire


**2.1 Segments flous** . . . . . **9**
**2.2 Une nouvelle définition : les morceaux de plans discrets flous** . . . . . **11**
**2.3 Générateur de morceaux de plans discrets flous** . . . . . **12**


## Introduction

La notion d'objets discrets flous [DRRRD03], qui correspond à l'étude d'objets discrets bruités, a été introduite en 2003 par I. Debled-Rennesson et al. à travers l'étude des droites discrètes bruitées. Plus récemment une amélioration a été donnée dans [DRFRD05] . Je vais tout d'abord présenter les éléments importants ressortant de cette étude, puis j'introduirai une nouvelle définition pour pouvoir généraliser les méthodes employées aux plans discrets bruités. Je présenterai ensuite un algorithme de génération de morceaux de plans discrets flous.

## 2.1 Segments flous

En reposant sur l'approche arithmétique des droites discrètes, les segments flous les généralisent, en admettant que certains pixels soient « manquants ». Définissons quelques notions qui nous permettrons d'introduire la définition d'un segment flou.

**Définition 2.1.1.** Soit $\mathcal{S}_b$ un ensemble de points. 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 $\mathcal{D}$. On appelle *distance verticale* de la droite $\mathcal{D}(a, b, \mu, \omega)$, la valeur $\frac{\omega-1}{\max(|a|,|b|)}$, qui représente la différence des ordonnées de deux points de même abscisse se situant chacun sur une des deux droites d'appui de $\mathcal{D}$.

**Définition 2.1.2.** Une *droite support* d'un ensemble convexe $C$ est une droite $l$ telle que $C$ est contenu entièrement dans un des demi-plans délimités par $l$ et telle que l'intersection entre $C$ et $l$ n'est pas vide.

**Définition 2.1.3.** La *distance verticale d'un ensemble convexe* $C$ est la distance verticale minimale entre toutes les paires de droites support parallèles. Une autre manière de la définir est comme étant la longueur maximale de l'intersection entre $C$ et une droite verticale.

Ces notions sont illustées à la figure 2.1.

**Définition 2.1.4.** Soit $\mathcal{S}_b$ un ensemble de points 8-connectés[4]. Une droite englobante de $\mathcal{S}_b$ est dite *optimale* si sa distance verticale est minimale, c'est-à-dire si la distance verticale de $\mathcal{S}_b$ est égale à la distance verticale de son enveloppe convexe $\mathcal{CH}(\mathcal{S}_b)$.

[4]Dans le plan il n'y a que deux types de voisinage. Les pixels sont voisins par face ou par sommet. Dans le premier cas ils sont 4-connexes et dans le second cas ils sont 8-connexes.

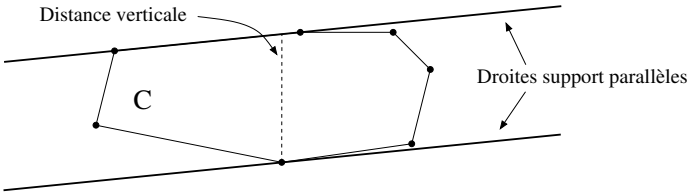



Figure 2.1 – Distance verticale et droites support parallèles d'un ensemble convexe $C$.

Nous pouvons à présent donner la définition du segment flou :

**Définition 2.1.5.** Un ensemble $\mathcal{S}_b$ est un *segment flou d'épaisseur* $\nu$ si et seulement si sa droite englobante optimale a une distance verticale inférieure ou égale à $\nu$.

Cette définition met en évidence la propriété suivante : 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 $\mathcal{S}_b$.

Cette relation n'est pas anodine. En effet, l'avantage de travailler sur l'enveloppe convexe est que cette dernière possède des propriétés intéressantes. Si l'on note $L_i$ et $U_i$ respectivement les plus bas et les plus hauts points d'intersection entre l'enveloppe convexe $C$ et la droite verticale d'équation $x = i$, et $\mathcal{V}(i)$ la distance entre $L_i$ et $U_i$, alors $\mathcal{V}$ a la propriété d'être une fonction concave. Il en résulte que chaque maximum local de $\mathcal{V}$ est aussi un maximum global. De plus il est montré dans [DRFRD05] que ce maximum est atteint lorsque la droite verticale passe par un des sommets de $C$, et que les droites support qui définissent ce maximum sont les droites d'appui inférieures et supérieures de la droite englobante optimale de $C$. On peut ainsi déduire la droite englobante optimale à partir des positions des maxima. C'est sur cette propriété que repose l'algorithme de reconnaissance des segments flous développé dans [DRFRD05] que je vais exposer ci-dessous.

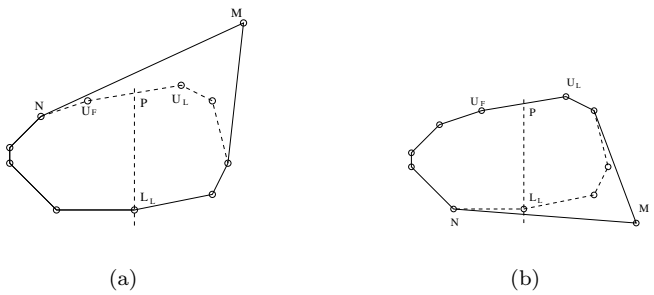



Figure 2.2 – L'ajout d'un nouveau point (a) au-dessus ou (b) en-dessous

Soit $\mathcal{S}_b = \{(x_i, y_i), 0 \leqslant i < n\}$ un segment flou dans le premier octant[5] et $\mathcal{D}(a, b, \mu, \omega)$ sa droite englobante optimale. On suppose 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 2.2. Nous allons ajouter un point $M$. La nouvelle enveloppe convexe $C$ est obtenue par l'algorithme de Melkman[6] [Mel87]. Nous calculons le reste $r$ de $M$ par rapport à $\mathcal{D}$. Si $r \in [0, \omega - 1]$ alors $M \in \mathcal{D}$ et la distance verticale reste la même. Si $r \notin [0, \omega - 1]$ cela signifie que le point a été ajouté au-dessus ou en-dessous de $\mathcal{D}$. Nous devons recalculer la distance verticale maximale. Grâce aux propriétés de l'enveloppe convexe, les auteurs montrent que pour le cas illustré à la figure 2.2(a) la hauteur verticale maximale est obtenue au niveau d'un sommet de $C$ qui se trouve entre $L_L$ et $M$ dans la partie inférieure de l'enveloppe convexe.

[5] Tout comme l'espace 3D peut être divisé en 48 parties, l'espace 2D peut être divisé en 8 parties suivant les symétries par rapport à la droite $Ox$, $Oy$ et la première bisectrice. Le premier octant est le huitième de partie pour lequel $0 \leqslant x \leqslant y,\ y \neq 0$.

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

De même, pour le cas illustré à la figure 2.2(b), la hauteur verticale maximale est obtenue au niveau d'un sommet de $C$ qui se trouve entre $U_L$ et $M$ dans la partie supérieure de l'enveloppe convexe. Et comme $\mathcal{V}$ est concave il n'est pas utile de parcourir tous les points mais seulement de parcourir les points jusqu'à ce que nous tombions sur un maximum local.

Cette méthode a permis aux auteurs de mettre au point un algorithme de reconnaissance incrémental linéraire de segments flous d'épaisseur $\nu$. On ajoute les points par ordre croissant de leur coordonnée $x$ et on applique la méthode décrite ci-dessus. Si au cours de l'algorithme la distance verticale de la droite englobante optimale devient supérieure à $\nu$, alors l'ensemble de points n'est pas un segment flou d'épaisseur $\nu$, sinon il l'est.

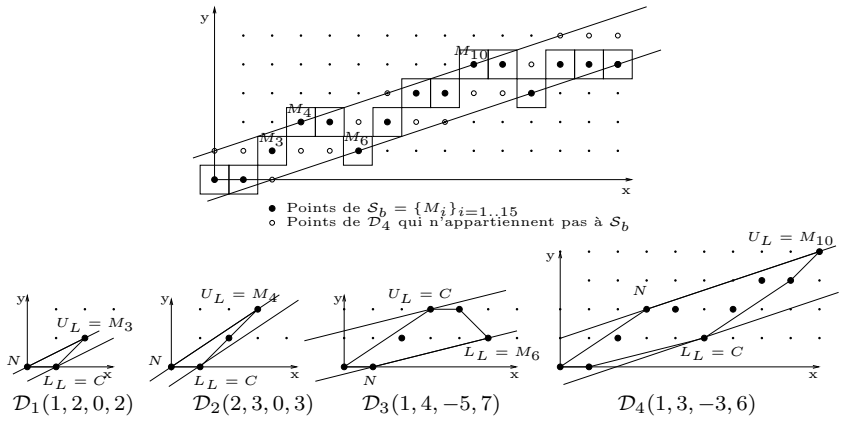



Figure 2.3 – Un exemple de reconnaissance incrémentale d'un segment flou d'épaisseur 2

Un exemple du déroulement de l'algorithme est illustré à la figure 2.3. 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 en-dessous.

## 2.2 Une nouvelle définition : les morceaux de plans discrets flous

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 admettant que certains voxels soient manquants. Aucune recherche n'ayant encore été effectuée en dimension 3, nous allons nous inspirer de la méthode utilisée en dimension 2 pour poser les nouvelles définitions qui permettront de formaliser notre travail. La généralisation de l'approche décrite en 2.1 ne sera néanmoins pas faite stricto sensu. En effet nous allons dans cette partie considérer non pas la distance verticale, mais la distance euclidienne entre les plans afin de définir l'épaisseur de notre morceau de plan discret flou. Cela nous permettra d'utiliser des algorithmes de géométrie algorithmique afin de mettre au point notre méthode de reconnaissance de morceaux de plans discrets flous.

On peut généraliser de façon naturelle la définition de la droite support de la manière suivante :

**Définition 2.2.1.** Soit $E$ un ensemble de points en dimension 3 et $P$ un plan. On dit que $P$ est un *plan support* de $E$ si tous les points de $E$ se trouvent d'un seul et même côté du plan $P$ et que $P \cap E \neq \emptyset$.

Cela nous permet d'introduire la notion d'épaisseur d'un ensemble de points :

**Définition 2.2.2.** Soit $E$ un ensemble de points de dimension 3. On appelle *épaisseur* de $E$, la plus petite distance entre deux plans support parallèles de $E$.

*Rappel.* La distance euclidienne entre un point $M(x_M, y_M, z_M)$ et un plan $\mathcal{P} : ax + by + cz + d = 0$ est la valeur $\frac{|ax_M + by_M + cz_M + d|}{\sqrt{a^2+b^2+c^2}}$. La distance entre deux plans parallèles est alors obtenue en prenant un point sur un des plans et en calculant sa distance par rapport à l'autre plan.

Si nous notons $\mathcal{P}_1$ le plan d'équation $ax + by + cz + d_1 = 0$ et $\mathcal{P}_2$ le plan parallèle à $\mathcal{P}_1$ d'équation $ax + by + cz + d_2 = 0$, alors la distance entre $\mathcal{P}_1$ et $\mathcal{P}_2$ est donnée par la formule $\frac{|d_2-d_1|}{\sqrt{a^2+b^2+c^2}}$. En effet, soit $M(x_M, y_M, z_M)$ un point appartenant à $\mathcal{P}_1$, alors $ax_M + by_M + cz_M + d_1 = 0 \Leftrightarrow ax_M + by_M + cz_M = -d_1$. De plus, d'après la formule donnée en rappel, la distance entre $M$ et $\mathcal{P}_2$ est $\frac{|ax_M+by_M+cz_M+d_2|}{\sqrt{a^2+b^2+c^2}}$. En y substituant l'égalité précédente on obtient le résultat annoncé.

Comme pour les droites support, la généralisation de la notion de droite englobante s'effectue aisément :

**Définition 2.2.3.** Soit $\mathcal{E}_b$ un ensemble de points. 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 $\frac{\omega-1}{\sqrt{a^2+b^2+c^2}}$, qui représente la distance euclidienne entre les deux plans d'appui de $\mathcal{P}$.

**Définition 2.2.4.** Soit $\mathcal{E}_b$ un ensemble de points en dimension 3. Un plan englobant de $\mathcal{E}_b$ est dit *optimal* si sa hauteur est minimale, c'est-à-dire si elle est égale à l'épaisseur de $\mathcal{E}_b$.

Nous avons à présent toutes les notions nécessaires pour pouvoir introduire une nouvelle définition : celle des morceaux de plans discrets flous.

**Définition 2.2.5.** Un ensemble $\mathcal{E}_b$ de points de $\mathbb{Z}^3$ est un *morceau de plan discret flou d'épaisseur* $\nu$ si et seulement si son plan englobant optimal a une hauteur inférieure ou égale à $\nu$.

Cette définition met en évidence une propriété intéressante : le problème de la reconnaissance des morceaux de plans discrets flous est un problème équivalent à celui du calcul de l'épaisseur d'un ensemble de points en dimension 3.

Or il se trouve que le problème du calcul de l'épaisseur d'un ensemble de points en dimension 3 est un problème bien connu en géométrie algorithmique. On peut donc envisager d'utiliser les méthodes existantes, en les adaptant, afin d'en déduire un algorithme de reconnaissance incrémental de morceaux de plans discrets flous. Avant d'expliquer ces méthodes, je vais présenter un algorithme de génération de morceaux de plans discrets flous, réalisé pour les besoins de nos expérimentations

## 2.3 Générateur de morceaux de plans discrets flous

Comme nous le verrons dans la partie 4, notre étude comporte une grande partie expérimentale. En effet, l'approche étant nouvelle, nous avons voulu programmer tous les algorithmes issus de nos conjectures. Cette démarche requiert bien évidemment d'avoir des objets sur lesquels effectuer les tests. C'est pourquoi j'ai mis au point un générateur de morceaux de plans discrets flous.

Dans le cadre du stage de DEA nous avons néanmoins restreint notre étude aux morceaux de plans discrets flous 18-connexes et fonctionnels du premier 48$^{\text{e}}$ d'espace. Cette restriction semble raisonnable et est motivée par les observations suivantes. Premièrement, l'étude dans le premier 48$^{\text{e}}$ d'espace n'est pas vraiment une restriction puisque les symétries du cube nous permettent toujours de nous ramener à une configuration de ce 48$^{\text{e}}$ d'espace. Deuxièmement, la reconnaissance des morceaux de plans discrets flous n'est qu'une première étape en vue de la polyhédrisation d'objets discrets. Or dans ce type de problème nous travaillons habituellement sur les bords de l'objet, et le plus souvent ils sont fonctionnels et 18-connexes.

L'idée est la suivante : nous disposons d'un plan discret épais $\mathcal{P}(a, b, c, \mu, \omega)$, avec $0 \leqslant a \leqslant b \leqslant c,\ c > 0$, et nous voulons générer un morceau bruité de $\mathcal{P}$. Pour cela nous créons aléatoirement un polygone $P$ dans le plan $Oxy$ et, pour tout $(x, y) \in P$, nous calculons l'ensemble de tous les voxels $v(x, y, z)$, pour lesquels $z \in [-\frac{ax+by-\mu}{c}, -\frac{ax+by-\mu-\omega+1}{c}]$. L'ensemble de tous ces voxels forme un morceau épais de plan discret $\mathcal{M}_{\mathcal{P}}$. Il nous reste à le rendre fonctionnel. Pour cela nous partons d'un voxel $v$ choisi arbitrairement dans $\mathcal{M}_{\mathcal{P}}$. Nous parcourons la liste des quarante tricubes possibles dans le 48$^{\text{e}}$ d'espace étudié (cf. annexe A) et nous regardons si $v$ est le centre d'un des tricubes. Si c'est le cas, pour tout voxel $w(x_w, y_w, z_w)$ du tricube retenu centré en $v$, nous supprimons les voxels $(x, y, z)$ tels que $x = x_w, y = y_w$ et $z \neq z_w$. De

manière informelle, on enlève tous les voxels de $\mathcal{M}_{\mathcal{P}}$ qui sont *au-dessus* ou *en-dessous* du tricube centré en $v$. L'ensemble des voxels restants forme notre morceau de plan discret bruité. L'algorithme 1 reprend cette démarche.

**Algorithme 1** : GenererPlanBruite

**Entrées** : $\mathcal{P}(a, b, c, \mu, \omega)$ un plan discret épais.
**Sorties** : Un ensemble de voxels $E$ représentant un morceau bruité et fonctionnel de $\mathcal{P}$.

1 **début**
  *// Étape 1 : Générer un morceau de $\mathcal{P}$*
2  $E \longleftarrow \emptyset$
3  $P \longleftarrow$ Polygone aléatoire du plan $Oxy$
4  **pour tous les** $(x, y) \in P$ **faire**
5  $z_0 \longleftarrow -\frac{ax+by-\mu}{c}$
6  $z_1 \longleftarrow -\frac{ax+by-\mu-\omega+1}{c}$
7  **pour** $z \leftarrow z_0$ **à** $z_1$ **faire**
8  Ajouter le voxel $v(x, y, z)$ à $E$

  *// Étape 2 : Rendre le morceau fonctionnel*
9  $\grave{a}Visiter \longleftarrow \emptyset$
10  $v(x_v, y_v, z_v) \longleftarrow$ un voxel de $E$ pris au hasard
11  $E \longleftarrow E \setminus \{\text{voxels } (x, y, z) \in E \mid x = x_v,\ y = y_v \text{ et } z \neq z_v\}$
12  `AjouterFile`$(v, \grave{a}Visiter)$
13  **tant que** $\grave{a}Visiter \neq \emptyset$ **faire**
14  $v \longleftarrow$ `ExtraireFile`$(\grave{a}Visiter)$
15  Marquer $v$ visité
16  *trouvé* $\longleftarrow$ `RechercherTricube`$(v, E, t)$
17  **si** *trouvé* **alors**
18  **pour** *chacun des 8 voisins $w(x_w, y_w, z_w)$ de $v$ qui appartiennent au tricube $t$* **faire**
19  $E \longleftarrow E \setminus \{\text{voxels } (x, y, z) \in E \mid x = x_w,\ y = y_w \text{ et } z \neq z_w\}$
20  **si** *$w$ n'est pas marqué visité* **et** $w \notin \grave{a}Visiter$ **alors**
21  `AjouterFile`$(w, \grave{a}Visiter)$

22  **retourner** $E$
23 **fin**

*Remarque.* La fonction `RechercherTricube`$(v, E, t)$ à la ligne 16 de l'algorithme 1 va simplement parcourir la liste des 40 tricubes du premier 48$^{e}$ d'espace et retourner dans $t$ le premier tricube centré en $v$ qui se superpose à un voisinage de $v$ qui se trouve dans le plan $\mathcal{P}$.

*Remarque.* La forme du plan bruité obtenu grâce à l'algorithme 1 est très fortement liée à la façon dont on parcourt les voisins du voxel courant à la boucle de la ligne 18. Une étude des différents parcours a été menée dans [Siv04].

Quelques morceaux de plans discrets bruités générés grâce à cet algorithme sont proposés à la figure 2.4.

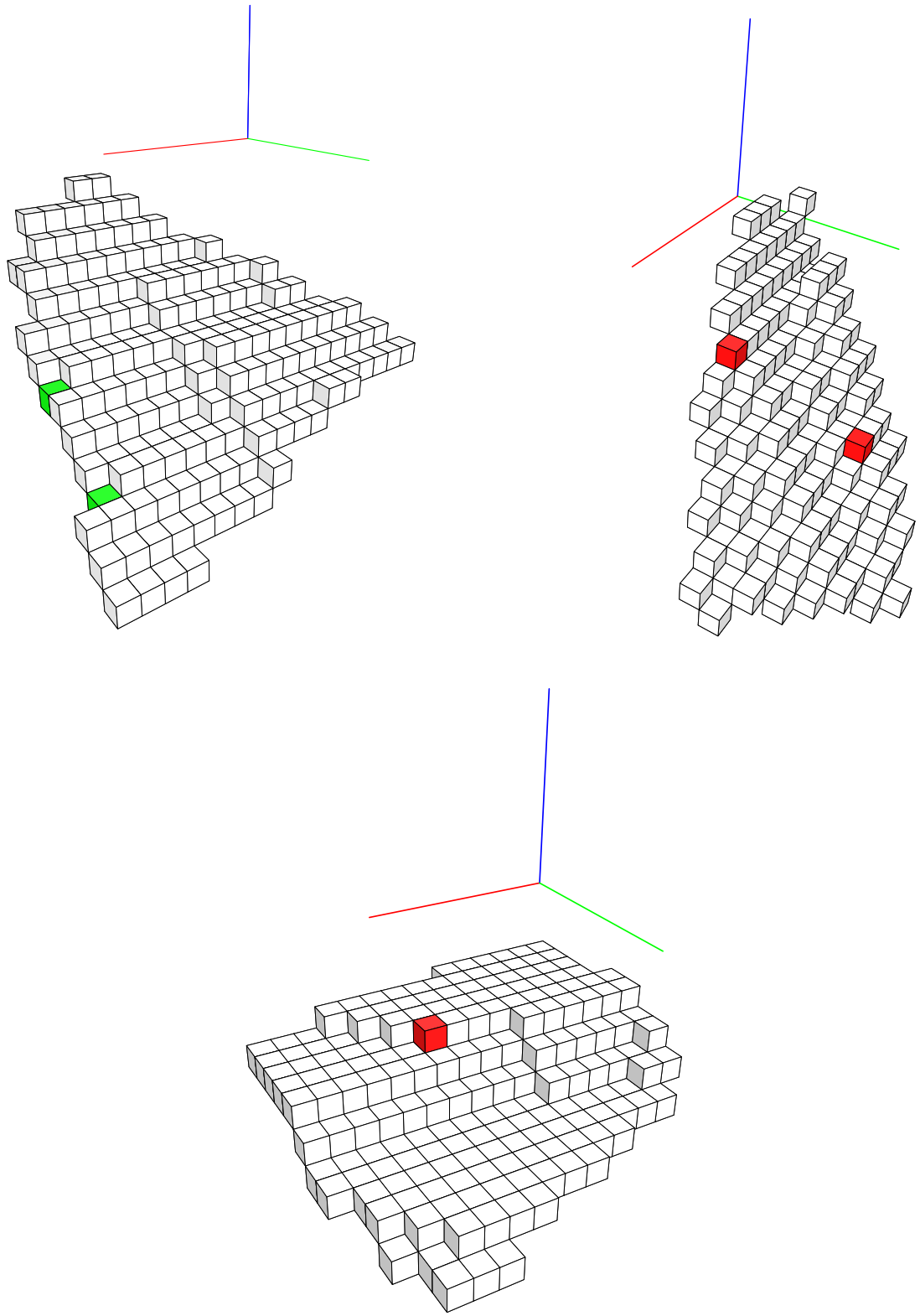


Figure 2.4 – Quelques morceaux de plans discrets flous générés à partir de l'algorithme 1

# Chapitre 3

# Calcul de l'épaisseur d'un ensemble de points

## Sommaire


**3.1 Méthode de Houle et Toussaint** . . . . . **16**
**3.2 Méthode de Gärtner et Herrmann** . . . . . **17**
**3.3 Comparaison des deux méthodes** . . . . . **18**
**3.4 Une autre approche** . . . . . **18**


## Introduction

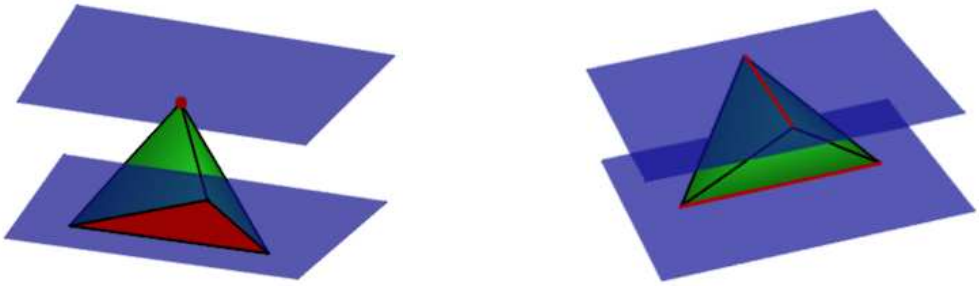
Comme nous l'avons vu dans la partie précédente, la reconnaissance de morceaux de plans flous est un problème équivalent à celui du calcul de l'épaisseur d'un ensemble de points en dimension 3. C'est pourquoi j'ai effectué un état de l'art dans ce domaine. Les différentes méthodes connues ont toutes une étape commune de simplification que je vais présenter, mais avant tout rappelons le problème :
Étant donné un ensemble de points $P$ en dimension 3, nous voulons calculer la plus petite distance entre deux plans support parallèles de $P$.

Premièrement on peut remarquer que l'épaisseur d'un ensemble de $n$ points $P$ est la même que celle de son enveloppe convexe $\mathcal{CH}(P)$. Il devient donc naturel de calculer cette enveloppe convexe dans un premier temps. Cela peut être effectué en $\mathcal{O}(n \log n)$ dans le pire cas [PH77].

Ensuite la notion de paire antipodale est utilisée pour simplifier le problème :

**Définition 3.0.1.** Soient $E$ un ensemble de points et $\mathcal{CH}(E)$ son enveloppe convexe. Deux arêtes $a_1$ et $a_2$ de $\mathcal{CH}(E)$ 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* (*antipodal edge-edge pair*, ou *EE-pair*, en anglais). On définit de la même manière les paires *sommet-sommet*, *sommet-arête*, *sommet-face*, *arête-face* et *face-face* antipodales.

L'épaisseur d'un ensemble de points en dimension 3 est alors obtenue en prenant la distance euclidienne minimale entre deux plans support parallèles passant soit par une paire sommet-face antipodale (figure 3.1(a)), soit par une paire arête-arête antipodale (figure 3.1(b)) de $\mathcal{CH}(P)$. Une démonstration de cette proposition est fournie dans [HT88].



(a) (b)

Figure 3.1 – Les deux paires antipodales pour lesquelles la distance entre les deux plans support qu'elles définissent peut être minimale. (a) Une paire sommet-face antipodale ; (b) Une paire arête-arête antipodale ; les éléments définissant la paire antipodale sont représentés en rouge et les deux plans support parallèles en bleu.

## 3.1 Méthode de Houle et Toussaint

M.E. Houle et G.T. Toussaint [HT88] proposent un algorithme permettant de calculer l'épaisseur d'un ensemble de points, en adaptant une méthode proposée par K.Q. Brown dans sa thèse [Bro79] pour calculer le diamètre[7] d'un ensemble de points. Après les simplifications proposées auparavant, le problème revient à trouver toutes les paires arête-arête et sommet-face antipodales. Pour cela, les auteurs divisent $\mathcal{CH}(P)$ en une *demi-enveloppe supérieure* et une *demi-enveloppe inférieure* de telle sorte qu'un plan contenant une face de la demi-enveloppe supérieure (resp. inférieure) se situe au-dessus (resp. en-dessous) de l'enveloppe convexe. Ils transforment ensuite les faces, arêtes et sommets de chaque demi-enveloppe en utilisant une transformation géométrique introduite par Brown [Bro79] qui s'effectue en temps linéaire.

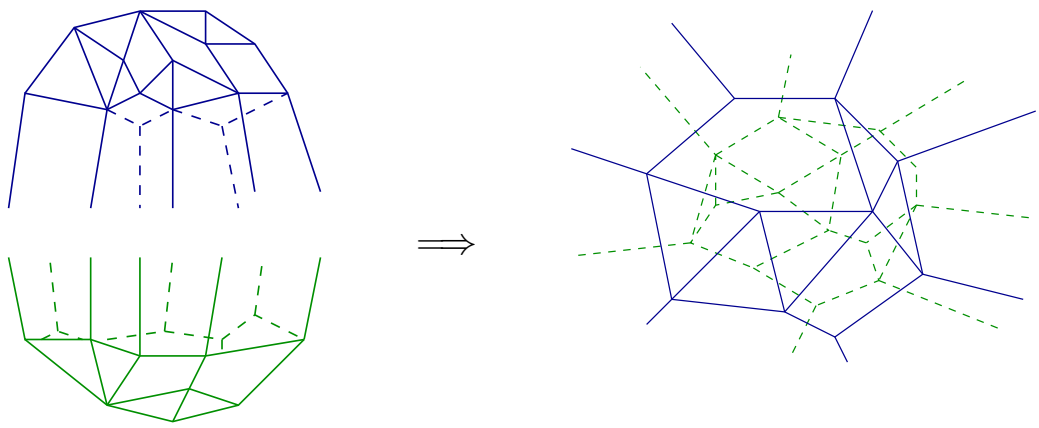
Brown observa que la caractéristique la plus importante des plans support parallèles est qu'ils ont le même vecteur normal. Il décida donc d'ignorer les coordonnées cartésiennes des sommets de $\mathcal{CH}(P)$ qui définissent les éléments formant une paire antipodale, en faveur des vecteurs normaux des plans support. Ainsi un plan qui n'est pas orthogonal au plan $xOy$, possède une équation de la forme $z = ax + by + c$ et se transforme en un point $(a, b) \in \mathbb{R}^2$. Cette transformation peut être vu comme la combinaison d'un passage au dual et d'une projection orthographique.

Houle et Toussaint transforment ainsi leur demi-enveloppe supérieure (resp. inférieure) en une subdivision planaire supérieure (resp. inférieure) comme illustré sur la figure 3.2. Une face, qui est contenue dans un plan support, est transformée en un point de $\mathbb{R}^2$, une arête commune à deux faces est transformée en une arête joignant deux sommets de la subdivision et un sommet est transformé en une région de la subdivision. Cette transformation a la propriété suivante : un sommet issu de l'intersection de deux arêtes de des deux subdivisions représente deux plans support parallèles contenant une paire arête-arête antipodale ; de même qu'un sommet d'une subdivision situé dans une région de l'autre subdivision représente deux plans support parallèles contenant une paire sommet-face antipodale. Ils montrent aussi que deux arêtes infinies parallèles représentent une paire arête-arête antipodale.

Ainsi, en examinant tous les sommets du recouvrement[8] $\mathcal{R}$ de ces deux subdivisions ainsi que les demi-droites parallèles, on peut énumérer toutes les paires antipodales désirées. Le calcul du recouvrement est effectué en utilisant l'algorithme de Guibas et Seidel [GS86]. Cette méthode est une variante de la technique de la droite de balayage utilisé par Bentley et Ottmann dans [BO79] et a une complexité de $\mathcal{O}(n + I)$ où $I$ représente le nombre de paires arête-arête antipodales de $\mathcal{CH}(P)$. La distance entre les

[7] Le diamètre d'un ensemble de point est défini comme étant la plus grande distance entre deux points quelconques de l'ensemble.

[8] Le recouvrement $\mathcal{R}$ de deux subdivisions planaires $\mathcal{S}_1$ et $\mathcal{S}_2$ est la subdivision planaire dont les régions correspondent à toutes les intersections entre une région $R_1$ de $\mathcal{S}_1$ et une région $R_2$ de $\mathcal{S}_2$ : $\mathcal{R} = \{R_1 \cap R_2 \mid R_1 \in \mathcal{S}_1 \text{ et } R_2 \in \mathcal{S}_2\}$.



(a) Les demi-enveloppes convexes

(b) Le recouvrement des subdivisions planaires

Figure 3.2 – La transformation géométrique de Brown

plans support pouvant se calculer en temps constant, on obtient donc un algorithme avec une complexité totale de $\mathcal{O}(n \log n + I)$.

Une implémentation de cet algorithme a été proposé par Schwerdt et al. [SSMJ99]. Une légère modification a été effectuée pour s'affranchir de certaines dégénérescences de l'algorithme de Houle et Toussaint : les demi-enveloppes sont transformées en une instance de graphe planaire embarquée dans la demi-sphère unitaire positive $\mathbb{S}^2_+$. Ils utilisent alors toujours une technique de balayage à la Bentley et Ottmann pour retrouver les paires antipodales, mais adaptée à la demi-shpère.

## 3.2 Méthode de Gärtner et Herrmann

En 2001, B. Gärtner et T. Herrmann ont proposé une approche directe en énumérant les paires antipodales directement [GH01]. Pour cela ils reformulent le problème du calcul de l'épaisseur d'un ensemble $E$ de $n$ points en dimension 3, en terme de problème d'optimisation avec contraintes linéaires, mais fonction objective non convexe. Ainsi ils recherchent deux plans support parallèles $h_1 : ax + by + cz + d = 0$ et $h_2 : ax + by + cz + 1 = 0$ ayant une distance minimale. C'est-à-dire que le problème revient à trouver $a, b, c$ et $d$ minimisant la distance au carré $\frac{(1-d)^2}{a^2+b^2+c^2}$ tels que les contraintes

$$ap_x + bp_y + cp_z + d \leqslant 0$$

et

$$ap_x + bp_y + cp_z + 1 \geqslant 0$$

soient satisfaites pour tous les points $p(p_x, p_y, p_z) \in E$.

Pour trouver toutes les paires antipodales ils généralisent en quelque sorte la méthode du « rotating caliper » [Tou83] au cas tridimensionnel. Ils choisissent une face de départ $f$ sur l'enveloppe convexe $\mathcal{CH}(P)$ et calculent ses sommets antipodaux $V = \{v_1, \ldots, v_k\}$. Le coût de cette opération est linéaire. Une rotation des plans $h_1$ et $h_2$, supportant $V$ et $f$ respectivement, est effectuée autour d'une arête $a$ incidente à $f$, jusqu'à ce que $h_2$ contienne l'autre face $f'$ incidente à $a$. Durant la rotation, le parallélisme entre les plan $h_1$ et $h_2$ est conservé et on reporte toutes les paires arête-arête antipodales contenant $a$ ainsi que les paires face-sommet antipodales contenant $f'$.

La partie importante est la suivante : étant donnés une paire sommet-arête antipodale $(w, a)$ et des plans parallèles $h_1$ et $h_2$ contenant respectivement $w$ et $a$, lorsqu'on effectue une rotation de $h_2$ autour de $a$, deux événements peuvent se produire :

1. $h_2$ rencontre une nouvelle face $f'$ incidente à $a$, auquel cas on vient de trouver une paire sommet-face antipodale $(w, f')$,
2. $h_1$ rencontre un nouveau sommet $v$, auquel cas on vient de trouver une paire arête-arête antipodale $((wv), a)$ .

Dans la partie 4 nous ferons référence à cette procédure de rotation sous le nom `EffectuerRotation`. Pendant la rotation, les sommets sont marqués pour pouvoir éviter de tester deux fois la même arête. Ainsi une rotation s'exécute en temps linéaire. En répétant l'opération pour les $\mathcal{O}(n)$ arêtes de $\mathcal{CH}(P)$ on obtient un algorithme de complexité au pire des cas $\mathcal{O}(n^2)$.

Une version de cet algorithme est implémentée dans la librairie CGAL[9] au niveau de la classe `Width_3`.

## 3.3 Comparaison des deux méthodes

Comparons dans un premier temps la complexité des deux algorithmes. L'algorithme de Houle et Toussaint a l'avantage d'avoir une complexité $\mathcal{O}(n \log n + I)$ qui est sensible au nombre de paires arête-arête antipodales de $\mathcal{CH}(P)$. D'ailleurs l'implémentation de Schwerdt [SSMJ99] confirme qu'en pratique l'algorithme se comporte relativement bien quand $I$ est petit. Néanmoins Houle et Toussaint ont mis en évidence des cas où $I = \Theta(n^2)$, donc au pire des cas l'algorithme a une complexité en $\mathcal{O}(n^2)$.

L'algorithme de Gärtner et Herrmann a lui aussi une complexité théorique au pire cas de $\mathcal{O}(n^2)$ mais, même si le nombre de paires antipodales arête-arête est petit, la complexité peut être $\Theta(n^2)$. Des entrées potentiellement mauvaises pour l'algorithme sont des polyèdres avec des sommets de grand degré. Il peut alors arriver que le voisinage de ces sommets soit testé lors de nombreuses rotations.

Puisque les deux algorithmes ont été programmés, Gärtner et Herrmann ont effectué un comparatif des performances de leur implémentation et de celle de Schwerdt. Il en ressort qu'en pratique l'implémentation de Gärtner et Herrmann est plus de deux fois plus rapide que celle de Schwerdt présentée en 3.1. Cela peut en partie s'expliquer par la gestion des dégénérescences de l'algorithme de Houle et Toussaint. En effet, pour pouvoir effectuer les transformations géométriques de Brown, il faut que les faces de l'enveloppe convexe ne soient pas verticales. Or cette configuration peut arriver fréquemment en pratique. Pour éviter que cette configuration ne pose problème Schwerdt a utilisé une méthode de projection dans la demi-sphère positive $\mathbb{S}^2_+$. Mais l'algorithme de balayage est alors plus coûteux.

Au contraire les seules dégénérescences qui peuvent se produire dans l'algorithme de Gärtner et Herrmann sont dues au fait que, lors de la rotation, le plan $h_1$ rencontre plusieurs sommets en même temps. Mais ce cas de figure est beaucoup plus simple à gérer. En effet les auteurs montrent qu'un parcours de graphe au niveau du voisinage des sommets suffit à régler le problème.

Un autre avantage de l'algorithme de Gärtner et Herrmann est qu'il utilise directement la géométrie de l'enveloppe convexe pour trouver les paires antipodales. Il est donc plus adapté à la méthode que nous utilisons dans le cadre de la reconnaissance des morceaux de plans flous. De plus, il semble beaucoup plus facile à utiliser dans un processus incrémental que l'algorithme de Houle et Toussaint. C'est pourquoi nous avons choisi cet algorithme pour nos programmes.

## 3.4 Une autre approche

Pour terminer cet état de l'art, je vais brièvement présenter une autre approche qui permet une amélioration de la complexité.

En 1992, B. Chazelle et al. [CEGS92] proposent d'appliquer la technique de recherche paramétrique de Megiddo (*Megiddo's parametric searching technique*) au problème du calcul de l'épaisseur. Cette méthode repose sur le fait que l'analyse d'algorithmes parallèles permet d'obtenir des algorithmes séquentiels efficaces pour certains problèmes. Pour plus d'information sur cette technique, on peut se référer à [Meg83]. En appliquant ce paradigme au calcul de l'épaisseur, les auteurs parviennent à améliorer la complexité à $\mathcal{O}(n^{8/5+\epsilon})$, pour tout $\epsilon > 0$.

[9] `http://www.cgal.org/`

En 1994, P.K. Agarwal et al. [AAS94] reprennent la même technique. Pour la partie coûteuse de l'algorithme – la recherche des paires arêtes-arêtes antipodales – le problème du calcul de l'épaisseur est réduit à celui du calcul de la plus proche paire de lignes (*closest line-pair problem*). Ce dernier problème peut être formulé de la façon suivante :

Soit un ensemble $\mathcal{L}$ de $n$ lignes dans $\mathbb{R}^3$. On définit la distance $d(l_1, l_2)$ entre deux lignes de $\mathcal{L}$ comme étant le $\min_{p \in l_1, q \in l_2} d(p, q)$ où $d(p, q)$ représente la distance euclidienne entre les points $p$ et $q$. Le problème de la plus proche paire de lignes est de calculer

$$\min_{l_1, l_2 \in \mathcal{L}} d(l_1, l_2)$$

Pour résoudre ce problème les auteurs utilisent un algorithme diviser-pour-régner randomisé. En combinant cet algorithme avec la technique de recherche paramétrique de Megiddo ils arrivent à une complexité moyenne de $\mathcal{O}(n^{17/11+\epsilon})$, pour tout $\epsilon > 0$. L'année suivante [AS95] ils améliorent cette complexité à $\mathcal{O}(n^{3/2+\epsilon})$, pour tout $\epsilon > 0$.

À ma connaissance il n'existe aucune implémentation de ces algorithmes. En outre, il semble assez peu intéressant d'utiliser ces méthodes pour notre problème de reconnaissance de morceaux de plans flous. D'une part il paraît difficile de rendre ces méthodes incrémentales, d'autre part pour atteindre la meilleure complexité connue, à savoir $\mathcal{O}(n^{1,5})$, il faut utiliser des structures de données lourdes à gérer qui en pratique donnerait probablement de moins bons résultats que les implémentations précédentes.

# Chapitre 4

# Étude expérimentale

## Sommaire


**4.1 Approche initiale** . . . . . **21**
4.1.1 Parcours choisi . . . . . 21
4.1.2 Premier algorithme de reconnaissance . . . . . 21
**4.2 Premières observations et nouvelle version** . . . . . **22**
4.2.1 Calcul incrémental de l'enveloppe convexe . . . . . 23
4.2.2 Algorithme résultant . . . . . 24
**4.3 Tentative d'amélioration de l'algorithme incrémental** . . . . . **25**
**4.4 Algorithme incrémental retenu** . . . . . **27**


## Introduction

Comme annoncé dans l'introduction générale, l'objectif du stage est de mettre au point un algorithme incrémental de reconnaissance de morceaux de plans discrets flous. Nous avons déjà vu que ce problème est équivalent à celui du calcul de l'épaisseur d'un ensemble de points en dimension 3. J'ai présenté plusieurs algorithmes issus du domaine de la géométrie algorithmique capable de résoudre ce problème mais aucun d'eux n'est incrémental. L'idée initiale du stage fut donc de générer aléatoirement des morceaux de plans discrets flous afin de simuler une reconnaissance incrémentale de ces derniers en utilisant un des algorithmes présentés dans la partie 3. À chaque étape nous ajoutons un point et nous recalculons l'épaisseur sur l'ensemble des points. Cela nous permet d'observer les différentes modifications qui surviennent à chacune des étapes. On espère ainsi pouvoir trouver des simplifications nous permettant d'améliorer l'algorithme.

Étant donné que notre approche repose sur l'enveloppe convexe en dimension 3 et que nous savons qu'une borne inférieure de ce problème est en $\Omega(n \log n)$, nous n'espérons pas trouver un algorithme avec une meilleure complexité, par exemple linéaire comme pour les segments flous (cf. paragraphe 2.1).

Afin d'observer l'évolution de l'épaisseur au cours de la reconnaissance du morceau de plan flou, j'ai dû développer une interface de visualisation 3D dont quelques captures d'écran sont fournies à l'annexe B. Le développement s'est effectué en C++. J'ai utilisé OpenGL[10] pour la partie visualisation 3D, Qt[11] pour l'interface graphique et les vues 2D représentant les projections dans le plan $Oxy$, et la librairie graphique CGAL[12] pour les tests géométriques (position d'un point par rapport à un plan, coplanarité de quatre points, etc...), pour la gestion de la structure en demi-arête des maillages représentant les polyèdres et pour le calcul de l'épaisseur du premier algorithme incrémental proposé. CGAL est un projet commun

[10] http://www.opengl.org/
[11] http://www.trolltech.com/
[12] http://www.cgal.org/

entre plusieurs universités d'Europe et d'Israël, dont l'implémentation repose sur la programmation générique en C++, et fait donc appel à des notions évoluées du langage comme les templates, la spécialisation ou encore les classes de traits, j'ai donc du lire beaucoup de documentation sur ces concepts afin d'en comprendre le fonctionnement [VJ02].

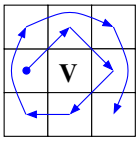
## 4.1 Approche initiale

### 4.1.1 Parcours choisi

Pour pouvoir simuler une reconnaissance incrémentale nous avons dit que nous allions ajouter les voxels les uns après les autres. Nous pouvons alors nous demander dans quel ordre les ajouter. Rappelons que nous travaillons avec des morceaux fonctionnels de plans discrets flous du premier 48$^{e}$ d'espace. Nous pouvons donc simplifier le parcours en considérant la projection de ces morceaux dans le plan $Oxy$. En gardant à l'esprit que la reconnaissance des morceaux de plans discrets flous n'est qu'une première étape en vue de la polyédrisation d'objets voxels, et que dans ce type de problème nous essayons d'avoir des faces les plus grandes possibles, nous allons ajouter les voxels de façon à étendre la face régulièrement dans le plan $Oxy$ à partir d'un point donné. Une façon d'envisager le parcours est donc de s'éloigner de façon à former des cercles concentriques à un voxel de départ. Ce parcours fait penser à un parcours géodésique, mais il ne repose pas sur la distance géodésique, nous avons donc décidé de le nommer *parcours pseudo-géodésique*.

Pour programmer ce type de parcours nous avons utilisé une file. La file est initialisée avec le centre d'un tricube de départ choisi au hasard dans le morceau de plan flou. A chaque étape, le prochain voxel $v$ du parcours est en tête de la file. On l'extrait de la file et on ajoute en queue de file les 26-voisins de $v$ qui appartiennent au morceau de plan flou et qui n'ont pas encore été visités, dans l'ordre proposé à la figure 4.1(a). La figure 4.1(b) présente la projection dans le plan $Oxy$ d'un morceau de plan dont 44 voxels ont été ajoutés à partir de $V_0$ suivant le parcours décrit précédemment.

D'autres types de parcours sont présentés dans [Coe02, Siv04].




(a)

| | | | | | | |
|---|---|---|---|---|---|---|
| | 33 | 31 | 30 | 32 | 34 | |
| 29 | 22 | 13 | 12 | 14 | 23 | 38 |
| 27 | 11 | 6 | 2 | 7 | 16 | 36 |
| 25 | 9 | 1 | $\mathbf{V_0}$ | 3 | 15 | 35 |
| 26 | 10 | 5 | 4 | 8 | 17 | 37 |
| 28 | 21 | 19 | 18 | 20 | 24 | 39 |
| | 43 | 41 | 40 | 42 | 44 | |

(b)

Figure 4.1 – (a) Parcours des voxels d'un tricube centré en $V$ : les voisins 4-connexes sont parcourus avant les voisins 8-connexes ; (b) Parcours pseudo-géodésique d'un morceau de plan discret fonctionnel à partir de $V_0$. Les voxels sont visités suivant la numérotation croissante.

### 4.1.2 Premier algorithme de reconnaissance

Nous pouvons maintenant présenter un algorithme naïf de reconnaissance de morceaux de plans flous. Soit $\mathcal{E}_b$ un morceau de plan flou généré par l'algorithme 1 et $\nu$ l'épaisseur maximale du morceau de plan discret flou à reconnaître. Nous choisissons arbitrairement un tricube de départ $t$ dans $\mathcal{E}_b$. Ceci nous

assure que l'épaisseur est bien définie. Puis nous ajoutons les autres voxels un à un en suivant un parcours pseudo-géodésique à partir du centre de $t$ (figure 4.1). A chaque étape nous recalculons l'épaisseur de l'ensemble des points déjà ajoutés en utilisant l'algorithme de Gärtner et Herrmann étudié dans la partie 3.2 et nous pouvons en déduire à chaque étape les caractéristiques du plan englobant optimal. Si l'épaisseur est supérieure à $\nu$ alors $\mathcal{E}_b$ n'est pas un morceau de plan discret flou d'épaisseur $\nu$, sinon il l'est. L'algorithme 2 reprend cette démarche.

**Algorithme 2** : Reconnaissance naïve incrémentale de morceaux de plans flous

**Entrées** : $\mathcal{E}_b$ un morceau de plan discret flou, $\nu$ un réel.
**Sorties** : $estMorceauPlanFlou$ un booléen, vrai si $\mathcal{E}_b$ est un morceau de plan discret flou d'épaisseur $\nu$, faux sinon. Dans le cas où $estMorceauPlanFlou$ est vrai, $a, b, c, \mu$ et $\omega$ représentent les caractéristiques du plan englobant optimal de $\mathcal{E}_b$.

1 **début**
2 $\quad estMorceauPlanFlou \longleftarrow$ vrai
3 $\quad E \longleftarrow$ l'ensemble des voxels d'un tricube $t$ de départ choisi au hasard dans $\mathcal{E}_b$
4 $\quad$ **tant que** $estMorceauPlanFlou$ **et** $\mathcal{E}_b$ *n'a pas été totalement parcouru* **faire**
5 $\quad\quad$ *épaisseur* $\longleftarrow \infty$
6 $\quad\quad$ Ajouter à $E$ le prochain voxel de $\mathcal{E}_b$ en suivant un parcours pseudo-géodésique
7 $\quad\quad C \longleftarrow$ enveloppe convexe de $E$
8 $\quad\quad$ **pour chaque** *arête* $e$ *de* $C$ **faire**
9 $\quad\quad\quad \langle a_t, b_t, c_t, d_{t1}, d_{t2}\rangle \longleftarrow$ `EffectuerRotation(`$e$`)`
10 $\quad\quad\quad$ **si** $\frac{|d_{t2}-d_{t1}|}{\sqrt{a^2+b^2+c^2}} <$ *épaisseur* **alors**
11 $\quad\quad\quad\quad$ *épaisseur* $\longleftarrow \frac{|d_{t2}-d_{t1}|}{\sqrt{a^2+b^2+c^2}}$
12 $\quad\quad\quad\quad a \longleftarrow a_t$
13 $\quad\quad\quad\quad b \longleftarrow b_t$
14 $\quad\quad\quad\quad c \longleftarrow c_t$
15 $\quad\quad\quad\quad \mu \longleftarrow \min(-d_{t1}, -d_{t2})$
16 $\quad\quad\quad\quad \omega \longleftarrow |d_{t2} - d_{t1}| + 1$
17 $\quad\quad$ **si** *épaisseur* $> \nu$ **alors**
18 $\quad\quad\quad estMorceauPlanFlou \longleftarrow$ faux ;
19 $\quad$ **retourner** $estMorceauPlanFlou$
20 **fin**

Si le morceau de plan discret flou $\mathcal{E}_b$ de l'algorithme 2 est formé de $n$ voxels, alors la boucle de la ligne 4 peut comporter au plus $n$ itérations. Le calcul de l'enveloppe convexe effectué à la ligne 7 s'effectue en $\mathcal{O}(n \log n)$ (cf. [PH77]). La boucle de calcul de l'épaisseur à la ligne 8 parcours chaque arête de l'enveloppe convexe $C$ et pour chacune d'elle effectue la procédure de rotation `EffectuerRotation` décrite dans la partie 3.2. Les valeurs renvoyées sont les coefficients des plans support parallèles $h_1 : ax+by+cz+d_1 = 0$ et $h_2 : ax + by + cz + d_2 = 0$ dont la distance est minimale pour cette rotation. Étant donné qu'il y a $\mathcal{O}(n)$ arêtes dans $C$ et que `EffectuerRotation` prend un temps de $\mathcal{O}(n)$, alors la boucle de calcul de l'épaisseur se fait au total en $\mathcal{O}(n^2)$. La complexité totale au pire des cas de l'algorithme 2 est donc de $\mathcal{O}(n^3)$. Cette complexité n'est bien sûr pas optimale, mais en pratique le programme est assez rapide pour pouvoir effectuer nos expérimentations.

## 4.2 Premières observations et nouvelle version

**Observation 1.** Les définitions de la partie 2.2 mettent en évidence le fait que les plans support parallèles définissant l'épaisseur d'un ensemble de points en dimension 3 sont les mêmes que les plans d'appui du plan englobant optimal $\mathcal{P}_E$ de ce même ensemble de points. Nous pouvons donc obtenir les caractéristiques

$a, b, c, \mu$ et $\omega$ de $\mathcal{P}_E$ à partir des plans support parallèles obtenus au cours du calcul de l'épaisseur. Ainsi nous pouvons calculer le reste par rapport à $\mathcal{P}_E$ (cf. partie 1.2.2) de chaque point $M$ que nous ajoutons au cours de la reconnaissance. Or, d'après la remarque de la définition 1.2.6, si $r_{(a,b,c,\mu)}(M) \in [0, \omega - 1]$ alors le point $M$ appartient à $\mathcal{P}_E$. Dans ce cas $\mathcal{P}_E$ n'est pas modifié après l'ajout de $M$. Ainsi, une première simplification à la méthode proposée dans la partie 4.1.2, consiste à recalculer l'épaisseur uniquement lorsque le point que nous ajoutons a un reste qui n'est pas dans l'intervalle $[0, \omega - 1]$.

**Observation 2.** L'enveloppe convexe construite dans l'algorithme de Gärtner et Herrmann est recalculée à chaque étape de l'algorithme 2 sur un ensemble de points qui ne diffère que d'un point d'une étape à l'autre. Nous pouvons donc avantageusement remplacer ce procédé par un algorithme incrémental de calcul d'enveloppe convexe.

### 4.2.1 Calcul incrémental de l'enveloppe convexe

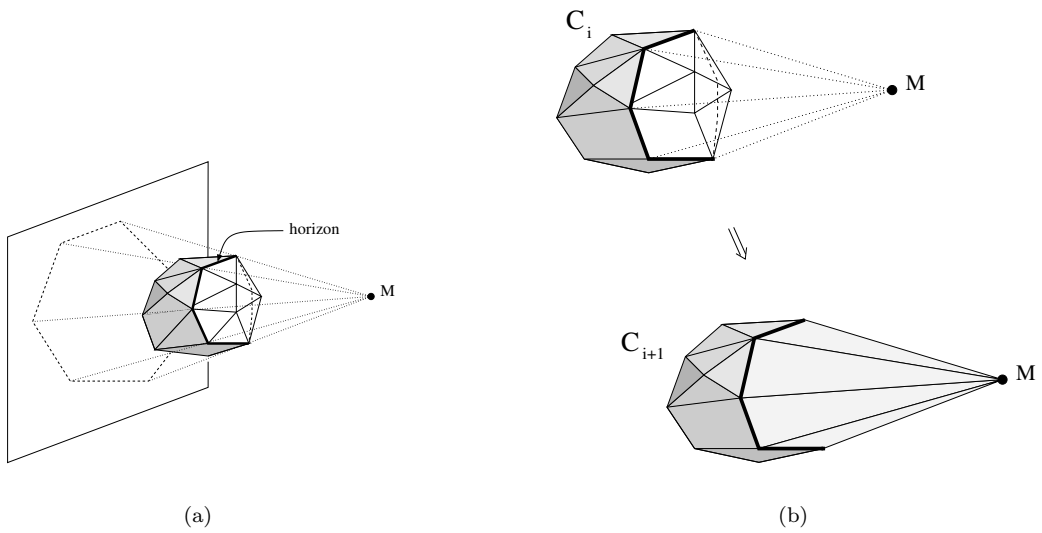



Figure 4.2 – (a)La ligne d'horizon de l'enveloppe convexe ; (b) L'évolution de l'enveloppe convexe après ajout du point $M$.

Pour notre algorithme de reconnaissance incrémental nous avons programmé l'algorithme décrit dans le livre [dBvKOS00]. Ce dernier repose sur l'idée suivante : à une étape $i$ de l'algorithme nous avons une enveloppe convexe $C_i$. Nous ajoutons un point $M$. S'il se trouve à l'intérieur de $C_i$ alors nous ne faisons rien. Dans le cas contraire nous recherchons toutes les faces de $C_i$ visibles[13] à partir de $M$. Cet ensemble de faces forme une région délimitée par une courbe qu'on appelle *horizon* (figure 4.2(a)). Toutes ces faces sont alors supprimées et chaque extrémité des segments de l'horizon est relié au nouveau point $M$(figure 4.2(b)) afin de former de nouvelles faces triangulaires. 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 $C_{i+1}$.

Pour un ensemble de $n$ points, la complexité de cet algorithme est en moyenne de $\mathcal{O}(n \log n)$ si on l'utilise dans sa version off-line randomisée, c'est-à-dire si l'on connaît l'ensemble des points que l'on va ajouter au cours de l'algorithme et qu'on les ajoute dans un ordre aléatoire. On peut alors effectuer une étape de pré-traitement afin de calculer le *graphe de conflit*. Cette structure de donnée est un graphe

[13]Chaque face de l'enveloppe convexe définit un plan qui coupe l'espace en deux sous-espaces, l'un qui contient toute l'enveloppe convexe et que l'on qualifie de fermé et l'autre que l'on qualifie d'ouvert. Une face de l'enveloppe convexe est visible à partir d'un point $P$ si ce dernier est dans le demi-espace ouvert délimité par le plan qui contient la face.

biparti permettant de connaître l'ensemble des faces visibles depuis un point en temps proportionnel à la taille du graphe. Il faut alors mettre à jour ce graphe à chaque étape de l'algorithme.

Dans sa version on-line, c'est-à-dire lorsqu'on ne possède aucune information sur les points que l'on va ajouter, la complexité est au pire des cas de $\mathcal{O}(n^2)$ car à chaque étape il faut parcourir l'ensemble des faces de l'enveloppe convexe pour faire les tests de visibilité à partir du point $M$. Pour le problème de reconnaissance de morceaux de plans discrets flous nous pouvons envisager d'utiliser la version off-line puisque nous connaissons l'ensemble des points que l'on va ajouter. Mais pour les applications envisagées de cet algorithme, comme la polyédrisation d'objets voxels, il semble que l'utilisation de la méthode on-line s'impose si nous ne voulons pas prendre en compte tous les voxels de l'objet.

### 4.2.2 Algorithme résultant

**Algorithme 3** : Reconnaissance naïve incrementale améliorée

**Entrées** : $\mathcal{E}_b$ un morceau de plan discret flou, $\nu$ un réel.
**Sorties** : $estMorceauPlanFlou$ un booléen, vrai si $\mathcal{E}_b$ est un morceau de plan discret flou d'épaisseur $\nu$, faux sinon. Dans le cas où $estMorceauPlanFlou$ est vrai, $a, b, c, \mu$ et $\omega$ représentent les caractéristiques du plan englobant optimal de $\mathcal{E}_b$.

1 **début**
  *// Initialisation*
2  $estMorceauPlanFlou \longleftarrow$ vrai
3  $C \longleftarrow$ enveloppe convexe d'un tricube $t$ de départ choisi au hasard dans $\mathcal{E}_b$
4  $\langle a, b, c, d_1, d_2 \rangle \longleftarrow$ `CalculerEpaisseur`$(C)$
5  $\mu \longleftarrow \min(d_1, d_2)$
6  $\omega \longleftarrow |d_2 - d_1| + 1$
  *// Boucle principale*
7  **tant que** $estMorceauPlanFlou$ **et** $\mathcal{E}_b$ *n'a pas été totalement parcouru* **faire**
8  *épaisseur* $\longleftarrow \infty$
9  Soit $M$ le prochain voxel dans le parcours pseudo-géodésique
10  Mettre à jour $C$ en ajoutant $M$ avec la méthode incrémentale décrite en 4.2.1
11  $r \longleftarrow ax_M + by_M + cz_M - \mu$
12  **si** $r \notin [0, \omega - 1]$ **alors**
13  **pour chaque** *arête* $e$ *de* $C$ **faire**
14  $\langle a_t, b_t, c_t, d_{t1}, d_{t2} \rangle \longleftarrow$ `EffectuerRotation`$(e)$
15  **si** $\frac{|d_{t2} - d_{t1}|}{\sqrt{a^2+b^2+c^2}} <$ *épaisseur* **alors**
16  *épaisseur* $\longleftarrow \frac{|d_{t2} - d_{t1}|}{\sqrt{a^2+b^2+c^2}}$
17  $a \longleftarrow a_t$
18  $b \longleftarrow b_t$
19  $c \longleftarrow c_t$
20  $\mu \longleftarrow \min(-d_{t1}, -d_{t2})$
21  $\omega \longleftarrow |d_{t2} - d_{t1}| + 1$
22  **si** *épaisseur* $> \nu$ **alors**
23  $estMorceauPlanFlou \longleftarrow$ faux ;
24  **retourner** $estMorceauPlanFlou$
25 **fin**

Ces observations nous amènent à proposer une amélioration de l'algorithme 2. Nous allons tout d'abord remplacer le calcul de l'enveloppe convexe effectué sur l'ensemble des points à chaque étape par l'algorithme incrémental de calcul d'enveloppe convexe présenté dans la partie précédente. Ce dernier ne modifie que les parties de l'enveloppe convexe qui sont visibles depuis le point ajouté. Ainsi, si le point ajouté

ne voit qu'une petite partie de l'enveloppe convexe – ce qui arrive généralement en pratique – le nombre d'opérations effectuées sera petit ce qui améliorera l'algorithme en pratique. Ensuite, les caractéristiques $a, b, c, \nu$ et $\omega$ du plan englobant optimal nous permettent de calculer le reste du point par rapport à ce plan. Or nous avons vu que si ce reste n'appartient pas à l'intervalle $[0, \omega - 1]$ alors l'épaisseur de l'ensemble de points ne change pas et donc les rotations autour des arêtes de l'enveloppe convexe sont inutiles puisqu'elles vont retourner les caractéristiques du plan englobant optimal actuel. Nous effectuons donc les rotation uniquement si le reste est dans l'intervalle $[0, \omega - 1]$. La démarche est récapitulée dans l'algorithme 3.

Supposons que nous ayons à reconnaître un ensemble $\mathcal{E}_b$ contenant $n$ points. Les calculs de l'enveloppe convexe et de l'épaisseur initiales aux lignes 3 et 4 de l'algorithme 3 se font sur les voxels d'un tricube donc leur coût est constant. Les boucles des lignes 7 et 13 comportent $\mathcal{O}(n)$ itérations. La fonction `EffectuerRotation(`$e$`)` à la ligne 14 effectue la procédure de rotation décrite dans la partie 3.2. Nous avons déjà vu que le coût de cette rotation est en $\mathcal{O}(n)$.

Nous obtenons donc un algorithme de même complexité théorique au pire des cas que l'algorithme 2, à savoir $\mathcal{O}(n^3)$, mais plus rapide en pratique puisque les rotations ne sont pas effectuées à chaque étape.

## 4.3 Tentative d'amélioration de l'algorithme incrémental

À travers les nombreux tests de reconnaissance de morceaux de plans flous que nous avons effectués, nous avons pu constater un autre fait intéressant.

**Observation 3.** Chaque point ajouté qui modifie les caractéristiques du plan englobant optimal semble devenir un point d'appui du nouveau plan englobant optimal. Le même résultat a été obtenu lors de l'étude des segments flous et les nombreuses reconnaissances de morceaux de plans flous testées tendent à confirmer cette observation (cf. figure 4.3).

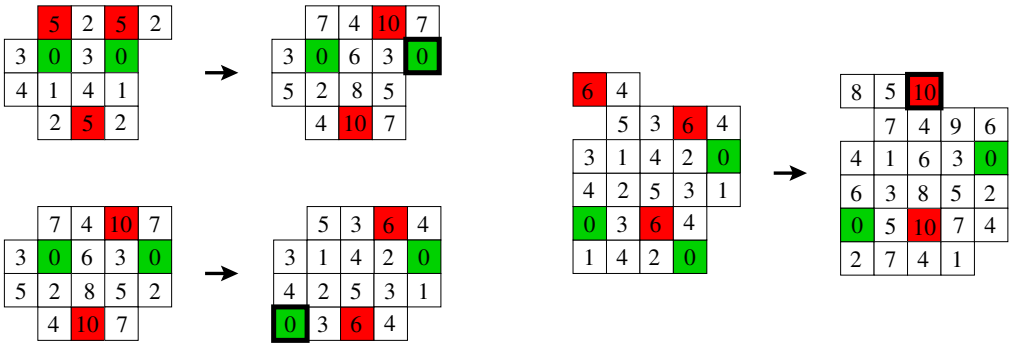


Figure 4.3 – Vue par restes de trois étapes de la reconnaissance d'un morceau de plan discret flou. Les points d'appui inférieurs (resp. supérieurs) du plan englobant optimal sont représentés en vert (resp. rouge). A chaque étape, le point entouré en gras est ajouté et modifie les caractéristiques du plan englobant optimal. De plus il devient point d'appui du nouveau plan englobant optimal.

Mais finalement nous sommes tombés sur le contre-exemple illustré à la figure 4.4.

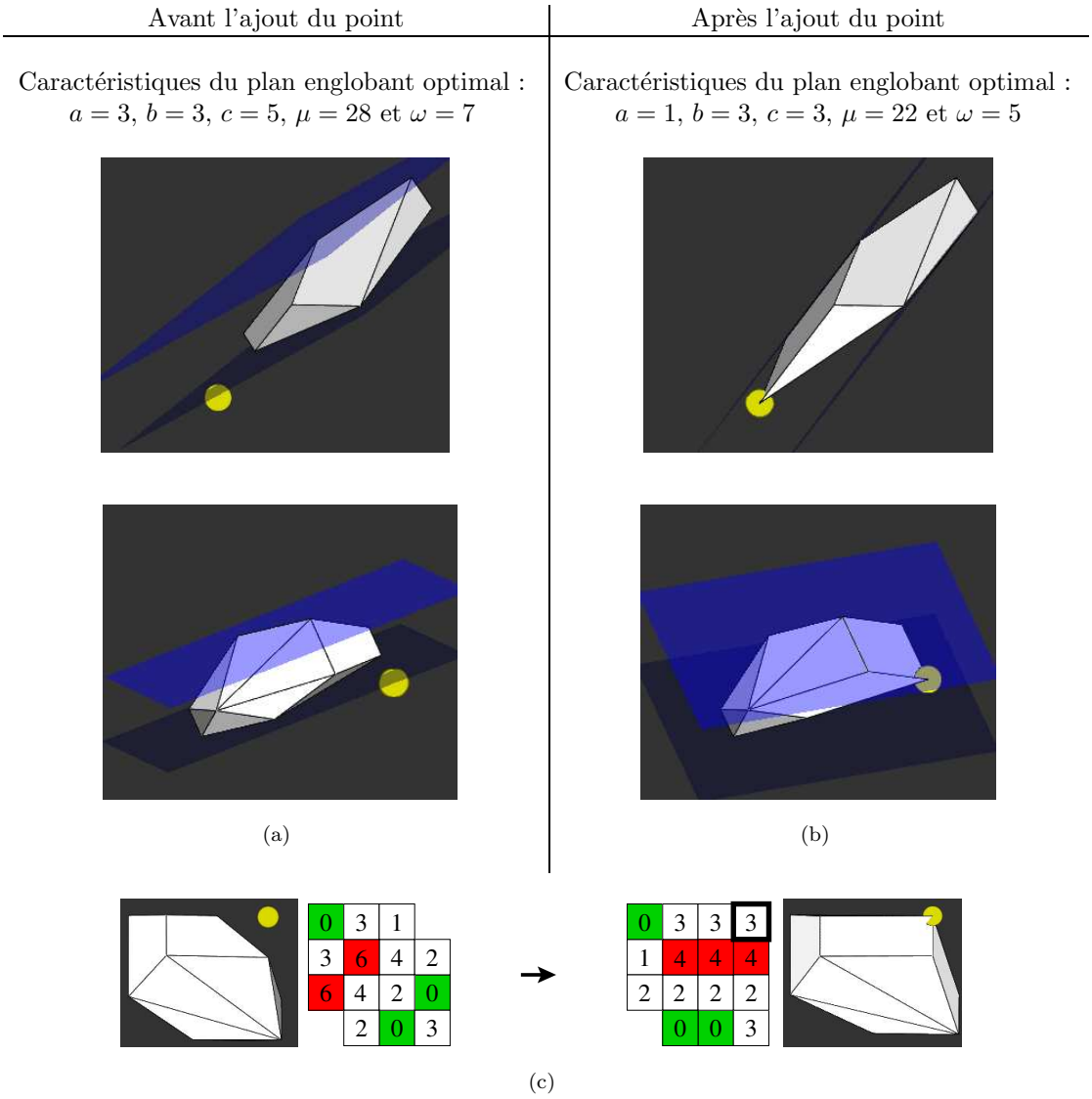



Figure 4.4 – Contre-exemple de l'observation 3. Le point ajouté est indiqué en jaune sur les vues 3D et entouré en gras sur les vues en projection. Les plans bleus sont les plans support dont la distance est minimale. (a) Deux différents points de vue de l'enveloppe convexe et des plans support avant l'ajout du point. (b) Les mêmes vues après l'ajout du point. (c) Vue par restes avant et après l'ajout du point. Les points d'appui inférieurs (resp. supérieurs) du plan englobant optimal sont représentés en vert (resp. rouge). Les deux étapes sont accompagnées de la vue 3D qui leur correspond.

**Analyse du contre-exemple :** Nous voyons bien sur les deux vues de la figure 4.4(a) que le point ajouté se situe sous le plan englobant optimal. Cela signifie que son reste n'est pas dans l'intervalle $[0, \omega - 1]$, et par conséquent le plan englobant optimal doit être recalculé. Nous effectuons alors une étape de la procédure incrémental de l'algorithme 3. Nous obtenons la nouvelle enveloppe convexe ainsi que les nouveaux plans support comme illustré sur les deux vues de la figure 4.4(b). D'après l'observation 3, le point ajouté devrait être un point d'appui du nouveau plan englobant optimal, or la vue du haut montre bien qu'il se situe strictement entre les deux plans support, ce qui contredit cette observation.

**Explication :** Pour comprendre pourquoi le point ajouté, $M$, n'était pas un point d'appui, nous avons calculé l'ensemble des paires antipodales contenant $M$ dans l'un des éléments de la paire. Chacune de ces paires définit deux plans support parallèles, et nous avons calculé la distance entre ces deux plans. Nous avons alors gardé la plus petite distance obtenue, à savoir $\frac{11}{\sqrt{142}} \approx 0.923$. Or cette distance est plus grande que celle obtenue en calculant la hauteur du nouveau plan englobant optimal, qui est $\frac{5-1}{\sqrt{1^2+3^2+3^2}} = \frac{4}{\sqrt{19}} \approx 0.918$. C'est ce qui explique que $M$ ne soit pas un point d'appui.

Or dans ce cas, cela signifie que la paire antipodale définissant les plans support dont la distance est minimale[14] était déjà présente dans l'ancienne enveloppe convexe.

En effet, supposons le contraire : la paire antipodale minimale n'était pas présente dans l'ancienne enveloppe convexe. Si c'est à présent la paire antipodale minimale de la nouvelle enveloppe convexe cela signifie qu'elle est formée d'au moins un élément (face, arête ou sommet) qui a été créé lors de l'ajout du point $M$. Or nous avons vu dans la partie 4.2.1 que les éléments créés sont soit les arêtes qui relient $M$ à l'horizon, soit les faces triangulaires formées par un segment de l'horizon et deux des arêtes précédemment mentionnées, soit le point $M$ lui-même. Dans tous les cas les éléments créés contiennent $M$. Par conséquent, la paire antipodale minimale contient $M$. On en déduit donc que $M$ appartient à un des plans support défini par cette paire antipodale, et donc a fortiori que $M$ est un point d'appui du nouveau plan englobant minimale. Or ceci est contredit par le contre-exemple présenté. □

Ainsi en conservant les anciennes paires antipodales qui sont toujours dans la nouvelle enveloppe convexe et en ajoutant celles qui sont créées au cours La réflexion précédente nous permet de déduire une procédure incrémentale pour calculer toutes les paires antipodales : si, à une étape $i$ de l'algorithme 3, toutes les paires antipodales de l'enveloppe convexe $C_i$ sont stockées dans un ensemble $E_i$, alors nous pouvons obtenir l'ensemble $E_{i+1}$ de toutes les paires antipodales de l'enveloppe convexe $C_{i+1}$ obtenues après l'ajout du point $M$ de la façon suivante :

- on supprime de $E_i$ l'ensemble des paires antipodales dont un des éléments à été supprimé par l'ajout de $M$,
- on ajoute à $E_i$ l'ensemble des paires antipodales obtenues en effectuant les rotations seulement autour des arêtes de $C_{i+1}$ qui sont incidentes à $M$, puisque c'est seulement dans les éléments incidents à $M$ qu'on trouve les nouvelles paires antipodales,
- on a alors $E_{i+1} \longleftarrow E_i$.

Cette procédure incrémentale est à la base de l'algorihme que je vais présenter dans le paragraphe suivant.

## 4.4 Algorithme incrémental retenu

En combinant la procédure précédente avec celle du calcul incrémental de l'enveloppe convexe, nous obtenons l'algorithme 4. L'idée est que si à chaque étape nous disposons de l'ensemble des paires antipodales, alors l'épaisseur est donnée par la paire définissant les plans support parallèles dont la distance est minimale. Néanmoins, pour pouvoir maintenir incrémentalement les paires antipodales, il nous faut une structure de données pouvant supporter les opérations suivantes :

- supprimer un paire antipodale,
- ajouter une nouvelle paire antipodale,
- rechecher une paire antipodale particulière,
- trouver la paire antipodale qui définit les plans support dont la distance est minimale.

Étant donné qu'il n'y a pas une opération dominante par rapport aux autres, nous voulons que chaque opération soit effectuée efficacement. C'est pourquoi nous avons choisi d'utiliser les arbres rouge et noir. Une étude détaillée de cette structure de donnée se trouve dans [CLRS02]. Je rappelle simplement que les arbres rouge et noir sont des arbres binaires de recherche ordonnés équilibrés. Les opérations `Recherher`,

[14] Pour clarifier la démonstation qui suit, j'appellerai cette paire antipodale, la *paire antipodale minimale*.

`Inserer`, `Supprimer` et `Minimum` se font toutes en temps $\mathcal{O}(\log n)$ où $n$ représente le nombre de noeuds dans l'arbre.

**Algorithme 4** : Reconnaissance incrémentale de morceaux de plans flous

**Entrées** : $\mathcal{E}_b$ un morceau de plan discret flou, $\nu$ un réel.
**Sorties** : $estMorceauPlanFlou$ un booléen, vrai si $\mathcal{E}_b$ est un morceau de plan discret flou d'épaisseur $\nu$, faux sinon. Dans le cas où $estMorceauPlanFlou$ est vrai, $a, b, c, \mu$ et $\omega$ représentent les caractéristiques du plan englobant optimal de $\mathcal{E}_b$.

**Variables** : $A$, un arbre rouge et noir permettant de stocker les paires antipodales

1 **début**
  *// Initialisation*
2  $estMorceauPlanFlou \longleftarrow$ vrai
3  $C \longleftarrow$ enveloppe convexe d'un tricube $t$ de départ choisi au hasard dans $\mathcal{E}_b$
4  **pour chaque** *paire antipodale $p$ de $C$* **faire**
5  `Inserer`($p$,$A$)

  *// Boucle principale*
6  **tant que** $estMorceauPlanFlou$ **et** $\mathcal{E}_b$ *n'a pas été totalement parcouru* **faire**
7  Mettre à jour $C$ en ajoutant le voxel $M$ en suivant un parcours pseudo-géodésique
8  Soit $S$ l'ensemble des éléments de $C$ supprimés par l'ajout de $M$
9  **pour chaque** *élement $e \in S$* **faire**
    *// Les paires antipodales dont un des élements a été détruit sont supprimées de $A$*
10  $p \longleftarrow$ `Rechercher`($e$,$A$)
11  **si** $(p \neq \emptyset)$ **alors**
12  `Supprimer`($p$,$A$)

    *// Nous ajoutons les nouvelles paires antipodales ; elle sont toutes issues de $M$*
13  **pour chaque** *arête $e$ de $C$ incidente à $M$* **faire**
14  $E_{pa} \longleftarrow$ `EffectuerRotation`($e$)
15  **pour chaque** *paire antipodale $p \in E_{pa}$* **faire**
16  **si** `TestValidité`($p$) $\leqslant \nu$ **alors** `Inserer`($p$,$A$)

17  **si** $A \neq \emptyset$ **alors**
18  $\langle a, b, c, d_1, d_2 \rangle \longleftarrow$ `Minimum`($A$)
19  $\mu \longleftarrow \min(-d_1, -d_2)$
20  $\omega \longleftarrow |d_2 - d_1| + 1$
21  **sinon**
22  $estMorceauPlanFlou \longleftarrow$ faux ;

23  **retourner** $estMorceauPlanFlou$
24 **fin**

La fonction `Rechercher`($e$,$A$) retourne la liste des paires antipodales contenant l'élément $e$, qui se trouvent dans l'arbre $A$. Si cette liste n'est pas vide alors `Supprimer`($p$,$A$) supprime toutes les paires antipodales $p \in A$. La fonction `TestValidité` va s'assurer que le vecteur normal des plans support appartient bien au 48$^{e}$ d'espace dans lequel nous nous trouvons. De plus, elle va vérifier que la distance entre les plans support parallèles définis par les paires antipodales renvoyées par la fonction `Effectuer-Rotation` est bien inférieure à $\nu$. Si les deux conditions sont respectées, alors la paire antipodale est utile et nous la stockons grâce à la fonction `Inserer`($p$,$A$) à la ligne 16. Si à la suite de toutes les insertions l'arbre $A$ est vide, cela signifie qu'il n'y a plus de paire antipodale dont la distance entre les plans support qu'elle définit est inférieure à $\nu$. Donc $\mathcal{E}_b$ n'est pas un morceau de plan flou d'épaisseur $\nu$. Dans le cas contraire, la fonction `Minimum`($A$) va retourner les caractéristiques des plans support parallèles définit par les paires antipodales qui appartiennent à $A$ dont la distance est minimale. Nous pouvons alors retrouver

les caractéristiques du plan englobant optimal de notre ensemble de points.

**Analyse de la complexité :** Nous allons effectuer une analyse dans le cas moyen. Pour un ensemble $\mathcal{E}_b$ contenant $n$ points, notons $I$ le nombre maximal de paires antipodales que nous pouvons obtenir au cours de l'algorithme.

Les étapes d'initialisation se font en temps constant car nous travaillons sur l'enveloppe convexe d'un tricube. La boucle principale à la ligne 6, va effectuer au plus $n$ itérations. Chaque itération commence par mettre à jour l'enveloppe convexe. En considérant que nous utilisons la version on-line de l'algorithme vu à la partie 4.2.1, nous savons que cette étape s'éffectue en temps $\mathcal{O}(n)$. La boucle de la ligne 9 va supprimer dans le pire cas les $I$ paires antipodales contenues dans $A$. Vu que $A$ est un arbre rouge et noir cette étape s'effectue en $\mathcal{O}(I \log I)$ au pire cas. Pour estimer le coût de la boucle de la ligne 13 nous aurons besoin du théorème suivant, que l'on trouve dans [dBvKOS00] :

**Théorème.** *Soit $P$ un polyèdre convexe avec $n$ sommets. Le nombre d'arêtes de $P$ est au plus $3n-6$ et le nombre de faces de $P$ est au plus $2n-4$.*

*Démonstration.* Nous partons de la formule d'Euler, qui pour un polyèdre à $n$ sommets, $n_a$ arêtes et $n_f$ faces est :

$$n - n_a + n_f = 2$$

Chaque face du polyèdre a au minimum 3 arêtes qui lui sont incidentes et chaque arête du polyèdre est incidente à deux faces, nous avons donc $3n_f \leqslant 2n_a$. En substituant $n_a$ par cette inégalité dans la formule d'Euler nous obtenons

$$3n_f/2 \leqslant n + n_f - 2$$

donc $n_f \leqslant 2n-4$. De la même manière, une substitution de $n_f$ dans la formule d'Euler nous permet d'obtenir que $n_a \leqslant 3n-6$. □

Soit $C$ l'enveloppe convexe à une étape $i$ de l'algorithme. Supposons que $C$ contienne $r$ sommets, alors d'après le théorème précédent $C$ contient au plus $3r-6$ arêtes. Chaque arête étant formée de deux sommets, la somme des degrés des sommets de $C$ est au plus $6r-12$. Ainsi, en moyenne, le degré d'un sommet est borné par $6-12/r<6$. Donc le degré moyen d'un sommet est au plus 6. Nous en déduisons qu'en moyenne, la boucle de la ligne 13 qui effectue la recherche des nouvelles paires antipodales, comporte au plus 6 itérations. A chaque itération, une rotation est effectuée et nous savons que cette opération se fait en $\mathcal{O}(n)$ (cf. partie 3.2). De plus les paires antipodales découvertes pendant la rotation sont stockées dans $A$ et au maximum il peut y en avoir $I$. Il en résulte que la complexité totale en moyenne de la boucle de la ligne 13 est de $\mathcal{O}(n + I \log I)$. Enfin on peut obtenir l'épaisseur minimale à la ligne 18 en temps $\mathcal{O}(\log I)$ d'après les propriétés de l'arbre rouge et noir.

Ce qui nous donne une complexité totale moyenne en $\mathcal{O}(n(n + I \log I + (n + I \log I) + \log I))$, c'est-à-dire $\mathcal{O}(n^2 + nI \log I)$.
D'un autre côté, puisqu'il faut stocker toutes les paires antipodales, le coût en espace de notre algorithme est de $\mathcal{O}(n+I)$.

Si $I$ est de l'ordre de $\Theta(n^2)$ alors la complexité moyenne est de $\mathcal{O}(n^3 \log n)$. Quand $I$ est de l'ordre de $\Theta(n)$, nous obtenons une complexité moyenne de $\mathcal{O}(n^2 \log n)$. Or c'est plutôt le deuxième cas qui arrive en pratique avec les morceaux de plans flous auxquels nous nous sommes restreint. Cela s'explique par le fait que les enveloppes convexes obtenues au cours de l'algorithme ne comportent pas baucoup d'arêtes par rapport au nombre de points $n$ de l'ensemble $\mathcal{E}_b$. Ainsi la fonction `EffectuerRotation` est plus rapide et la taille de l'arbre $A$ est plus petite.

# Conclusion et perspectives

Dans le domaine de la géométrie discrète, le problème de reconnaissance d'objets discrets a été étudié par de nombreux chercheurs [DRR96, VC00, Buz02, Gér03]. J'ai rapidement présenté dans ce mémoire quelques méthodes développées pour résoudre le problème en dimension 2 et 3. Nous avons alors vu que la grande majorité d'entre-elles n'étaient valables que pour les objets discrets présentant une régularité que nous trouvons rarement en pratique sur les données manipulées. En effet ces dernières sont issues de processus d'acquisition qui induisent du bruit dans les données. J'ai alors présenté un nouvelle classe d'objets discrets, les objets flous, destinés à prendre en compte ce bruit. Après avoir expliqué le problème en dimension 2, j'ai généralisé la méthode en proposant dans un premier temps une nouvelle définition, celle du morceau de plan discret flou, puis j'ai montré la relation qu'il existait avec le problème du calcul de l'épaisseur d'un ensemble de points en dimension 3. J'ai ensuite étudié et analysé les différentes approches adoptées pour résoudre ce problème.

Dans un second temps, j'ai proposé et programmé un algorithme naïf utilisant une de ces méthodes afin de simuler une reconnaissance incrémental de morceaux de plans flous. En me reposant sur les propriétés analytiques des plans discrets, j'ai mis en évidence plusieurs simplifications permettant de rendre l'algorithme plus rapide en pratique. Afin de pouvoir tester ces algorithmes j'ai aussi programmé un générateur de morceaux de plans flous ainsi qu'une interface de visualisation permettant d'observer l'évolution de l'épaisseur en fonction des points ajoutés. Les nombreux tests de reconnaissance de morceaux de plans discrets flous m'ont dans un premier temps laissé penser qu'un point ajouté au cours de l'algorithme qui modifiait les caractéristiques du plan englobant optimal devenait un point d'appui du nouveau plan englobant optimal. Mais j'ai trouvé un contre-exemple à cette observation. Une analyse de ce contre-exemple m'a permis de mettre en évidence un cas de figure auquel nous n'avions pas pensé. J'ai alors proposé à partir de cette analyse une méthode incrémentale pour construire l'ensemble des paires antipodales. L'algorithme étant plus général que les deux précédents la complexité théorique n'est pas optimale. Néanmoins les algorithmes se comportent bien en pratique. L'objectif du stage est donc rempli, mais laisse envisager de nombreuses perspectives de travaux.

Une étude du même problème, mais en considérant cette fois-ci la distance verticale entre les plans support, peut être envisagée. Nous pensons que cela pourra permettre quelques optimisations des algorithmes présentés dans ce mémoire, notamment au niveau de la recherche de la paire antipodale définissant les plans support dont la distance est minimale. Nous espérons ainsi mettre au point des algorithmes performants combinant les approches de géométrie discrète et de géométrie algorithmique.

Ces travaux pourrons alors être exploités pour obtenir une polyédrisation d'objets discrets bruités tridimensionnels. Ce travail permettra de représenter les surfaces discrètes bruitées avec un nombre minimal d'approximations de facettes planes discrètes, tout en contrôlant l'approximation effectuée. Cette représentation paramétrique pourra être utilisée pour extraire rapidement des propriétés géométriques des surfaces discrètes ou encore pour lisser « sous contrôle » la surface des objets discrets bruités et ensuite la transformer en surface euclidienne permettant ainsi l'utilisation de calculs rapides de la géométrie euclidienne sur celle-ci (transformations géométriques, tests de visibilité, d'intersections, ...).

# Bibliographie


[AAS94] P.K. Agarwal, B. Aronov, and M. Sharir. Computing envelopes in four dimensions with applications. In *SCG'94 : Proceedings of the tenth annual symposium on Computational geometry*, pages 348–358, Stony Brook, New York, United States, 1994.

[And93] E. Andrès. Le plan discret. In *Discrete Geometry for Computer Imagery*, Strasbourg, September 1993.

[AS95] P.K. Agarwal and M. Sharir. Efficient randomized algorithms for some geometric optimization problems. In *SCG'95 : Proceedings of the eleventh annual symposium on Computational geometry*, pages 326–335, Vancouver, British Columbia, Canada, 1995.

[BO79] J. Bentley and T. Ottmann. Algorithms for reporting and counting geometric intersections. In *IEEE Transaction on Computers C 28*, pages 643–647, 1979.

[Bro79] K.Q. Brown. *Geometric transforms for fast geometric algorithms*. Ph.D. thesis, Dept. Comput. Sci., Carnegie-Mellon Univ., Pittsburgh, PA, 1979.

[Buz02] L. Buzer. An incremental linear time algorithm for digital line and plane recognition using a linear incremental feasibility problem. In *DGCI'02 : Proceedings of the 10th International Conference on Discrete Geometry for Computer Imagery*, volume 2301, pages 372–381, Bordeaux, France, 2002.

[CEGS92] B. Chazelle, H. Edelsbrunner, L. Guibas, and M. Sharir. Diameter, width, closest line pair, and parametric searching. In *SCG'92 : Proceedings of the eighth annual symposium on Computational geometry*, pages 120–129, Berlin, Germany, 1992.

[CLRS02] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein. *Introduction à l'algorithmique*. Dunod, 2nd edition, 2002.

[Coe02] D. Coeurjolly. *Algorithmique et géométrie discrète pour la caractérisation des courbes et des surfaces*. Thèse, Université Lumière Lyon 2, December 2002. `http://liris.cnrs.fr/~dcoeurjo/these.html`

[dBvKOS00] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry : Algorithms and Applications*. Springer-Verlag, Heidelberg, 2nd edition, 2000.

[DR95] I. Debled-Rennesson. *Étude et reconnaissance des droites et plans discrets*. Thèse, Université Louis Pasteur, Strasbourg, December 1995. `http://www.loria.fr/~debled/TheseDebledRennesson.pdf`

[DRFRD05] I. Debled-Rennesson, F. Feschet, and J. Rouyer-Degli. Optimal blurred segments decomposition in linear time. In *DGCI'05 : Proceedings of the 12th International Conference on Discrete Geometry for Computer Imagery*, volume 3429 of *Lecture Notes in Computer Science*, pages 371–382, Poitiers, France, April 2005.

[DRR96] I. Debled-Rennesson and J.P. Reveillès. Incremental algorithm for recognizing pieces of digital planes. In *Spie's International Symposium on Optical Science, Engineering, and Instrumentation*, volume 2826 of *Vision Geometry V*, page 140, Denver, USA, August 1996.

[DRRRD03] I. Debled-Rennesson, J.-L. Rémy, and J. Rouyer-Degli. Segmentation of discrete curves into fuzzy segments. In *IWCIA'03 : Proceedings of the 9th International Workshop on Combinatorial Image Analysis*, volume 12 of *Electronic Notes in Discrete Mathematics*, Palerme, Italie, May 2003.

[FP99] J. Françon and L. Papier. Polyhedrization of the boundary of a voxel object. In *DGCI'99 : Proceedings of the 8th International Conference on Discrete Geometry for Computer Imagery*, volume 1568 of *Lecture Notes in Computer Science*, pages 425–434, Marne-la-Vallée, France, March 1999.

[Gér03] Y. Gérard. A fast and elementary algorithm for digital plane recognition. In *IWCIA'03 : Proceedings of 9th International Workshop on Combinatorial Image Analysis*, volume 12 of *Electronic Notes in Discrete Mathematics*, pages 1–7, Palerme, Italie, May 2003.

[GH01] B. Gärtner and T. Herrmann. Computing the width of a point set in 3-space. *J. Exp. Algorithmics*, 4 :3, 2001.

[Gra92] D.J. Grabiner. Farey nets and multidimensional continued fractions. *Monatsh. Math.*, 114 :35–60, 1992.

[GS86] L. Guibas and R. Seidel. Computing convolutions by reciprocal search. In *SCG'86 : Proceedings of the second annual symposium on Computational geometry*, pages 90–99, Yorktown Heights, New York, United States, 1986.

[HT88] M.E. Houle and G.T. Toussaint. Computing the width of a set. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 10(5) :761–765, 1988.

[Meg83] N. Megiddo. Applying parallel computation algorithms in the design of serial algorithms. *Journal of the ACM*, 30(4) :852–865, 1983.

[Meg84] N. Megiddo. Linear programming in linear time when the dimension is fixed. *Journal of the ACM*, 31(1) :114–127, 1984.

[Mel87] A. Melkman. On-line construction of the convex hull of a simple polygon. *Information Processing Letters*, 25 :11–12, 1987.

[Pap99] L. Papier. *Polyédrisation et visualisation d'objets discrets tridimensionnels.* Thèse, Université Louis Pasteur, Strasbourg, 1999.

[PH77] F. P. Preparata and S. J. Hong. Convex hulls of finite sets of points in two and three dimensions. *Communications of the ACM*, 20(2) :87–93, 1977.

[Rev91] J.P. Reveillès. *Géométrie discrète, calculs en nombre entiers et algorithmique.* Thèse d'état, Université Louis Pasteur, Strasbourg, 1991.

[Siv04] I. Sivignon. *De la caractérisation des primitives à la reconstruction polyédrique de surfaces en géométrie discrète.* Thèse, Institut National Polytechnique de Grenoble, 2004. `http://www.lis.inpg.fr/pages_perso/sivignon/these.html.fr`

[SSMJ99] J. Schwerdt, M. Smid, J. Majhi, and R. Janardan. Computing the width of a three-dimensional point set : an experimental study. *Journal of Experimental Algorithmics*, 4 :8, 1999.

[Tou83] G.T. Toussaint. Solving geometric problems with the rotating calipers. In *Proceedings of IEEE MELECON'83*, Athens, Greece, May 1983.

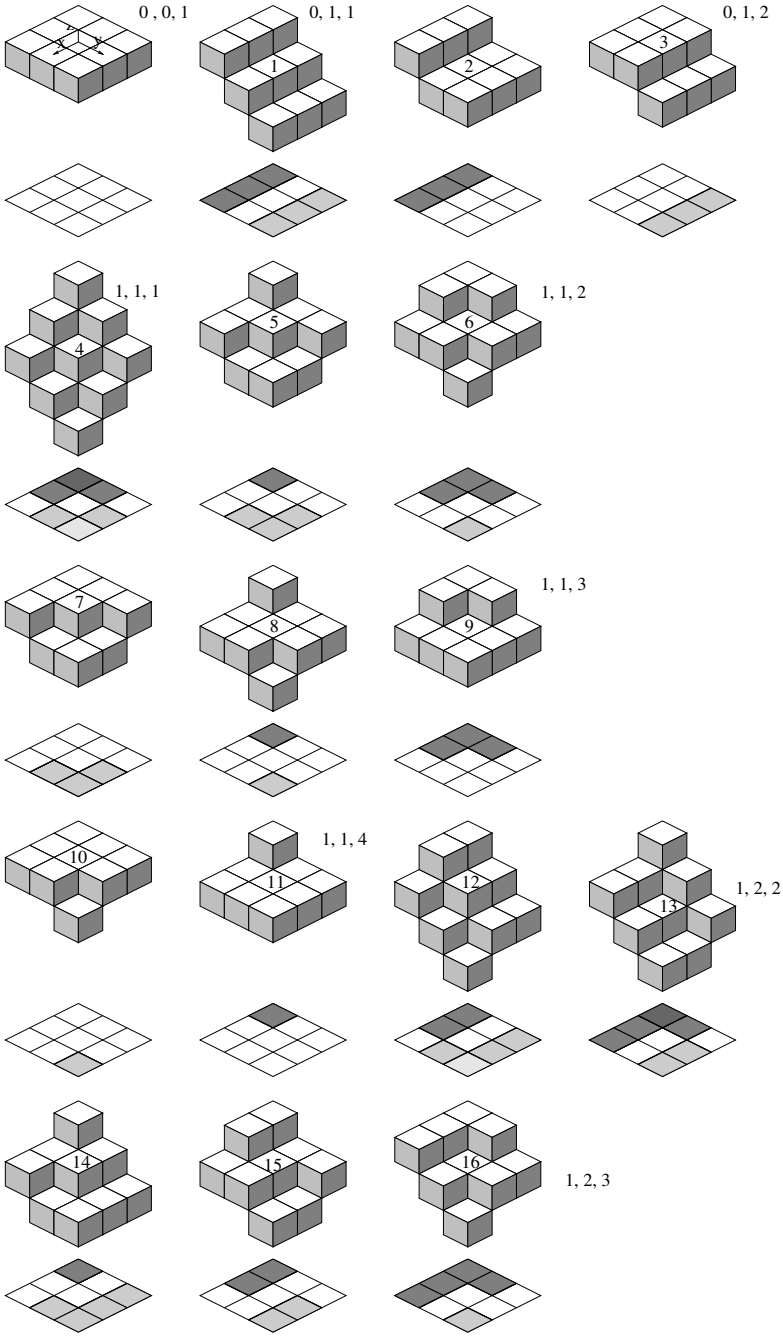
[VC00] J. Vittone and J.M. Chassery. Recognition of digital naive planes and polyhedrization. In *DGCI'00 : Proceedings of the 9th International Conference on Discrete Geometry for Computer Imagery*, volume 1953 of *Lecture Notes in Computer Science*, pages 296–307, Uppsala, Sweden, December 2000.

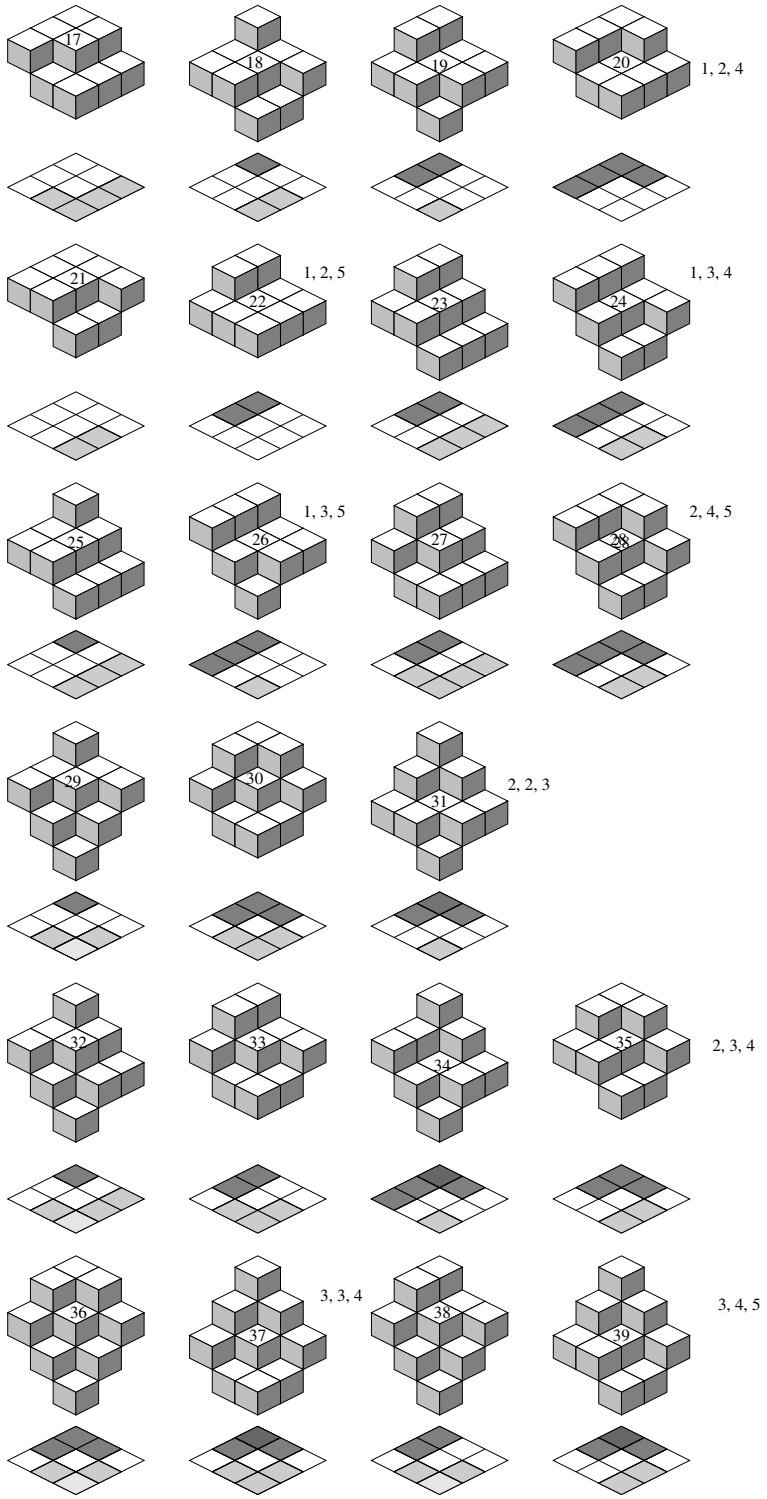
[Vit99] J. Vittone. *Caractérisation et reconnaissance de droites et de plans en géométrie discrète.* Thèse, Université Joseph Fourier, Grenoble, 1999. `http://www.lis.inpg.fr/stages_dea_theses/theses/these/Th_Vittone.html`

[VJ02] D. Vandevoorde and N.M. Josuttis. *C++ Templates : The Complete Guide.* Addison-Wesley, 2002.

# Annexe A

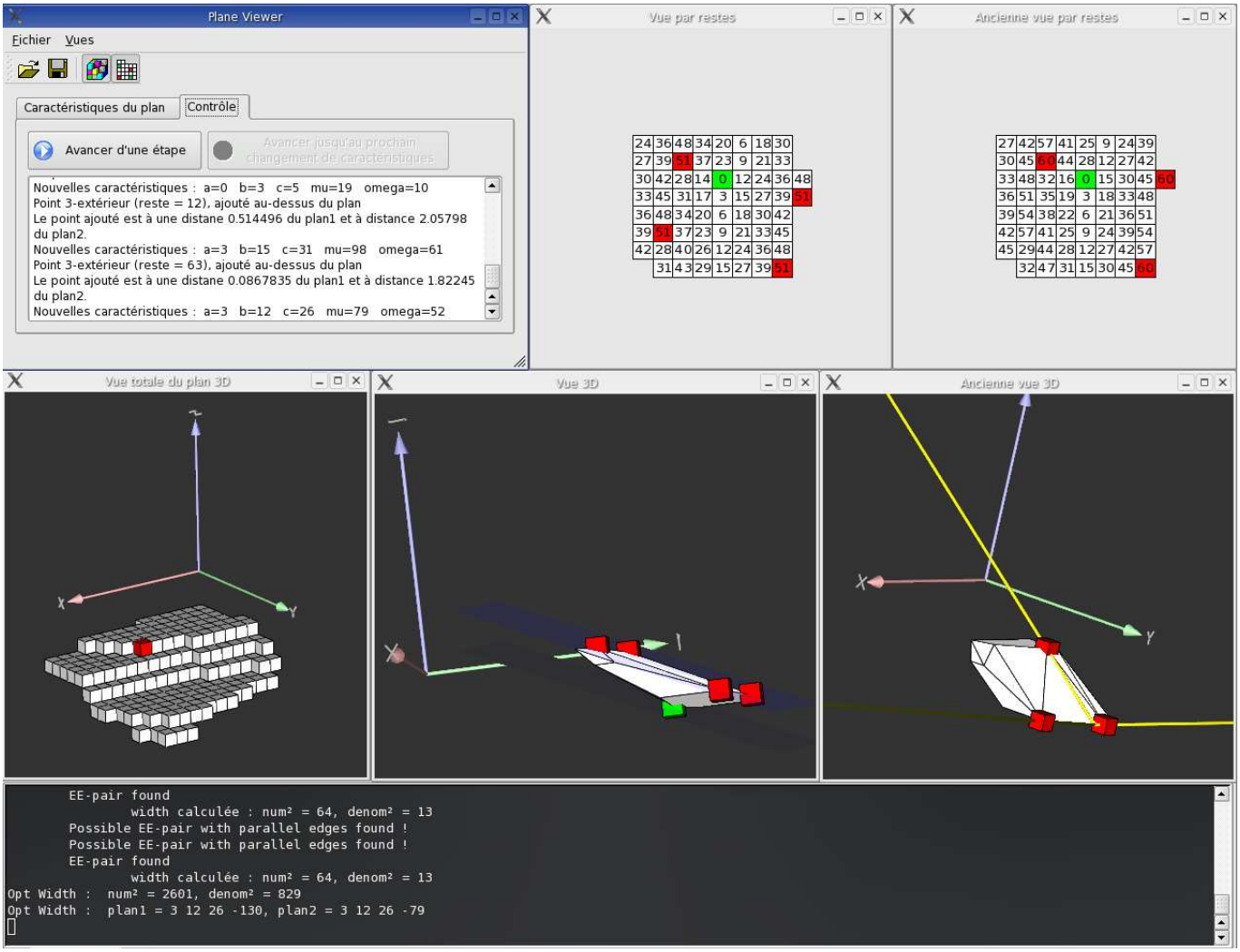
# Liste des tricubes du premier 48$^{e}$ d'espace


0 , 0, 1
0, 1, 1
0, 1, 2
1, 1, 1
1, 1, 2
1, 1, 3
1, 1, 4
1, 2, 2
1, 2, 3
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16


17
18
19
20
1, 2, 4
21
1, 2, 5
22
23
24
1, 3, 4
25
1, 3, 5
26
27
28
2, 4, 5
29
30
31
2, 2, 3
32
33
34
35
2, 3, 4
36
3, 3, 4
37
38
39
3, 4, 5

# Annexe B

# Interface de visualisation 3D


Plane Viewer
Fichier Vues
Caractéristiques du plan
Contrôle
Avancer d'une étape
Avancer jusqu'au prochain changement de caractéristiques
Nouvelles caractéristiques : a=0 b=3 c=5 mu=19 omega=10
Point 3-extérieur (reste = 12), ajouté au-dessus du plan
Le point ajouté est à une distane 0.514496 du plan1 et à distance 2.05798 du plan2.
Nouvelles caractéristiques : a=3 b=15 c=31 mu=98 omega=61
Point 3-extérieur (reste = 63), ajouté au-dessus du plan
Le point ajouté est à une distane 0.0867835 du plan1 et à distance 1.82245 du plan2.
Nouvelles caractéristiques : a=3 b=12 c=26 mu=79 omega=52
Vue par restes
Ancienne vue par restes
Vue totale du plan 3D
Vue 3D
Ancienne vue 3D
EE-pair found
width calculée : num² = 64, denom² = 13
Possible EE-pair with parallel edges found !
Possible EE-pair with parallel edges found !
EE-pair found
width calculée : num² = 64, denom² = 13
Opt Width : num² = 2601, denom² = 829
Opt Width : plan1 = 3 12 26 -130, plan2 = 3 12 26 -79