

## *Etude et reconnaissance des droites et plans discrets*

### RESUME

Dans de nombreux domaines, en particulier en infographie, imagerie médicale et traitement d'images, les données manipulées sont discrètes et en général, les théorèmes euclidiens ne se transfèrent pas dans un espace discret. C'est pourquoi l'élaboration d'une géométrie discrète théorique s'avère nécessaire en définissant les objets géométriques non pas algorithmiquement mais par des définitions arithmétiques rigoureuses. Grâce à celles-ci, une algorithmique graphique sans faille peut être établie en dimension 2 comme en dimension 3.

Cette thèse s'inscrit dans ce cadre et, à partir des définitions arithmétiques des droites et plans discrets, nous développons l'étude de ces objets discrets et nous établissons des algorithmes de construction et de reconnaissance performants. Ces algorithmes s'appliquent à la segmentation de courbes discrètes 2D et 3D ainsi qu'à la facettisation d'objets voxels.

### Mots-clés

Droite discrète, courbe discrète, inégalité diophantienne, plan discret, surface discrète, reconnaissance, segmentation, polygonalisation, facettisation.

## *Study and recognition of digital straight lines and planes*

### ABSTRACT

Usual euclidian theorems are not satisfactory when applied to the discrete data used in a large number of domains (computer graphics, medical imagery, image processing). Hence it becomes necessary to define a theoretical digital geometry based on a consistent arithmetical definition of geometrical objects which allows to build an algorithmics without any failure for computer graphics in dimension 2 as well as in dimension 3.

In this framework, this thesis describes the study of digital objects developed from the arithmetical definitions of digital straight lines and planes and leads to efficient construction and recognition algorithms. These algorithms are applied to 2D and 3D digital curves segmentation and to the facettisation of voxel objects.

### Keywords

Digital line, digital curve, diophantine inequality, digital plane, digital surface, recognition, segmentation, polygonalization, facettization.

## THESE

Présentée le 15 Décembre 1995 en vue de l'obtention du  
Titre de DOCTEUR de l'UNIVERSITE LOUIS PASTEUR  
(Mention Sciences)  
Spécialité : Informatique

---

*Etude et reconnaissance des droites  
et plans discrets*

---

**Isabelle DEBLED-RENNESON**

M. J. FRANÇON, président du jury et rapporteur interne,  
Professeur à l'Université Louis Pasteur, Strasbourg  
M. J. BERSTEL, examinateur,  
Professeur à l'Université Paris VI, Paris  
M. J.M. CHASSERY, rapporteur externe,  
Directeur de Recherche au CNRS, Grenoble  
M. R.A. MELTER, rapporteur externe,  
Professeur à l'Université de Long Island, New-York  
M. J.P. REVEILLÈS, directeur de thèse,  
Professeur à l'Université d'Auvergne, Clermont-Ferrand



*A Jean-loup et Matthieu*



Je tiens à remercier :

Monsieur Jean-Pierre Reveillès, Professeur à l'Université d'Auvergne de Clermont-Ferrand qui a accepté d'être mon directeur de thèse, m'a communiqué sa passion pour la géométrie discrète, m'a conseillée et encouragée pendant ces trois années de travail.

Monsieur Jean Françon, Professeur à l'Université Louis Pasteur de Strasbourg pour ses nombreux conseils et pour avoir accepté de rapporter ce travail.

Messieurs Jean-Marc Chassery, Directeur de recherche au CNRS de Grenoble, Robert Melter, Professeur à l'Université de Long Island et Jean Berstel, Professeur à l'Université Paris VI, pour l'intérêt qu'ils ont porté à ce travail, et pour avoir accepté de le juger.

Monsieur Raymond Seroul pour son enthousiasme et l'intérêt qu'il a manifestés pour ce travail et en particulier pour ses commentaires sur les droites naïves dont il nous a fait bénéficier.

Toute les membres de l'équipe de géométrie discrète pour leur amitié et leurs nombreux conseils.

Ma famille et mes amis pour leur patience, leur soutien et pour certains leur aide dans la relecture de ce travail.



# Table des matières

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Historique du problème de la reconnaissance des droites discrètes 2D</b>	<b>7</b>
2.1	Introduction . . . . .	8
2.2	Algorithmes de reconnaissance utilisant les trois critères de Freeman . . . . .	12
2.2.1	L'algorithme de Wu . . . . .	12
2.2.2	Un problème soulevé par Hung . . . . .	15
2.2.3	Application à la polygonalisation de courbes discrètes . . . . .	16
2.3	Interprétation géométrique de l'algorithme d'Euclide et application à la reconnaissance de droites discrètes . . . . .	22
2.3.1	Les droites discrètes . . . . .	23
2.3.2	Le code fonctionnel . . . . .	23
2.3.3	Transformations géométriques et code . . . . .	24
2.3.4	Interprétation géométrique de l'algorithme d'Euclide . . . . .	25
2.3.5	Algorithme de reconnaissance de segments discrets avec calcul des caractéristiques . . . . .	27
2.3.6	Application à la polygonalisation de courbes discrètes . . . . .	32
2.4	Conclusion . . . . .	34
<b>3</b>	<b>Un algorithme linéaire de polygonalisation de courbes discrètes planes</b>	<b>37</b>
3.1	Introduction . . . . .	38
3.2	Un algorithme incrémental de reconnaissance de segments de droites discrètes naïves . . . . .	39
3.2.1	Points d'appui et droites d'appui . . . . .	39
3.2.2	Segment reconnu . . . . .	41
3.2.3	Point faiblement et fortement extérieur . . . . .	42
3.2.4	Croissance d'un segment reconnu . . . . .	43
3.2.5	L'algorithme incrémental . . . . .	47



3.2.6	Un exemple détaillé de déroulement de l'algorithme de reconnaissance de segments de droites discrètes . . . . .	49
3.3	La polygonalisation de courbes discrètes . . . . .	51
3.3.1	L'algorithme linéaire de polygonalisation de courbes discrètes 8-connexes dans tous les octants . . . . .	51
3.3.2	Polygonalisation de cercles discrets . . . . .	57
3.4	La reconnaissance des segments de droites et les arbres de Stern-Brocot	59
3.4.1	Définition et propriétés de l'arbre de Stern-Brocot . . . . .	60
3.4.2	Lien avec l'algorithme de reconnaissance de segments de droites discrètes . . . . .	61
3.5	Conclusion . . . . .	66
<b>4</b>	<b>Reconnaissance des droites connexes d'épaisseurs quelconques</b>	<b>69</b>
4.1	Introduction . . . . .	70
4.2	Reconnaissance des segments de droites 4-connexes et segmentation de courbes 4-connexes . . . . .	71
4.3	Reconnaissance de segments de droites connexes non naïves d'épaisseurs quelconques . . . . .	76
4.3.1	Décomposition d'une droite connexe non naïve . . . . .	76
4.3.2	Utilisation de l'algorithme de reconnaissance de segments de droites naïves pour la reconnaissance des segments de droites connexes non naïves . . . . .	77
4.3.3	Un algorithme de reconnaissance de segments de droites connexes non naïves d'épaisseurs quelconques . . . . .	79
4.4	Conclusion . . . . .	85
<b>5</b>	<b>Les droites discrètes 3D et la segmentation de courbes discrètes 26-connexes</b>	<b>87</b>
5.1	Introduction . . . . .	88
5.2	Définition et premières propriétés . . . . .	88
5.2.1	La droite discrète 3D . . . . .	88
5.2.2	Relation entre les épaisseurs arithmétiques et la connexité de la droite discrète 3D . . . . .	90
5.2.3	Un algorithme de parcours de droites discrètes naïves 3D . . . . .	93
5.3	Algorithme de reconnaissance de segments de droites naïves 3D . . . . .	94
5.4	Segmentation de courbes discrètes 26-connexes . . . . .	95
5.5	Conclusion . . . . .	99
<b>6</b>	<b>Généralités sur les plans discrets naïfs</b>	<b>101</b>
6.1	Introduction . . . . .	102
6.2	Définitions et premières propriétés . . . . .	102

6.2.1	Le plan discret naïf . . . . .	102
6.2.2	Symétries . . . . .	105
6.2.3	Représentations d'un plan discret naïf par lignes de niveaux . . . . .	106
6.2.4	Plans discrets équivalents . . . . .	107
6.3	Sections par les plans de coordonnées et conséquences . . . . .	108
6.3.1	Présence de droites discrètes 2D dans un plan naïf . . . . .	108
6.3.2	Paliers d'un plan naïf . . . . .	110
6.3.3	Algorithmes de parcours . . . . .	111
6.3.4	Intersection de deux plans naïfs . . . . .	117
6.4	Rôle de la suite des restes . . . . .	119
6.4.1	"Erreur" commise . . . . .	119
6.4.2	Représentation par restes . . . . .	120
6.4.3	Points d'appui et plans d'appui . . . . .	121
6.4.4	Périodicité . . . . .	123
6.4.5	Un algorithme de construction des faces visibles d'un plan naïf . . . . .	127
6.5	Conclusion . . . . .	130
<b>7</b>	<b>Historique du problème et premier algorithme arithmétique de reconnaissance de morceaux de plans discrets</b> . . . . .	<b>131</b>
7.1	Introduction . . . . .	132
7.2	Algorithmes reposant sur la construction de l'enveloppe convexe 3D d'un ensemble de points entiers (Kim, Stojmenović et Tošić) . . . . .	133
7.3	Le critère de régularité de P. Veelaert . . . . .	137
7.4	Une première approche arithmétique . . . . .	139
7.4.1	Utilisation de l'algorithme de Kim . . . . .	139
7.4.2	Un exemple de cas négligé par Kim . . . . .	140
7.4.3	Reformulation du théorème de Kim et premier algorithme arithmétique de reconnaissance de morceaux de plans discrets . . . . .	141
7.5	Conclusion . . . . .	145
<b>8</b>	<b>Reconnaissance incrémentale des plans discrets naïfs</b> . . . . .	<b>146</b>
8.1	Introduction . . . . .	147
8.2	Notions fondamentales pour la reconnaissance de morceaux de plans discrets . . . . .	148
8.2.1	Morceau reconnu d'un plan discret . . . . .	148
8.2.2	Polygones d'appui . . . . .	150
8.2.3	Point k-extérieur à un morceau reconnu de plan discret . . . . .	151
8.3	Résultats théoriques . . . . .	156
8.3.1	Le problème incrémental . . . . .	156
8.3.2	Ajout d'un point 1-extérieur à un morceau reconnu $S$ du plan $P$ . . . . .	157

8.3.3	Ajout d'un point fortement extérieur à un morceau reconnu $S$ du plan $P$ . . . . .	164
8.4	L'algorithme incrémental de reconnaissance de morceaux de plans discrets naïfs . . . . .	175
8.4.1	Initialisation des caractéristiques . . . . .	175
8.4.2	L'algorithme simplifié . . . . .	176
8.4.3	L'algorithme complet . . . . .	179
8.4.4	Un exemple de déroulement de l'algorithme de reconnaissance de morceaux de plans discrets . . . . .	181
8.5	Conclusion . . . . .	185
<b>9</b>	<b>Applications de l'algorithme de reconnaissance de morceaux de plans discrets</b> . . . . .	<b>188</b>
9.1	Introduction . . . . .	189
9.2	Les tricubes des plans discrets naïfs . . . . .	189
9.2.1	Définition et dénombrement des tricubes . . . . .	189
9.2.2	Application à la localisation des points réguliers d'une image .	194
9.3	Polyédrisation d'objets voxels symétriques . . . . .	197
9.3.1	Principe utilisé . . . . .	197
9.3.2	Exemples de polyédrisations d'objets voxels symétriques . . .	198
9.3.3	Nombreux problèmes restant à résoudre . . . . .	200
9.4	Conclusion . . . . .	200
<b>10</b>	<b>Conclusion</b> . . . . .	<b>202</b>

# Table des algorithmes

Algorithme de reconnaissance de segments de droites de Li de Wu. . . . .	14
Algorithme d'Euclide modifié. . . . .	26
Algorithme de reconnaissance de segments de droites naïves d'A.Troesch. . . . .	29
Algorithme de polygonalisation issu de l'algorithme de reconnaissance d'A.Troesch. . .	33
Algorithme incrémental de reconnaissance de segments de droites naïves du premier octant. . . . .	48
Algorithme linéaire de segmentation de courbes discrètes 8-connexes. . . . .	53
Algorithme incrémental de reconnaissance de segments de droites discrètes 4-connexes du septième octant. . . . .	72
Algorithme de reconnaissance de segments de droites discrètes connexes d'épaisseurs quelconques du septième octant. . . . .	80
Algorithme de construction d'un segment de droite discrète naïve 3D entre deux points A et B. . . . .	93
Algorithme de reconnaissance de segments de droites discrètes naïves 3D . . . . .	94
Algorithme de segmentation de courbes discrètes 26-connexes . . . . .	95
Algorithme de construction d'un plan discret naïf par "sections en y" . . . . .	112
Algorithme de construction d'un plan discret naïf par paliers de "sections en y" . . . . .	113
Algorithme de construction d'un plan discret naïf par "assemblage" des structures des "sections en y" . . . . .	117
Algorithme de construction des faces visibles d'un plan discret naïf . . . . .	129
Algorithme <i>CONVEX_PLANES(S)</i> (C.E. Kim) . . . . .	134
Algorithme <i>Digital_Hyperplane_Segments(S)</i> (I. Stojmenović et R. Tosić) . . . . .	136
Algorithme <i>Separability_ch(S, S')</i> (I. Stojmenović et R. Tosić) . . . . .	136
Algorithme <i>Eveness(S)</i> (P. Veelaert) . . . . .	139
Algorithme <i>Rec_Morc_Plan(S)</i> . . . . .	142
Algorithme incrémental de reconnaissance de morceaux de plans discrets naïfs	
<i>Initialisations</i> . . . . .	176
<i>Algorithme simplifié</i> . . . . .	178
<i>Algorithme complet</i> . . . . .	181



# Chapitre 1

## Introduction

Dans de nombreux domaines, en particulier en infographie, imagerie médicale et traitement d'images, les données manipulées sont discrètes et les théorèmes euclidiens se révèlent insatisfaisants ; en général, ils ne se transfèrent pas dans un espace discret. C'est pourquoi l'élaboration d'une géométrie discrète théorique s'avère nécessaire en définissant les objets géométriques non pas algorithmiquement mais par des définitions arithmétiques rigoureuses. Grâce à celles-ci, une algorithmique graphique sans faille peut être établie en dimension 2 comme en dimension 3.

Cette thèse s'inscrit dans ce cadre et, à partir des définitions arithmétiques des droites et plans discrets, nous développons l'étude de ces objets discrets pour établir des algorithmes de reconnaissance performants. Ces algorithmes s'appliquent à la segmentation de courbes discrètes 2D et 3D ainsi qu'à la facettisation d'objets voxels.

## Le problème de la reconnaissance des droites discrètes 2D

La reconnaissance des segments de droites a intéressé de nombreux auteurs pendant les vingt dernières années ; ce problème intervient dans beaucoup de domaines comme le traitement d'image, la reconnaissance de formes, l'algorithmique graphique, la programmation en nombres entiers ... Un bref historique non exhaustif des travaux effectués depuis 1970 est présenté dans le second chapitre.

Quatre conditions équivalentes caractérisent un segment de droite discrète :

- *Les trois propriétés de Freeman (70).*
- *La propriété de la corde de Rosenfeld (74).*
- *L'absence de segments irréguliers de Hung et Kasvand (84).*
- *La double inégalité diophantienne de Reveillès (89).*

Plusieurs algorithmes de reconnaissance de segments de droites discrètes utilisent les trois premières caractérisations. Nous décrivons l'un d'entre eux ; l'algorithme de Wu [42] datant de 1984 et les améliorations qui lui ont été apportées mais aucun des algorithmes reposant sur ces trois caractérisations n'est satisfaisant pour segmenter une courbe discrète car le contrôle de la structure de la droite repose sur des méthodes non incrémentales et trop compliquées. A l'encontre des algorithmes vus précédemment, celui d'A. Troesch [61] repose sur des énoncés mathématiques et utilise la définition arithmétique des droites discrètes donnée par J.P. Reveillès dans sa thèse d'état en 1989 [50], cependant, cet algorithme possède une complexité trop importante. Une droite discrète arithmétique de pente  $\frac{a}{b}$ , de borne inférieure  $\mu$  et d'épaisseur  $\omega$ , avec  $a$ ,  $b$ ,  $\mu$  et  $\omega$  dans  $\mathbb{Z}$ , est composée des points entiers de coordonnées  $(x, y)$  vérifiant la double inégalité diophantienne

$$\mu \leq ax - by < \mu + \omega \quad (1)$$

si  $\omega = \sup(|a|, |b|)$ , les droites discrètes sont appelées **droites discrètes naïves**, elles sont 8-connexes. Cette définition nous a permis d'élaborer l'algorithme très

performant exposé dans le chapitre suivant.

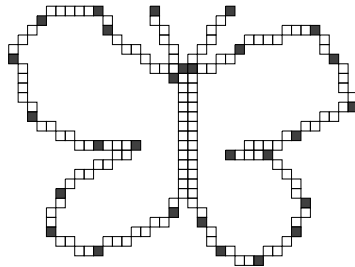
## Un algorithme linéaire de polygonalisation de courbes discrètes

Le troisième chapitre présente tout d'abord un algorithme incrémental de reconnaissance de segments de droites discrètes naïves du premier octant reposant sur le théorème fondamental suivant :

*Considérons un segment  $\Sigma$  de longueur  $l$  de la droite discrète naïve  $\mathcal{D}$  de pente  $\frac{a}{b}$  et de borne inférieure  $\mu$ . Le point  $M$  de coordonnées  $(x, y)$  avec  $x = l + 1$  est ajouté à  $\Sigma$ ,*

- (i) si  $\mu \leq ax - by < \mu + b$  alors  $\Sigma \cup \{M\}$  est un segment de la droite discrète  $\mathcal{D}$ ,*
- (ii) si  $ax - by = \mu - 1$  ou  $ax - by = \mu + b$  alors  $\Sigma \cup \{M\}$  est un segment de la droite discrète  $\mathcal{D}'$  dont la pente  $\frac{a'}{b'}$  et la borne inférieure  $\mu'$  se calculent en fonction de  $a, b, \mu$  et du point  $M$  ajouté,*
- (iii) si  $ax - by < \mu - 1$  ou  $ax - by > \mu + b$  alors  $\Sigma \cup \{M\}$  n'est pas un segment de droite discrète.*

Un algorithme linéaire de polygonalisation de courbes discrètes se déduit des symétries des droites discrètes et de l'algorithme incrémental de reconnaissance de segments de droites discrètes. Des exemples de segmentation d'ellipses, d'hyperboles, de cercles discrets et autres courbes discrètes sont donnés.



Enfin, le rapport étroit entre l'algorithme de reconnaissance et l'arbre de Stern-Brocot est mis en évidence.

L'algorithme de reconnaissance de segments de droites discrètes peut être étendu aux droites épaisses et aux droites naïves 3D ; ces applications sont exposées dans les deux chapitres suivants.

- **La reconnaissance des droites connexes d'épaisseurs quelconques**

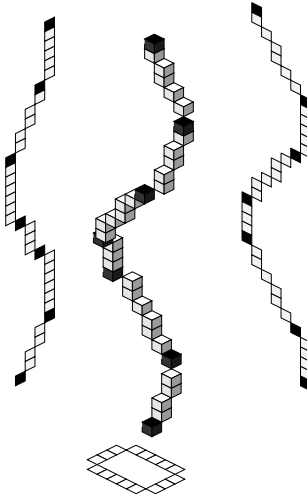
Les *droites connexes d'épaisseurs quelconques* sont les droites discrètes arithmétiques définies par la double inéquation diophantienne (1) dont l'épaisseur arithmétique  $\omega$  vérifie  $\omega > \sup(|a|, |b|)$ . L'algorithme de reconnaissance



des segments de droites naïves est adapté aux droites d'épaisseurs quelconques dans le chapitre 4 en décomposant celles-ci en plusieurs sous-structures.

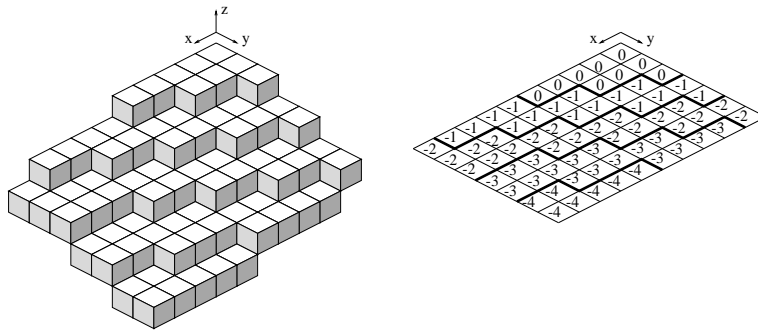
- **Les droites discrètes 3D**

Dans le cinquième chapitre, une nouvelle définition de droite discrète 3D est présentée puis un théorème de structure de ces droites et un algorithme de parcours sont proposés. Une *droite naïve 3D* se projetant bijectivement sur deux plans de coordonnées en des droites discrètes 2D, un algorithme de reconnaissance de segments de droites discrètes naïves 3D puis de segmentation de courbes discrètes 3D se déduit très simplement.



## Les plans discrets

Après avoir rappelé la définition d'un *plan discret naïf* comme généralisation évidente de celle de la droite discrète naïve, nous donnons dans le chapitre 6 plusieurs propriétés vérifiées par les plans discrets. En particulier, la propriété de fonctionnalité permet de donner plusieurs types de codage bidimensionnel d'un plan discret naïf. Ces codages, largement utilisés par la suite, sont appelés *représentation par restes* et *représentation par niveaux* d'un plan discret. Des algorithmes de parcours et de construction sont aussi développés dans ce chapitre.



Représentation par voxels et par lignes de niveaux du plan discret naïf de vecteur normal  $(6, 13, 27)$  et de borne inférieure 0 pour  $x$  et  $y$  appartenant à  $[0, 9] * [0, 6]$ .

## La reconnaissance des plans discrets

La discrétisation d'un plan ou d'une surface produit d'énormes quantités de voxels. Il est intéressant de pouvoir retrouver les caractéristiques analytiques qui leur ont donné naissance et ceci au moins localement. Cela permet de contrôler ces données *au voxel près* et de leur appliquer des traitements spécifiques : lissage, squelettisation, opérations topologiques, compression . . .

Cette question a intéressé plusieurs chercheurs et nous décrivons dans le chapitre 7 quelques unes des méthodes utilisées. Les difficultés rencontrées par leurs auteurs dans l'étude des plans discrets proviennent d'une mauvaise définition. La définition arithmétique des plans discrets est plus appropriée et nous a permis, en nous inspirant des résultats antérieurs, d'obtenir un premier algorithme de reconnaissance de morceaux de plans discrets convexes, systématique, capable de tester tous les vecteurs normaux possibles en s'appuyant sur les codages par niveaux et par restes. Grâce à lui, nous avons pu élaborer une version géométrique incrémentale présentée dans le chapitre 8.

Précisons les traits essentiels de cet algorithme. Soit  $V$  un ensemble *convexe* de voxels à reconnaître, nous parcourons ses sections parallèles à un plan de coordonnées, en ajoutant les voxels un à un. Après plusieurs changements de vecteur normal, nous obtenons celui d'un plan ou une impossibilité. L'algorithme utilise des propriétés géométriques des plans discrets énoncées sous forme de conjectures. Cette partie théorique est illustrée de nombreux exemples de morceaux de plans discrets représentés selon le besoin de visualisation par voxels, niveaux ou par restes.

Le coeur de l'algorithme utilise essentiellement la détermination de l'enveloppe convexe *plane* de deux ensembles de points liés à  $V$ . Au fil de la reconnaissance ces enveloppes convexes 2D évoluent en fonction du point ajouté au morceau de plan courant. Les constructions effectuées sont réalisées grâce à des notions de géométrie 2D, par conséquent, la complexité de cet algorithme est très faible, en tout cas bien inférieure à celle des techniques proposées antérieurement où la reconnaissance

des plans discrets repose sur la construction d'enveloppes convexes 3D d'ensembles de voxels, opération délicate et coûteuse.

Enfin, le chapitre 9 est consacré aux applications de cet algorithme :

- Voisinage d'un point dans un plan discret

Une étude précise des voisinages discrets minimaux autour d'un point appartenant à un plan discret est la première étape pour définir une notion de différentiabilité discrète. L'algorithme de reconnaissance de morceaux de plans discrets nous a permis de caractériser les différents voisinages possibles qui sont appelés *tricubes*.

Nous proposons dans le neuvième chapitre du mémoire un dénombrement précis des différentes configurations de 8 voxels voisins d'un point appartenant à un plan discret, puis une propriété sur le nombre de tricubes différents dans un plan discret naïf donné et enfin une application sur la détection des points réguliers d'une image.

- Polyédrisation d'objets voxels symétriques

Il est clair que la discrétisation d'une surface au voisinage d'un point *régulier* donne en général un morceau de plan discret. On peut donc espérer *polyédriser* des surfaces discrètes en appliquant localement l'algorithme de reconnaissance de morceaux de plans discrets. Le point délicat de ce travail réside dans la stratégie de parcours puisque la polyédrisation n'est évidemment pas canonique. Un premier essai est proposé à la fin de ce mémoire.

Une conclusion termine ce mémoire où le bilan et les perspectives de recherches sont présentés.

## Chapitre 2

# Historique du problème de la reconnaissance des droites discrètes 2D

## 2.1 Introduction

La reconnaissance des segments de droites a intéressé de nombreux auteurs pendant les vingt dernières années ; ce problème intervient dans beaucoup de domaines comme le traitement d'image, la reconnaissance de formes, l'algorithmique graphique, la programmation en nombres entiers ... Voici un bref historique non exhaustif des travaux effectués depuis 1970.

En 1970, H. Freeman [24] a donné une méthode générale pour coder une courbe 8-connexe dans  $\mathbb{Z}^2$ . A chacune des directions principales, on associe un entier  $\alpha$  variant de 0 à 7 selon le schéma suivant :

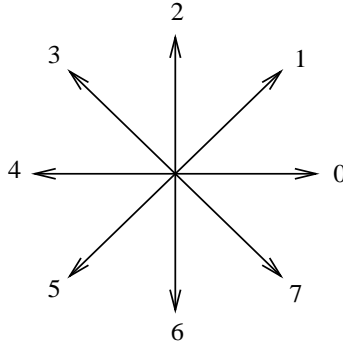


Figure 2.1 : Codage de Freeman.

On appelle *code de Freeman* d'une courbe, un point  $P_0$  et une suite  $(\alpha_i)$  pour  $i$  variant de 0 à  $n$  où  $\alpha_i$  est la direction suivant laquelle le point  $P_{i+1}$  est obtenu à partir de  $P_i$  :

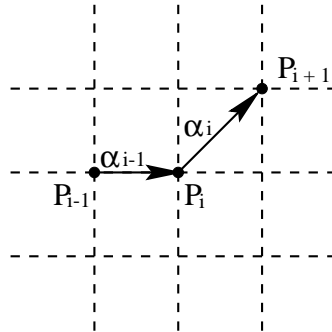


Figure 2.2 : Exemple avec  $\alpha_{i-1} = 0$  et  $\alpha_i = 1$ .

De plus, H. Freeman énonce trois critères que le code d'un arc discret doit vérifier pour être celui d'un segment de droite :

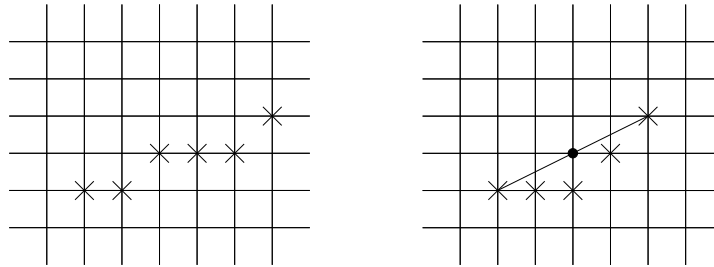
1. Deux directions au plus sont contenues dans le code et celles-ci ne diffèrent que d'une unité modulo 8.
2. Une de ces valeurs apparaît toujours seule.
3. Cet élément isolé est réparti aussi uniformément que possible.

Comme l'a indiqué T. Pavlidis dans [48], le troisième critère est assez flou et tous ont besoin d'une démonstration pour prouver que ces conditions caractérisent bien les droites discrètes.

En 1974 A. Rosenfeld [55] prouva que la propriété de la corde était une condition nécessaire et suffisante pour qu'un ensemble  $S$  de points discrets 8-connexes soit un segment de droite.

*Propriété de la corde :*

*$S$  vérifie la propriété de la corde si pour tout couple  $(P, Q)$  de points de  $S$  et pour tout point  $R$  de coordonnées  $(x, y)$  appartenant au segment de droite joignant  $P$  à  $Q$ , il existe un point (discret)  $T$  de coordonnées  $(i, j)$  de  $S$  tel que  $\max(|i-x|, |j-y|) < 1$ .*



**Figure 2.3 :** De gauche à droite, un ensemble vérifiant la propriété de la corde puis un autre ne la vérifiant pas.

Selon R. Brons [10], la propriété de la corde est une propriété conditionnelle et non structurelle comme les trois critères de Freeman. Dans [54], Ronse en a donné une preuve originale et simple.

A. Rosenfeld donna aussi une preuve des deux premiers critères de Freeman et plusieurs propriétés de régularité du code d'une droite discrète qui correspondent à la description du troisième critère.

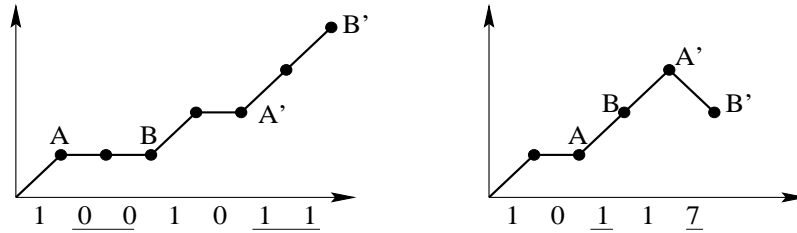
La même année, R. Brons [9] énonça un algorithme de construction de code ayant les trois propriétés de Freeman sans toutefois les démontrer.

Plus tard en 1975 et 1978, C. Arcelli et A. Massaroti [3, 4] prouvent que la courbe associée à un code donné par l'algorithme de Brons possède la propriété de la corde et par conséquent est le code d'un segment de droite discrète. Ils proposent ensuite un algorithme de reconnaissance de segments.

En 1982, Li de Wu [42] donne une nouvelle présentation des trois propriétés mentionnées par Freeman et prouve qu'elle est équivalente à la propriété de la corde. L'algorithme de reconnaissance qu'il propose est plus général que les précédents mais S.H.Y. Hung [27] y détecte une erreur.

En 1984 S.H.Y. Hung et T. Kasvand [28] introduisent une nouvelle caractérisation des codes de droites discrètes : l'absence d'irrégularité (unevenness) définie ci-dessous.

Deux segments d'égales longueurs dans la chaîne contenant le code sont dits **irréguliers** quand les sommes des éléments de chacun des deux segments diffèrent de plus de 1.



**Figure 2.4 :** Exemples de segments irréguliers : Sur le premier schéma,  $AB$  et  $A'B'$  sont de longueur 2, la somme des éléments de  $AB$  est égale à 0 et la somme des éléments de  $A'B'$  est égale à 2, les segments sont donc irréguliers. De même, sur le deuxième schéma les segments  $AB$  et  $A'B'$  sont des segments irréguliers ; la somme des éléments de  $AB$  égale 1 et la somme des éléments de  $A'B'$  est égale à 7.

Ils prouvent qu'un arc discret a la propriété de la corde s'il n'y a pas de segments "uneven" dans la chaîne contenant son code.

La même année L. Dorst et A.W.M. Smeulders [16] déterminent que le code  $C$  d'un segment de droite discrète peut être entièrement caractérisé par un quadruplet  $(n, q, p, s)$  défini par les quatre points suivants :

- $n$  le nombre d'éléments de  $C$ ,
- $q = \min\{k \in \{1, 2, \dots, n\} \mid k = n \vee \forall i \in \{1, 2, \dots, n - k\} : c_{i+k} = c_i\}$ ,
- $p = \sum c_i$  pour  $i$  variant de 1 à  $q$ ,
- $s : s \in \{0, 1, 2, \dots, q-1\} \mid \forall i \in \{1, 2, \dots, q\} : c_i = \lfloor p/q(i-s) \rfloor - \lfloor p/q(i-s-1) \rfloor$ .

Dans cette définition  $c_i$  représente le  $i$ ème élément de  $C$ , la fraction  $p/q$  représente la pente de la droite et  $s$  est un facteur de déphasage indiquant la position de la première période complète dans la chaîne contenant le code.

En 1991, ils proposent un algorithme linéaire de polygonalisation de courbes discrètes dans [17] en réécrivant les conditions de linéarité données par Li de Wu [42].

La même année, M. Lindenbaum et J. Koplowitz [44] suggèrent une autre paramétrisation des segments de droites discrètes utilisant un quadruplet plus simple à déterminer que celui proposé dans [16]. M. Lindenbaum et A. Bruckstein [43], en 1993, donnent un algorithme linéaire de segmentation de courbes discrètes reposant sur les propriétés des domaines des paramètres des segments de droites réelles dont la discrétisation correspond à un même ensemble de points discrets.

R.A. Melter et A. Rosenfeld [46], introduisent en 1989 le concept de segment de droite bruité qui repose sur la droite d'interpolation des moindres carrés. Dans cette voie, en 1993, avec I. Stojmenović et J. Zunić, R.A. Melter [47] prouve que la droite d'interpolation des moindres carrés détermine de manière unique la droite discrète support d'un segment, ils introduisent aussi une nouvelle caractérisation de segment de droite discrète.

J.P. Reveillès, en 1989, caractérise arithmétiquement les droites discrètes [51, 50] :

*Une droite discrète arithmétique de pente  $\frac{a}{b}$ , de borne inférieure  $\mu$  et d'épaisseur  $\omega$ , avec  $a, b, \mu$  et  $\omega$  dans  $\mathbb{Z}$ , est composée des points entiers de coordonnées  $(x, y)$  vérifiant la double inégalité diophantienne*

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

Cette approche est radicalement nouvelle et A.Troesch, en 1990 [61], utilise cette définition et obtient, en interprétant géométriquement l'algorithme d'Euclide, un algorithme de reconnaissance de segments de droites discrètes voisin de celui de Wu mais complète celui-ci par la détermination des caractéristiques des segments reconnus. V.A. Kovalesky [40] utilise une définition de droite discrète très proche de la définition arithmétique donnée par J.P. Reveillès.

*Quatre conditions équivalentes* caractérisent un segment de droite discrète ; elles sont énoncées sur la page suivante.

- *Les trois propriétés de Freeman.*
- *La propriété de la corde de Rosenfeld.*
- *L'absence de segments irréguliers de Hung et Kasvand.*
- *La double inégalité diophantienne de Reveillès*



Mentionnons aussi l'approche utilisant les mots de Sturm étudiée par J. Berstel et M. Pocchiola [5, 6].

Plusieurs algorithmes de reconnaissance de segments de droites discrètes [42, 17, 48, 3] utilisent les trois propriétés de Freeman, ce sont des algorithmes reposant sur la structure de ces droites (Brons [10]), or, en géométrie discrète la génération arithmétique d'une droite discrète [50] contrôle entièrement sa structure, il semble donc intéressant de regarder de manière plus précise dans la section suivante l'un de ces algorithmes : l'algorithme de Wu [42] et les améliorations qui lui ont été apportées. Ensuite, dans la section suivante, est détaillé le premier algorithme de reconnaissance de segment de droites [61] utilisant la définition arithmétique d'une droite discrète [50].

## 2.2 Algorithmes de reconnaissance utilisant les trois critères de Freeman

### 2.2.1 L'algorithme de Wu

Soit  $C$  une chaîne contenant le code de Freeman d'un arc discret respectant les deux premiers critères, i.e.  $C$  contient deux directions de base  $N$  et  $N+1$  (premier critère) et l'une de ces valeurs apparaît toujours isolée dans  $C$  (deuxième critère).

Exemples :

$C_1 : 0\ 0\ 1\ 0\ 0\ 1\ 0\ 0\ 1\ 0\ 1\ 0\ 0\ 1\ 0\ 1\ 0\ 0\ 1\ 0\ 0\ 1\ 0\ 0\ 1$

$C_1$  respecte les 2 critères,  $N = 0$  et  $N+1$  apparaît toujours seul .

$C_2 : 5\ 6\ 5\ 5\ 6\ 5\ 5\ 5\ 6\ 5\ 5\ 5$

$C_2$  respecte aussi les deux premiers critères.

$C_3 : 0\ 1\ 0\ 0\ 1\ 0\ 1\ 1\ 0\ 1$

$C_3$  viole la deuxième propriété, 1 et 0 n'apparaissent pas toujours de manière isolée.

Dans l'article [42], après avoir prouvé les deux premières propriétés de Freeman, Li de Wu définit le *troisième critère* de manière formelle :

1. *Les intervalles entre deux occurrences successives de la direction principale qui apparaît toujours seule doivent être égaux à  $N_1$  ou  $N_1 + 1$ , avec  $N_1$  un entier et une des valeurs  $N_1$  ou  $N_1 + 1$  apparaissant de manière isolée ; les intervalles*

entre deux occurrences successives de celle de ces deux valeurs qui apparaît seule doivent être égaux à  $N_2$  ou  $N_2 + 1$ , avec  $N_2$  un entier et une des valeurs  $N_2$  ou  $N_2 + 1$  apparaissant de manière isolée.

2. Les entiers  $N_1, N_2, \dots$  sont déterminés de manière unique par  $b$  la pente de la droite.

Il en déduit alors l'algorithme de reconnaissance de segments de droites discrètes donné sur la page suivante.

Exemples du déroulement de l'algorithme :

C1 : 0 0 1 0 0 1 0 0 1 0 1 0 0 1 0 1 0 0 1 0 0 1

◊  $N_1 = 0$  et le séparateur est 1

$$k_1 = 2$$

$$(n_1, n_2, \dots, n_8) = (2, 2, 2, 1, 2, 1, 2, 2)$$

$$k_2 = 0$$

$$M = 2$$

$k_1 = M$  et  $k_2 < M$  alors ,

$$C = (2, 2, 2, 1, 2, 1, 2, 2)$$

◊  $N_2 = 1$  et le séparateur est 1

$$k_1 = 3$$

$$n_1 = 1$$

$$k_2 = 2$$

$M = 1$  ou  $k_1 > M$  et  $k_2 > M$  donc le code est rejeté, C1 n'est pas le code d'un segment de droite discrète.

C2 : 5 6 5 5 6 5 5 5 6 5 5 5

◊  $N_1 = 5$  et le séparateur est 6

$$k_1 = 1$$

$$(n_1, n_2) = (2, 3)$$

$$k_2 = 3$$

$$M = 3$$

$k_1 < M$  et  $k_2 = M$  alors ,

$$C = (2, 3, 3)$$

◊  $N_2 = 2$  et le séparateur est 2, de plus il n'est présent qu'une fois dans C par conséquent C2 est le code d'une droite discrète.

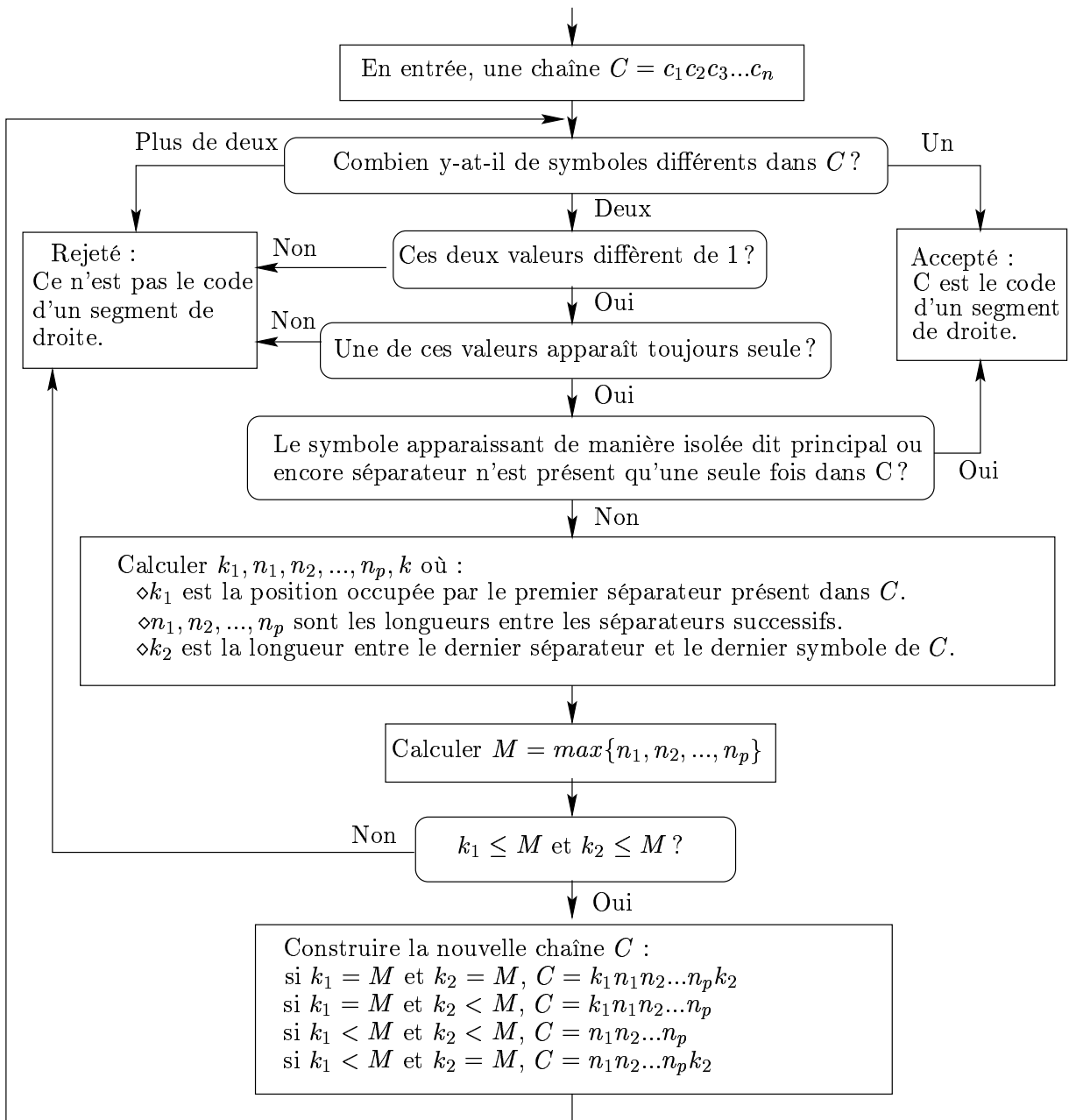


Figure 2.5 : L'algorithme de Wu.

Remarques :

- Cet algorithme ne calcule pas les caractéristiques des segments de droite trouvés.
- Il établit la droiture d'un arc discret en  $O(n)$  où  $n$  est le nombre d'éléments de la chaîne donnée en entrée.

### 2.2.2 Un problème soulevé par Hung

Dans [27], S.H.Y. Hung remarque certains cas négligés par Wu, regardons par exemple le code  $C = 0\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ 0$ . Déroulons l'algorithme de Wu avec la chaîne  $C$  en entrée :

$$\diamond N_1 = 0$$

$$k_1 = 2$$

$$(n_1, n_2, n_3) = (1, 1, 1)$$

$$k_2 = 2$$

Or,  $M = 1$  et  $k_1 > M$  et  $k_2 > M$  donc  $C$  est rejetée.

L'algorithme de Wu rejette cette chaîne alors que c'est celui d'un segment de droite ( $C' = 1\ 0\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ 0\ 1$  est reconnu par l'algorithme de Wu comme un segment de droite et  $C \subset C'$ ). Le fait que  $k_1$  et  $k_2$  soient supérieurs à  $M = \max\{n_1, n_2, \dots, n_p\}$  n'est pas suffisant pour rejeter le code, en effet, si comme dans l'exemple précédent  $k_1 = k_2 = M + 1$  et que tous les  $n_i$  ont la même valeur alors le code est celui d'une droite.

Pour remédier à cette omission, il faut modifier la fin de l'algorithme comme suit :

On remplace

par

$$\boxed{\text{Calculer } M = \max\{n_1, n_2, \dots, n_p\}}$$

$$\boxed{\text{Calculer } M' = \max\{k_1, n_1, n_2, \dots, n_p, k_2\}}$$

Ensuite,  $k_1$  et  $k_2$  sont inclus dans la nouvelle chaîne si et seulement s'ils sont égaux à  $M'$ . Il est clair si  $k_1$  et (ou)  $k_2 = M' \neq M + 1$  et (ou) les  $n_i$  n'ont pas tous la même valeur que la nouvelle chaîne sera rejetée au premier test de l'étape suivante.

## 2.2.3 Application à la polygonalisation de courbes discrètes

### a. Utilisation de l'algorithme de Wu

L'algorithme de Wu peut être utilisé dans un algorithme, donné sur la page suivante, de décomposition de courbes discrètes codées dans une chaîne R en segments de droites de longueur maximales.

- 
1. Partir d'une extrémité de R, le premier élément de code de la chaîne est mis dans la sous-chaîne SR.
  2. Tant que l'évaluation de la sous-chaîne SR par l'algorithme de Wu la reconnaît comme le code d'un segment de droite ajouter s'il existe l'élément de code suivant de R à SR sinon arrêter, toute la chaîne R a été parcourue.
  3. Initialiser une nouvelle sous-chaîne SR avec le dernier élément ajouté à l'ancienne sous-chaîne SR.  
Aller en 2.
- 

Pour une même sous-chaîne SR, l'extension et la vérification sont répétées jusqu'à ce que SR ayant pour longueur  $k_i + 1 (< n)$  ne soit plus le code d'un segment de droite. La chaîne de longueur  $k_i$  est alors un segment de droite de longueur maximale. Une nouvelle sous-chaîne indexée  $i+1$  est initialisée et le processus est répété jusqu'à ce que la chaîne entière ait été analysée.

L'algorithme de Wu analyse une chaîne de longueur  $j$  avec une complexité  $O(j)$ , pour la polygonalisation d'une courbe entière R de longueur  $n$  on a :

$$\begin{aligned} j &= 1, \dots, k_i \text{ avec } n = \sum k_i \\ i &= 1, \dots, K \text{ avec } K \text{ le nombre de segments} \end{aligned}$$

Par conséquent, la décomposition d'une chaîne entière en segments de droite en utilisant l'algorithme de Wu a au moins une complexité en  $O(n^2)$  :

$$\begin{aligned} \sum_{i=1}^K \sum_{j=1}^{k_i} O(j) &= \sum_{i=1}^K O(k_i(k_{i+1})/2) \\ &= \sum_{i=1}^K O\left(\frac{k_i^2}{2} + \frac{k_i}{2}\right) \\ &> K \times O(k^2) = O(n^2) \end{aligned}$$

avec  $k$  la moyenne supérieure des  $k_i$ .

## b. L'algorithme linéaire de Dorst et Smeulders

L. Dorst et A.W.M. Smeulders [17] ont trouvé un algorithme de polygonalisation de complexité  $O(n)$  en reconsidérant les conditions de linéarité données par Wu. Ils les reformulent dans un format séquentiel de telle sorte qu'une chaîne quelconque  $R$  peut être décomposée en sous-chaînes représentant des segments de droites en un seul parcours des  $n$  éléments de  $R$ . De plus, l'algorithme calcule les quatre paramètres qui caractérisent un segment de droite (dans [16] ils démontrent cette caractérisation).

Un bref exposé de la méthode employée semble nécessaire pour montrer que ce premier algorithme linéaire de polygonalisation de courbes discrètes datant de 1991 est assez compliqué et peu naturel.

La reformulation des conditions de linéarité nécessite un grand nombre de définitions :

La chaîne d'entrée  $R$  est appelée **chaîne d'ordre  $d=0$**  et notée  ${}^0R$ , elle est composée d'éléments ; les  ${}^0r_i$  qui sont compris entre 0 et 7. A chaque niveau de récursion  $d$ , on obtient une chaîne  ${}^dR$  et des éléments particuliers  $y$  sont définis :

- L'**élément maximal d'ordre  $d$** ,  ${}^d r_x$ , est défini par la valeur

$$\max\{{}^d r_i \mid i = 1, \dots, n({}^d R)\}$$

- L'**élément majoritaire d'ordre  $d$** ,  ${}^d r_M$ , est la valeur de l'élément le plus fréquent dans  ${}^d R$ .

Quand  ${}^d R$  a 2 éléments qui ont la même fréquence, donner à  ${}^d r_M$  la valeur de l'élément apparaissant en premier.

- L'**élément minoritaire d'ordre  $d$** ,  ${}^d r_N$ , est la valeur de l'élément le moins fréquent dans  ${}^d R$ .
- ${}^{d+1}p_i$  représente le nombre d'éléments de la  $i$ ème section d'éléments majoritaires concaténés de  ${}^d R$ .
- ${}^{d+1}p_0$  représente le nombre d'éléments majoritaires avant le premier  ${}^d r_N$  dans  ${}^d R$ .

L'**élément de début** du prochain ordre  ${}^{d+1}r_B$  est dit **complet** quand :

$${}^{d+1}p_0 = {}^{d+1}r_X - 1 \text{ alors } {}^{d+1}r_B = -\frac{1}{2}$$

sinon l'**élément de début** est dit **ouvert** et  ${}^{d+1}r_B = {}^{d+1}p_0 + \frac{1}{2}$ .

- L'élément  ${}^{d+1}r_i$  (ni premier, ni dernier) d'ordre  $d+1$  est calculé par :

$${}^{d+1}r_i = {}^{d+1}p_j + 1 \text{ avec } j = i \text{ si l'élément de début est}$$

ouvert et  $j = i - 1$  si l'élément de début est complet.

- ${}^{d+1}p_{n+1}$  représente le nombre de  ${}^d r_M$  après le dernier  ${}^d r_N$  dans  ${}^d R$ . L'**élément de fin** du prochain ordre  ${}^{d+1}r_E$  est dit **complet** quand :

$${}^{d+1}p_{n+1} = {}^{d+1}r_X - 1 \text{ alors } {}^{d+1}r_E = -\frac{1}{2}$$

- sinon l'**élément de fin** est dit **ouvert** et  ${}^{d+1}r_E = {}^{d+1}p_{n+1} + \frac{1}{2}$ .
- La chaîne  ${}^{d+1}R$  est la séquence de tous les  ${}^{d+1}r_i$  avec  $i = 1, \dots, n({}^{d+1}R)$ .  
On a :  $n({}^dR) = {}^{d+1}r_B + {}^{d+1}r_E + \sum {}^{d+1}r_i$ .
  - L'ordre terminal  $D$  est atteint quand  ${}^D R$  contient des éléments de même valeurs ( $= {}^D r_M$ ).

Exemples :

Si  ${}^0R = (5, 6, 5, 5, 6, 5, 5, 6, 5, 5, 5)$ , on a  ${}^0r_M = 5$ ,  ${}^0r_N = 6$ ,  ${}^0r_X = 6$ ,  
de plus  ${}^1p_0 = 1$ ,  ${}^1p_1 = 2$ ,  ${}^1p_2 = 2$ ,  ${}^1p_3 = 3$  et  ${}^1r_X = \max\{{}^1p_i + 1\} = 4$   
alors,

- ${}^1p_0 < {}^1r_X - 1$  l'élément de début est donc ouvert et  ${}^1r_B = {}^1p_0 + \frac{1}{2} = 1,5$
- L'élément de début est ouvert on peut alors calculer les  ${}^1r_i$  avec la relation  
 ${}^1r_i = {}^1p_i + 1$  d'où,  ${}^1r_1 = 3$  et  ${}^1r_2 = 3$ .
- ${}^1p_3 = {}^1r_X - 1$  l'élément de fin est complet et  ${}^1r_E = -\frac{1}{2}$  donc  ${}^1r_3 = 4$ .

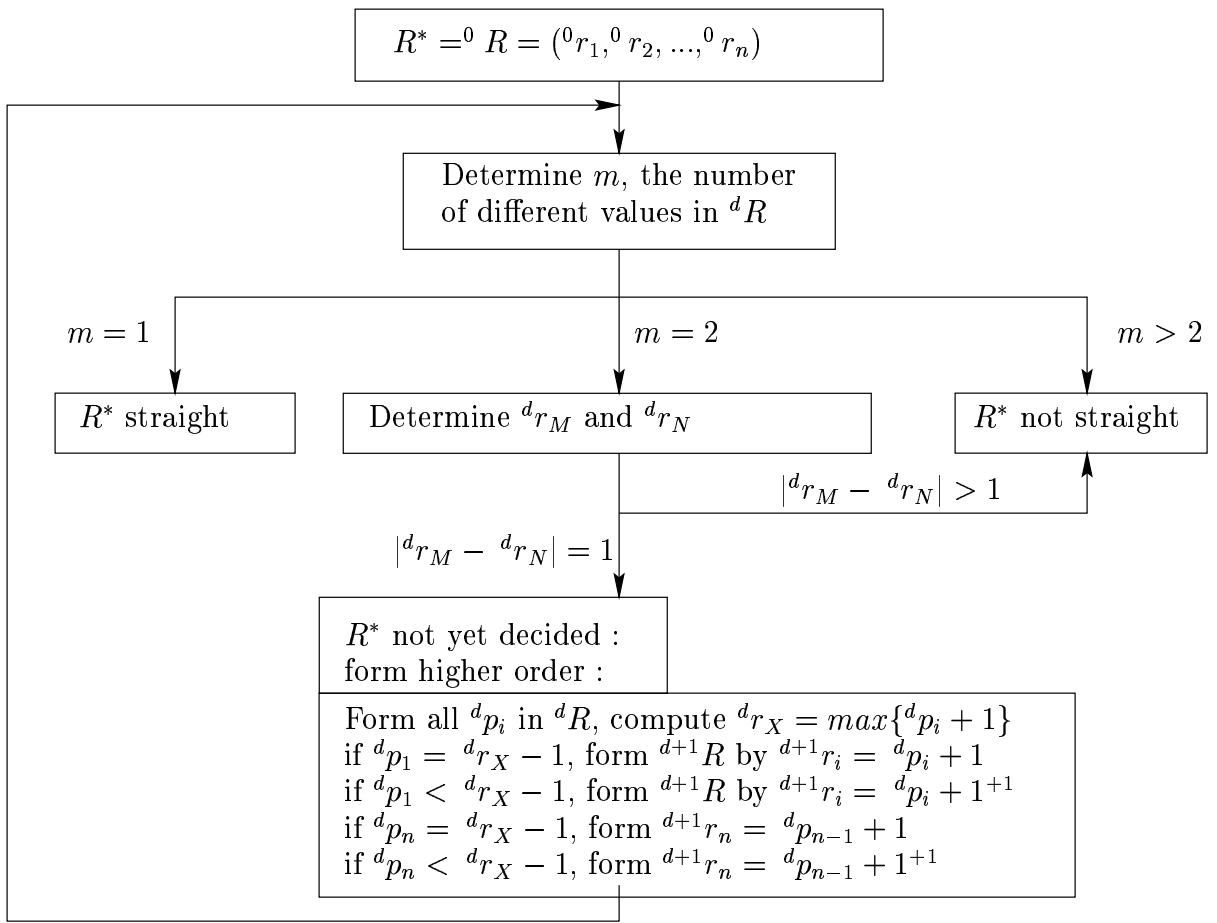
Par conséquent,  ${}^1R = (3, 3, 4)$ .

Avec les éléments décrits ci-dessus, ils redéfinissent les conditions de linéarité puis proposent un algorithme de reconnaissance (figure 2.6) qui à mon avis recèle quelques erreurs d'indices (en contradiction avec les définitions et l'exemple donnés).

Puis, à chaque ordre  $d$  ils associent un 4-uplet  ${}^d t$  formé avec les paramètres calculés :  ${}^d t = ({}^d r_M, {}^d r_N, {}^d r_B, {}^d r_E)$ .

Ils démontrent ensuite qu'une chaîne représentant une droite est *caractérisée de manière unique* par ce qu'ils appellent la **caractérisation**  $C^*$  de ses paramètres :  $C^* = (D, {}^0 t, {}^1 t, {}^2 t, \dots, {}^D t, n({}^D R))$ .

Après une extension, i.e. un élément rajouté à la chaîne  $R$ , il n'est plus nécessaire de recalculer tous les éléments de la nouvelle caractérisation  $C^*$ . Les quatre paramètres de chaque ordre évoluent en fonction des diagrammes donnés sur les pages suivantes pour chaque élément ajouté.



**Figure 2.6 :** *The linearity conditions.*

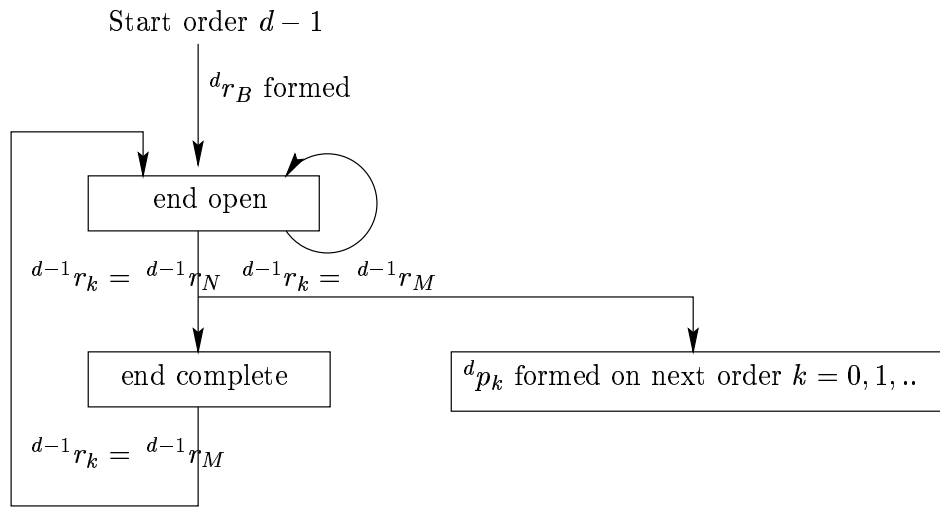
Les états que peut prendre  ${}^dr_E$  dans une chaîne représentant une droite sont :

- "end open" quand  ${}^dR$  ne se termine pas un  ${}^{d-1}r_N$ .
- "end complete" quand  ${}^dR$  finit par  ${}^{d-1}r_N$ .

Les états que peut prendre  ${}^dr_x$  dans une chaîne représentant une droite sont :

- "maximum unbounded" quand  ${}^dR$  ne contient pas d'élément complet d'ordre  $d-1$ .
- "maximum bounded" quand  ${}^dR$  a des éléments complets d'une seule valeur.
- "maximum known" quand  ${}^dR$  a deux éléments complets différents .





**Figure 2.7 :** *The states of the end run  ${}^d r_E$  while extending a straight string.*

Les états que peut prendre  ${}^d r_B$  dans une chaîne représentant une droite sont :

- "begin unknown" quand nous ne savons pas si l'élément de début de dR est ouvert ou complet.
- "begin open" quand l'élément de début est connu et ouvert ( ${}^d r_B > 0$ ).
- "begin complete" quand l'élément de début est connu et complet ( ${}^d r_B = -\frac{1}{2}$ ).

Les états que peuvent prendre  ${}^d r_M$  et  ${}^d r_N$  dans une chaîne représentant une droite sont :

- "no complete element known",
- "one element known",
- "majority known",
- "alternating",
- "all element known".

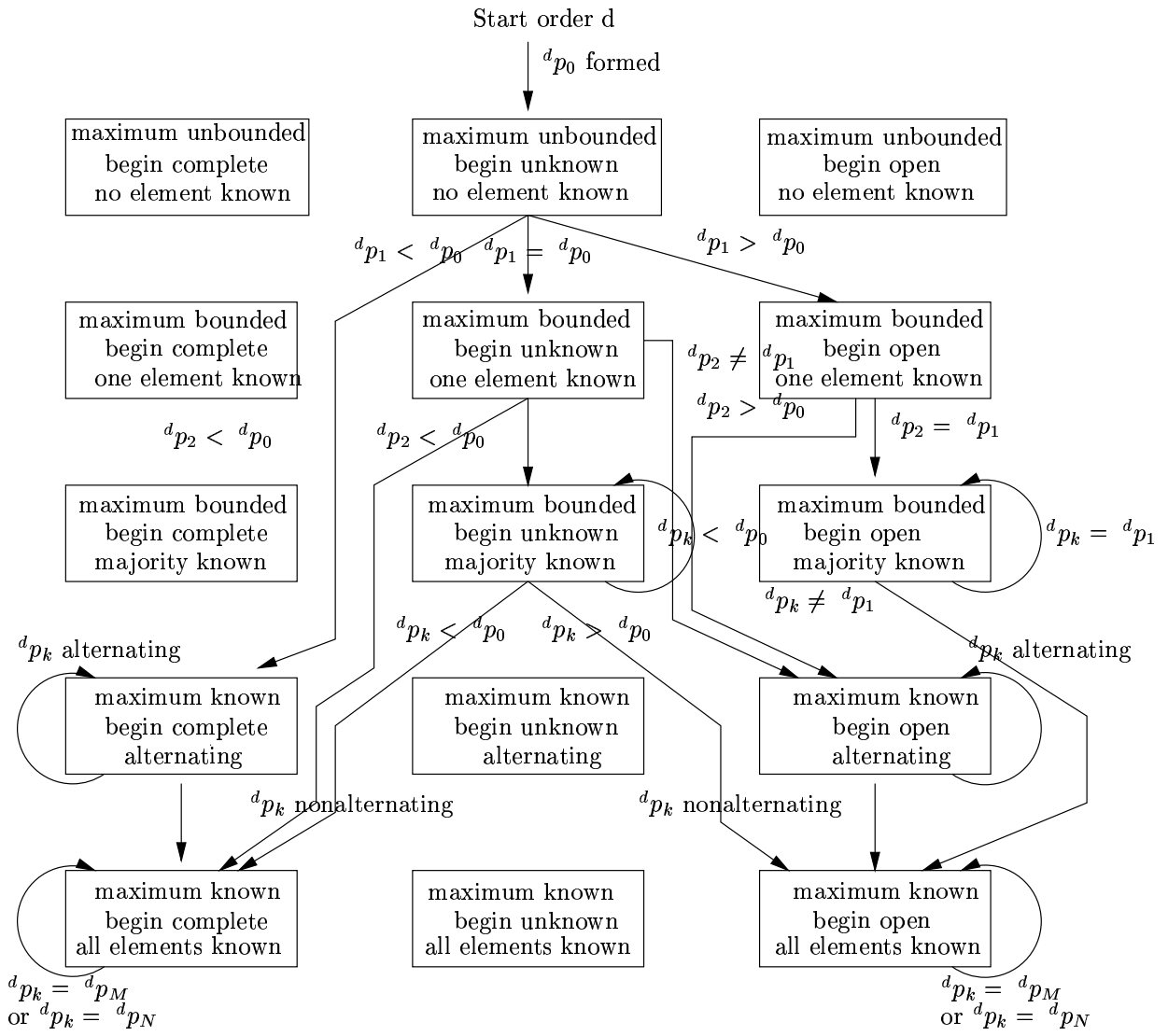


Figure 2.8

Ils en déduisent alors des *conditions de linéarités séquentielles* :

Soit  $S$  une extension d'une chaîne  $R$  par un élément , c'est à dire  ${}^0 s_i = {}^0 r_i$  pour  $i = 1, \dots, n({}^0 R)$  et  $n({}^0 S) = n({}^0 R) + 1$ .

*Théorème :*

$S$  est une droite si pour les ordres  $d = 0, \dots, d'$  (où  $d'$  est le plus petit ordre de  $R$  pour lequel l'état est "end open") les conditions séquentielles respectées par  ${}^d s_n$  et

$d_{s_E}$  sont :

(A0) Pour  ${}^dS$  dans l'état "start order  $d$ ",  $d_{s_n} > 0$ .

(A1) Pour  ${}^dS$  dans l'état "end complete" et pour  ${}^dR$  dans l'état "no complete element known",  $d_{s_n} > 1$  et  $d_{s_n} \geq d_{r_B} - 1$ .

(A2) Pour  ${}^dS$  dans l'état "end complete" et pour  ${}^dR$  dans l'état "one element known" ou "majority known",  $d_{s_n} > 1$  et  $d_{r_M} - 1 \leq d_{s_n} \leq d_{r_M} + 1$ .

(A3) Pour  ${}^dS$  dans l'état "end complete" et pour  ${}^dR$  dans l'état "Alternating" ou "all element known",  $d_{s_n} = \{d_{r_M}, d_{r_N}\}$ .

(B) Pour  ${}^dS$  dans l'état "end open",  $d_{s_E} < d_{r_X}$ .

Un algorithme linéaire de segmentation de courbes discrètes se déduit des conditions données à la page précédente.

Les quatre paramètres  $(n, p, q, s)$  caractérisant une droite [16], plus manipulables, sont ensuite calculés à partir des éléments de  $C^*$  et de règles de production assez complexes. La discrétisation des droites utilisée est celle qui garde les points entiers les plus proches situés sous la droite réelle.

Cette méthode ne contrôle pas la structure de la droite d'une manière simple, de nombreux états doivent être pris en considération et les caractéristiques  $(n, p, q, s)$  du segment trouvé ne sont pas calculées en même temps.

L'apport de la définition arithmétique d'une droite discrète [50] semble être une meilleure approche qui simplifie beaucoup le problème.

## 2.3 Interprétation géométrique de l'algorithme d'Euclide et application à la reconnaissance de droites discrètes

La méthode proposée par A.Troesch [61] repose sur l'interprétation géométrique de l'algorithme d'Euclide au moyen de transformations géométriques sur les droites discrètes définies de manière arithmétique [50]. Avec une approche différente, il obtient un algorithme voisin de celui de Wu [42], de même complexité, mais complète celui-ci en déterminant les caractéristiques des segments trouvés. Dans ce paragraphe figure un exposé des principaux résultats conduisant à l'algorithme de reconnaissance de segments de droites discrètes puis la mise en oeuvre de ces résultats aboutissant à un algorithme de polygonalisation de courbes discrètes.

### 2.3.1 Les droites discrètes

La définition arithmétique d'une droite discrète a été introduite par JP. Réveilles [51, 50] :

Une *droite discrète*

- ◇ de pente  $\frac{a}{b}$  avec  $b \neq 0$  et  $a \wedge b = 1$ ,
- ◇ de borne inférieure  $\mu$ ,
- ◇ d'épaisseur arithmétique  $\omega$ ,

est l'ensemble des points  $(x, y)$  de  $\mathbb{Z}^2$  satisfaisant la double inéquation diophantienne

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

avec tous les paramètres entiers.

Nous désignons la droite discrète précédente par  $\mathcal{D}(a, b, \mu, \omega)$ . Dans la suite, nous ne nous intéresserons qu'aux *droites naïves* qui vérifient  $\omega = \sup(|a|, |b|)$  et nous supposerons que les coefficients de la pente vérifient  $\frac{a}{b} > 0$ . Par suite, une droite naïve de pente  $\frac{a}{b}$  et de borne inférieure  $\mu$  sera notée  $\mathcal{D}(a, b, \mu)$ .

### 2.3.2 Le code fonctionnel

A. Troesch définit un code fonctionnel qui permet de coder les fonctions  $y = f(x)$ .

Soit  $f : \mathbb{Z} \rightarrow \mathbb{Z}$  une fonction, le nombre

$$c_x = f(x + 1) - f(x)$$

est appelé *code de f en x* et le *code de f* est la fonction  $c : \mathbb{Z} \rightarrow \mathbb{Z}$ .

Lorsque la pente de la fonction  $f$  est positive et inférieure à 1, le code de Freeman et le code fonctionnel coïncident. Dans la suite il n'utilise que ce dernier.

Il démontre ensuite que le code d'une droite discrète de pente  $\frac{a}{b}$  ne peut contenir que deux entiers distincts au plus et ces deux entiers diffèrent alors d'une unité :

$$c_x = \lfloor \frac{a}{b} \rfloor \text{ ou } c_x = \lfloor \frac{a}{b} \rfloor + 1$$

Nous appelons **palier horizontal** d'une droite discrète de pente inférieure à 1 un ensemble maximal de point de cette droite ayant la même ordonnée.

Exemple : Considérons la droite naïve  $\mathcal{D}(3, 5, 0)$ ,

$$\mathcal{D} = \{(x, y) | 0 \leq 3x - 5y < 5\}$$

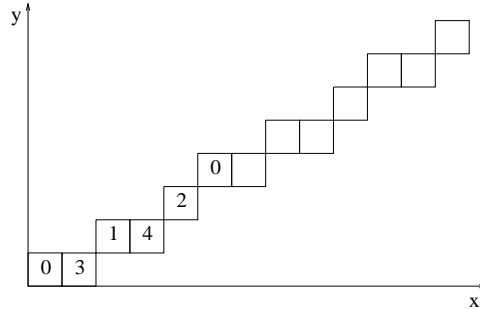


Figure 2.9

Sur les pixels figure la valeur  $3x - 5y$  au point  $(x, y)$  considéré. Les points  $(0, 0)$  et  $(1, 0)$  forment un palier horizontal. Le code fonctionnel à partir du point  $(0, 0)$  est 0 1 0 1 1 0 1 0 1 1 0 1 ...

### 2.3.3 Transformations géométriques et code

Regardons le comportement d'une droite discrète naïve  $\mathcal{D}(a, b, \mu, \omega)$  après lui avoir fait subir une transformation géométrique.

#### a. Glissement de pente entière

Un **glissement de pente entière**  $p$  est l'application  $G$  de  $\mathbb{Z}^2$  dans  $\mathbb{Z}^2$  définie par :

$$G(x, y) = (x, y - px)$$

Soit  $G$  le glissement de pente entière  $p \leq \frac{a}{b}$ , l'ensemble  $\mathcal{D}' = G(\mathcal{D})$  est la droite discrète de pente  $\frac{a}{b} - p$  et de borne inférieure  $\mu$ . Donc,

$$\mathcal{D}'(a - pb, b, \mu) = G(\mathcal{D}(a, b, \mu)).$$

De plus, si  $c$  est le code de  $\mathcal{D}$  et  $c'$  le code de  $\mathcal{D}'$  on a  $c' = c - p$ .

## b. Symétrie par rapport à la diagonale

La *symétrie par rapport à la diagonale* est l'application  $S$  de  $\mathbb{Z}^2$  dans  $\mathbb{Z}^2$  définie par :

$$S(x, y) = (y, x)$$

Si  $a < b$ , l'ensemble  $\mathcal{D}' = S(\mathcal{D})$  est la droite discrète de pente  $\frac{b}{a}$  et de borne inférieure  $1 - b - \mu$ . Donc,

$$\mathcal{D}'(b, a, 1 - b - \mu) = S(\mathcal{D}(a, b, \mu)).$$

De plus, il montre que la longueur du palier d'ordonnée  $y$  de la droite discrète de pente  $\frac{a}{b} < 1$  et de borne inférieure  $\mu$  est égale au code en  $y$  de la droite discrète de pente  $\frac{b}{a}$  et de borne inférieure  $1 - b - \mu$ . Les longueurs de palier de la droite de pente  $\frac{a}{b}$  ne peuvent prendre que deux valeurs  $\lfloor \frac{b}{a} \rfloor$  ou  $\lfloor \frac{b}{a} \rfloor + 1$ .

## c. La complémentation

La *complémentation* est l'application  $C$  de  $\mathbb{Z}^2$  dans  $\mathbb{Z}^2$  définie par :

$$C(x, y) = (x, x - y)$$

Si  $a < b$ , l'ensemble  $\mathcal{D}' = G(\mathcal{D})$  est la droite discrète de pente  $\frac{b-a}{b}$  et de borne inférieure  $1 - b - \mu$ . Donc,

$$\mathcal{D}'(b - a, b, 1 - b - \mu) = G(\mathcal{D}(a, b, \mu)).$$

De plus, si  $c$  est le code de  $\mathcal{D}$  et  $c'$  le code de  $\mathcal{D}'$  on a  $c' = 1 - c$ .

## 2.3.4 Interprétation géométrique de l'algorithme d'Euclide

Pour obtenir un nombre d'itérations moins important, il utilise une version légèrement modifiée de l'algorithme d'Euclide.

---

### Algorithme d'Euclide modifié

**Tant que  $r \neq 0$  répéter**

$r = a \bmod b$ ;

$r' = b - r$ ;

$r = \min(r, r')$ ;

a = b ;  
b = r ;  
**Fin répéter**

---

Chacune des itérations de cet algorithme peut se décomposer en une suite d'opérations :

remplacer a par a mod b ;  
si  $a \geq b$  – a remplacer a par b - a ;  
échanger a et b ;

Interprétée en termes d'opérations sur les droites discrètes, cette suite d'opérations arithmétiques correspond à la suite de transformations géométriques suivantes :

glissement de pente entière  $\begin{bmatrix} a \\ b \end{bmatrix}$  ;  
complémentation si  $a \geq b - a$  ;  
symétrie par rapport à la diagonale ;

Ceci se traduit en termes d'opérations sur les codes et on obtient l'algorithme d'Euclide appliqué sur le code des droites :

---

### Répéter

Soustraction de l'élément de code le plus petit à tous les éléments de code ;  
Echange des codes 1 et 0 dans le cas où les codes 1 ne sont pas isolés ;  
Remplacement des paliers de code 0...01 par leur longueur ;

**jusqu'à obtenir un code constant**

---

Dans le paragraphe précédent, nous avons vu que chacune de ces opérations effectuée sur le code d'une droite discrète redonne le code d'une droite discrète. Après une opération de glissement effectuée sur le code d'une droite, on obtient toujours le code d'une droite de pente inférieure à 1, i.e. située dans le premier octant.

Un exemple de déroulement de l'algorithme d'Euclide appliqué sur le code d'une droite discrète :

```

code :      1 1 0 1 1 1 0 1 1 0 1 1 0 1 1 1 0
itération 1 : 0 0 10 0 0 10 0 1 0 0 10 0 0 1
                3     4     3     3     4
itération 2 : 0 1 0 0 1
                2           3
itération 3 : 0 1
                2
stop, le code est constant

```

Par conséquent le code d'une droite discrète de pente rationnelle est caractérisé de la manière suivante :

1. Un nombre fini d'applications des opérations précédentes conduit à un code constant.
2. A chaque itération le code obtenu est formé uniquement de deux valeurs consécutives.

### 2.3.5 Algorithme de reconnaissance de segments discrets avec calcul des caractéristiques

Pour la reconnaissance des segments, la caractérisation donnée auparavant ne suffit pas, en effet, il faut tenir compte des extrémités.

Exemple :

```

code : 001 001 001 0001 001 0001 001 001 00
itération 1 : 3 3 3 4 3 4 3 3 3 (le dernier palier est complété par 1)
itération 2 : 0 0 0 1 0 1 0 0 0
                4     2     4

```

Ce code est rejeté à cause de la caractérisation 2 alors que c'est le code d'un segment de pente 5/17; il est contenu dans :

```
000100100010010010001001000100100010010010001001
```

La solution pour remédier à ce problème est de bien choisir les paliers externes que l'on conserve.

Dans une séquence finie formée de 0 et 1, pour laquelle les codes 1 sont isolés, on appelle *palier interne* un palier de code précédé par au moins un élément de code de valeur 1.

On choisit les conventions suivantes :



- S'il existe des paliers internes, un palier externe sera complet si sa longueur est strictement supérieure à la longueur du palier interne le plus court.
- S'il n'existe pas de palier interne, le palier externe le plus long sera complet.
- Seuls les paliers externes complets seront pris en compte, les autres seront négligés.

Exemples :

1. Reprenons l'exemple vu précédemment : 00100100100010010001001000100100  
 Les deux paliers externes ont pour longueur 3, qui est la longueur du palier interne le plus court ; on les néglige donc. Alors :

itération 1 :	3	3	4	3	4	3	3
itération 2 :	0	0	1	0	1	0	0
		3		2		3	
itération 3 :		1		0	1		
				2			

Stop, le code est constant, avec ces conventions il n'est plus rejeté.

2. code : 01000

Il n'y a pas de palier interne ; le palier terminal est complet, on néglige alors le palier initial et à la première itération on obtient un code constant égal à 4.

3. code : 10100101

Dans ce code, il n'y a pas de palier externe.

Si S est un segment de droite discrète, nous appelons *caractéristiques* de ce segment les quantités  $a, b, \mu, l, \frac{a}{b}$  étant la pente de la droite qui le porte,  $\mu$  sa borne inférieure et  $l$  sa longueur (i.e. son nombre de pixels).

Nous supposons que le segment est d'origine (0,0), cas auquel on peut toujours se ramener par translation.

Les caractéristiques du segment trouvé sont calculées au cours de l'algorithme de reconnaissance. Dans le cas où on aboutit à un code constant égal à q, le code donné en entrée est bien celui d'un segment de droite, alors ce code constant q obtenu après plusieurs transformations est celui d'un segment de droite discrète de pente entière q et de borne inférieure nulle. Il faut ensuite effectuer sur la pente et la borne inférieure les opérations inverses des opérations correspondant à chaque itération en utilisant les résultats donnés en 2.3.3.

Voici une version récursive de l'algorithme de reconnaissance qui en facilite la compréhension.

**ALGORITHME DE RECONNAISSANCE DE SEGMENTS DE DROITES DISCRETES SITUEES DANS LE PREMIER OCTANT :**

En entrée :

Une chaîne  $C$  codée avec le code fonctionnel (ou le code de Freeman) formée de 0 et de 1.

En sortie :

- $a, b, \mu$  et  $l$ , entiers représentant les caractéristiques du segment éventuel,
- `segment`, un booléen indiquant si  $C$  est le code d'un segment de droite ou non.

Initialisation :

$a = 1, b = 1, \mu = 0$  et  $l$  est le nombre d'éléments de  $C$   
`segment = vrai`

---

Procédure *reconnaissance*( $C, a, b, \mu, l, \text{segment}$ )

`max = L'élément de valeur maximale de C.`

`min = L'élément de valeur minimale de C.`

**Si** `max > min + 1` **alors**

`segment=faux; //C n'est pas le code d'un segment de droite//`

**Sinon**

**Si** `max ≠ min` **alors**

`// 1. Glissement //`

**Si** `min ≠ 0` **alors**

`soustraire l'élément de code le plus petit à tous les éléments de code;`

**Fsi**

`// 2. Complémentation //`

**Si** les codes 1 et 0 ne sont pas isolés **alors**

`échanger les codes 1 et 0;`

`complément = vrai;`

**Sinon**

`complément = faux;`

**Fsi**

`// 3. Symétrie //`

```

Si le palier initial est trop court alors
    le supprimer et mettre dans g la longueur de ce palier ;
Sinon
    g = 0 ;
Fsi
Si le palier terminal est trop court alors le supprimer ; Fsi
Remplacer les paliers de code 0...01 par leur longueur ;
// 4. Appel récursif //
reconnaissance(C, a, b,  $\mu$ , lg, segment) ;
// 5. Calcul des caractéristiques du segment trouvé //
Si segment = vrai alors
    échanger a et b ; // "inverse" de la symétrie //
     $\mu = 1 - a - \mu$  ;
    Tant que g > 0 répéter // déphasage éventuel //
         $\mu = \mu + a$  ;
        Si  $\mu > 0$  alors  $\mu = \mu - b$  ; Fsi
        g = g - 1 ;
    Fin répéter
    Si complément = vrai alors
        a = b - a ;
         $\mu = 1 - b - \mu$  ;
    Fsi
    a = a + min × b ;
Fsi
Sinon // min = max, le code est constant //
    a = min ; b = 1 ;  $\mu = 0$  ; // initialisation des caractéristiques //
Fsi
Fsi
l = nombre d'élément de C ;

```

---

Un exemple de déroulement de l'algorithme de reconnaissance :

Code donné en entrée :  $C = 0100100101$

Initialisation :  $a = 1, b = 1, m = 0, lg = 10$

◇ premier appel de la procédure reconnaissance(1, 1, 0, 10, vrai)

max = 1

min = 0

complément = faux

symétrie : Le palier initial est trop court, il est donc supprimé et  $g = 2$  et on obtient  $C = 3\ 3\ 2$ .

◇ deuxième appel de la procédure :

max = 3

min = 2

glissement :  $C = 1\ 1\ 0$

complémentation :  $C = 0\ 0\ 1$

complément = vrai

symétrie :  $C = 3$

$g = 0$

◇ troisième appel de la procédure :

max = min = 3 alors  $C$  est un segment de droite, initialisation des caractéristiques :

$$a = 3, b = 1, \mu = 0, l = 1$$

↔ retour dans le deuxième appel de la procédure :

$a = 1, b = 3,$

$\mu = 1 - 1 - 0 = 0,$

complément = vrai donc,  $a = 3 - 1 = 2$  et  $\mu = 1 - 3 - 0 = -2,$

$a = 2 + 2 \times 3 = 8,$

$l = 3.$

↔ retour dans le premier appel de la procédure :

$a = 3, b = 8,$

$\mu = 1 - 3 + 2 = 0,$

comme  $g = 2$  alors,  $\mu = -2,$

$l = 10.$

Fin de l'algorithme, les caractéristiques du segment trouvé sont :  $a = 3, b = 8,$

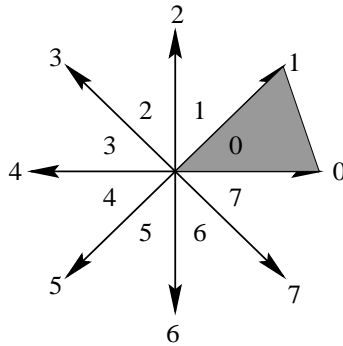
$$\mu = -2, l = 10.$$

### 2.3.6 Application à la polygonalisation de courbes discrètes

La procédure de reconnaissance énoncée dans le paragraphe précédent peut être utilisée dans un algorithme de polygonalisation de courbes discrètes.

En entrée : Une courbe 8-connexes quelconque codée dans une chaîne C selon le code de Freeman.

Le code C va être décomposé en sous-chaînes dont le code est celui de segments de droites de longueur maximale. Chaque sous-chaîne examinée se trouve dans un des 8 octants que nous numérotions comme suit :



**Figure 2.10 :** Les codes de Freeman correspondant à chaque octant se trouvent à l'extrémité des deux flèches qui le bordent.

Grace aux symétries intéressantes des droites discrètes, nous transformons le code SC' de chaque sous-chaîne composée par deux éléments de code de Freeman caractérisant un octant en une sous-chaîne SC formée de 0 et 1, SC est donc dans le premier octant et utilisable dans la procédure reconnaissance. L'octant initial est gardé en mémoire pour permettre le calcul des caractéristiques du segment d'origine.

Pour passer de l'octant d'origine à l'octant 0 et inversement, nous utilisons les résultats sur les symétries des droites discrètes donnés en [50] et rappelés ci-dessous.

Soit  $\mathcal{D}(a, b, \mu, \omega)$  une droite discrète alors sa transformée par symétrie par rapport à

- Ox est  $\mathcal{D}(a, -b, \mu, \omega)$  ou  $\mathcal{D}(-a, b, -\mu - \omega + 1, \omega)$
- Oy est  $\mathcal{D}(-a, b, \mu, \omega)$  ou  $\mathcal{D}(a, -b, -\mu - \omega + 1, \omega)$

- O est  $\mathcal{D}(-a, -b, \mu, \omega)$  ou  $\mathcal{D}(a, b, -\mu - \omega + 1, \omega)$
- la première bissectrice est  $\mathcal{D}(-b, -a, \mu, \omega)$  ou  $\mathcal{D}(b, a, -\mu - \omega + 1, \omega)$

---

### Algorithme de polygonalisation

**Initialiser** un segment SC avec le premier élément de C transformé par symétrie, garder en mémoire l'octant d'origine; segment=vrai;

**Tant que** C n'a pas été entièrement parcourue **répéter**

**Tant que** segment = vrai **répéter**

**Si** l'élément suivant de C se trouve dans le même octant que celui d'origine de SC **alors**

ajouter l'élément symétrisé à SC;

**Sinon** segment = faux;

**Fsi**

**Si** segment = vrai **alors**

appel de la procédure reconnaissance, si elle retourne vrai pour le booléen segment, garder en mémoire les caractéristiques trouvées;

**Fsi**

**Fin répéter**

SC moins le dernier élément ajouté est un segment de droite de longueur maximale, les caractéristiques du segment d'origine sont calculées avec celles gardées en mémoire à l'itération précédente en effectuant les symétries nécessaires pour retrouver l'octant d'origine;

**Initialiser** un nouveau segment avec le dernier élément ajouté; segment = vrai;

**Fin répéter**

---

Exemple :

Considérons le code de Freeman du cercle de Bresenham de rayon 2 :

$$C = 2\ 2\ 1\ 0\ 0\ 7\ 6\ 6\ 5\ 4\ 4\ 3$$

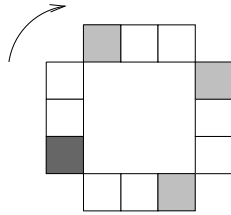
Les sous-chaînes données par l'algorithme sont :

$SC'_1 = 2\ 2\ 1$ , segment de droite se situant dans l'octant 1 de caractéristiques  
 $a = 3, b = 1, \mu = -2, l = 3$

$SC'_2 = 0\ 0\ 7$ , segment de droite se situant dans l'octant 7 de caractéristiques  
 $a = -1, b = 3, \mu = -2, l = 3$

$SC'_3 = 6\ 6\ 5$ , segment de droite se situant dans l'octant 5 de caractéristiques  
 $a = -3, b = -1, \mu = -2, l = 3$

$SC'_4 = 4\ 4\ 3$ , segment de droite se situant dans l'octant 3 de caractéristiques  
 $a = 1, b = -3, \mu = -2, l = 3$



**Figure 2.11** : Décomposition obtenue ; le pixel gris foncé est le pixel de départ et les pixels gris clairs sont les pixels de fins et débuts de segment.

Remarque : Par une démonstration similaire à celle faite en 2.2.3, on montre que cet algorithme a au moins une complexité en  $O(n^2)$  avec  $n$  le nombre d'éléments de la chaîne donnée en entrée.

## 2.4 Conclusion

Des algorithmes de reconnaissance de segments de droites discrètes reposant sur celui de Li de Wu [42] ont été exposés dans ce chapitre [27, 17] mais ils ne sont pas satisfaisants pour segmenter des courbes discrètes. L'algorithme linéaire de polygonalisation de courbes discrètes proposé par L. Dorst et A.W.M. Smeulders [17] repose sur une méthode compliquée de contrôle de la structure d'une droite discrète. A l'encontre des algorithmes vus précédemment, celui d'A. Troesch [61] repose sur des énoncés mathématiques et utilise la définition arithmétique des droites discrètes donnée par J.P. Reveillès [50], cependant, sa complexité est trop importante. De plus, les discussions sur les paliers extrémaux sont encore nécessaires ; la structure des droites n'est toujours pas entièrement contrôlée par l'apport mathématique. Pour obtenir un algorithme linéaire de segmentation de courbes discrètes, un algorithme de reconnaissance de segments de droites discrètes incrémental et simple est indispensable. Cet objectif fixé, nous avons expérimenté l'algorithme de reconnaissance d'A. Troesch sur de nombreux segments de droites discrètes en adoptant

une démarche incrémentale. Cette étude et une exploitation correcte de la définition arithmétique d'une droite discrète ont fourni des résultats très intéressants qui sont exposés dans le chapitre suivant et utilisés pour établir un algorithme linéaire de polygonalisation de courbes discrètes.





## Chapitre 3

# Un algorithme linéaire de polygonalisation de courbes discrètes planes

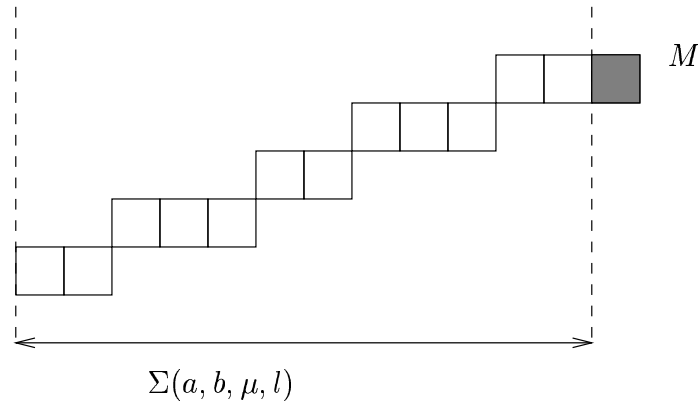
### 3.1 Introduction

L'étude sur différentes courbes discrètes de l'algorithme de segmentation inspiré des travaux d'A.Troesch a conduit à plusieurs remarques importantes et à l'élaboration d'un algorithme incrémentale de segments de droites discrètes puis d'un algorithme linéaire de polygonalisation de courbes discrètes très simple qui est exposé dans ce chapitre.

La situation incrémentale que nous avons étudiée utilise la définition suivante d'un segment de droite discrète :

L'ensemble des points d'une droite discrète naïve du premier octant, de caractéristiques  $a, b, \mu$  avec  $0 \leq a < b$  et d'abscisses appartenant à l'intervalle  $[0, l]$  est appelé **segment de droite discrète naïve** et noté  $\Sigma(a, b, \mu, l)$ .  $\mathcal{D}(a, b, \mu)$  sera appelée la **droite support** de  $\Sigma(a, b, \mu, l)$ .

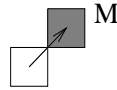
Considérons  $\Sigma$  un segment de droite discrète de caractéristiques  $a, b, \mu, l$  avec  $0 \leq a < b$  ( $\Sigma$  se situe dans le premier octant). Nous avons observé sur plusieurs exemples quelles sont les conséquences de l'ajout d'un point  $M$  au segment  $\Sigma$  et à quelles conditions  $\Sigma \cup \{M\}$  est un segment de droite discrète.



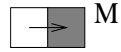
**Figure 3.1 :** *Situation incrémentale.*

$\Sigma$  étant dans le premier octant, il est clair que le point ajouté doit être obtenu

- soit avec le code de Freeman 1 :



– soit avec le code de Freeman 0 :



Sinon,  $\Sigma \cup \{M\}$  n'est clairement plus un segment de droite (violation du premier critère de Freeman).

Si on ajoute à  $\Sigma$  le point  $M(x_M, y_M)$  tel que  $x_M = l + 1$  dans la direction 0 ou 1, trois situations sont alors possibles :

1. Soit  $\Sigma \cup \{M\}$  est un segment de même droite discrète support que  $\Sigma$ , i.e. de caractéristique  $a, b, \mu$ .
2. Soit  $\Sigma \cup \{M\}$  n'est plus un segment de droite discrète.
3. Soit  $\Sigma \cup \{M\}$  est un segment de droite avec de nouvelles caractéristiques  $a', b', \mu'$ .

Des observations et une étude systématique de nombreux exemples ont permis d'élaborer un théorème qui, en fonction du point  $M$  ajouté à  $\Sigma$  et des caractéristiques de  $\Sigma$ , détecte par un calcul très simple dans quel cas de figure se trouve l'ensemble  $\Sigma \cup \{M\}$ . Ce théorème est exposé dans la première section de ce chapitre ainsi que l'algorithme incrémental de reconnaissance de segments de droites discrètes naïves du premier octant qui en est une conséquence immédiate.

Ensuite, grâce aux symétries des droites discrètes, un algorithme linéaire de polygonalisation de courbes discrètes, très efficace, est obtenu et des exemples de segmentation d'ellipses, d'hyperboles et de cercles discrets sont donnés.

Enfin, le rapport étroit entre l'algorithme de reconnaissance et l'arbre de Stern-Brocot est mis en évidence.

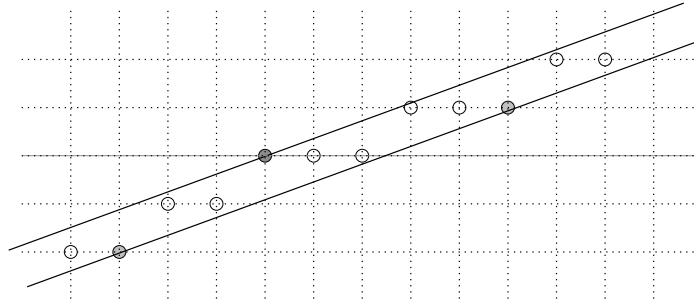
Dans toute ce chapitre, nous ne nous intéressons qu'aux droites discrètes naïves qui se situent dans le premier octant ( i.e. de pente  $0 \leq \frac{a}{b} < 1$ ).

## 3.2 Un algorithme incrémental de reconnaissance de segments de droites discrètes naïves

### 3.2.1 Points d'appui et droites d'appui

La réponse au problème posé en 3.1 repose sur des points particuliers d'une droite discrète naïve qui sont définis ci-dessous.

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



**Figure 3.2 :** Droites d'appui de la droite discrète  $\mathcal{D}(3, 8, -4)$  définie par l'équation  $-4 \leq 3x - 8y < 4$  pour  $x \in [0, 11]$ .

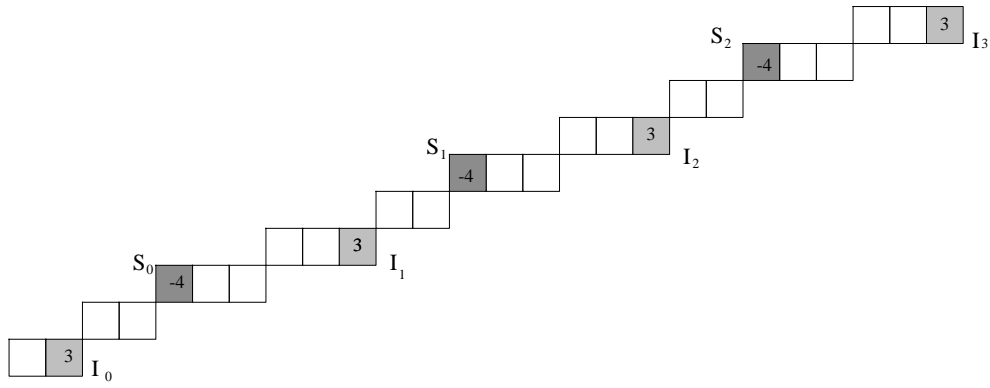
Nous appelons **pointillé de niveau  $k$**  de  $\mathcal{D}(a, b, \mu)$ , l'ensemble des points entiers vérifiant :

$$ax - by = k \text{ avec } k \in [\mu, \mu + b[$$

$\mathcal{D}(a, b, \mu)$  est la réunion de tous ses pointillés.

Les droites étudiées se situent dans le premier octant et, dans ce cas, les points de  $\mathcal{D}(a, b, \mu)$  se trouvent au dessus (resp. en dessous) de la droite d'appui d'équation  $ax - by = \mu + b - 1$  (resp.  $ax - by = \mu$ ) on la nommera donc **droite d'appui inférieure** (resp. **droite d'appui supérieure**) de  $\mathcal{D}$  et les pointillés correspondants **points d'appui inférieurs** (resp. **points d'appui supérieurs**).

Les solutions de l'équation  $ax - by = \mu$  sont de la forme  $x = x_0 + kb$  et  $y = y_0 + ka$  avec  $k \in \mathbb{Z}$  et  $(x_0, y_0)$  une solution particulière. Soit  $(u, v)$  tel que  $au - bv = 1$ , les solutions de l'équation  $ax - by = \mu + b - 1$  ont la forme  $x = x_0 - u + lb$  et  $y = y_0 - 1 - v + la$  avec  $l \in \mathbb{Z}$ . Par conséquent, dans chaque segment contenant  $b$  pixels nous trouvons au moins un point d'appui supérieur et un point d'appui inférieur. De plus, sur plusieurs périodes, les points d'appui inférieurs et supérieurs sont alternés (cf. figure 3.3).



**Figure 3.3 :** Points d'appui de  $\mathcal{D}(3, 8, -4)$  pour  $x \in [0, 25]$ ; en gris clair les points d'appui inférieurs appelés  $I_0, I_1, I_2, I_3$  et en gris foncé les points d'appui supérieurs,  $S_0, S_1$  et  $S_2$ . Les points d'appui sont pondérés par la valeur  $3x - 8y$  avec  $(x, y)$  les coordonnées des points considérés.

Deux points d'appui sont particulièrement importants dans la reconnaissance des segments de droites discrètes; les **points d'appui principaux** définis ci-dessous :

Soit  $\Sigma(a, b, \mu, l)$  un segment de droite ( $1 \leq a < b$ ) contenant au moins  $b$  pixels. Le **point d'appui principal inférieur** de  $\Sigma$  est le point d'appui inférieur possédant la plus petite abscisse. De même, le **point d'appui principal supérieur** de  $\Sigma$  est le point d'appui supérieur possédant la plus petite abscisse.

Sur la figure 3.3, le point d'appui principal supérieur est  $S_0$  et le point d'appui principal inférieur  $I_0$ .

### 3.2.2 Segment reconnu

Un segment de droite  $\Sigma(a, b, \mu, l)$  est un **segment reconnu** s'il possède au moins trois points d'appui.

Ainsi, un segment reconnu contient au moins trois points d'appui dont l'un est le point d'appui principal inférieur et l'autre le point d'appui principal supérieur de  $\Sigma$ ; le troisième est un point d'appui supérieur ou inférieur.

Sur la figure 3.3 nous voyons un segment reconnu,  $\Sigma(3, 8, -4, 25)$ .

### 3.2.3 Point faiblement et fortement extérieur

Considérons la droite discrète naïve  $\mathcal{D}(a, b, \mu)$ , les points entiers qui n'appartiennent pas à  $\mathcal{D}$  sont distribués dans une de ces deux classes :

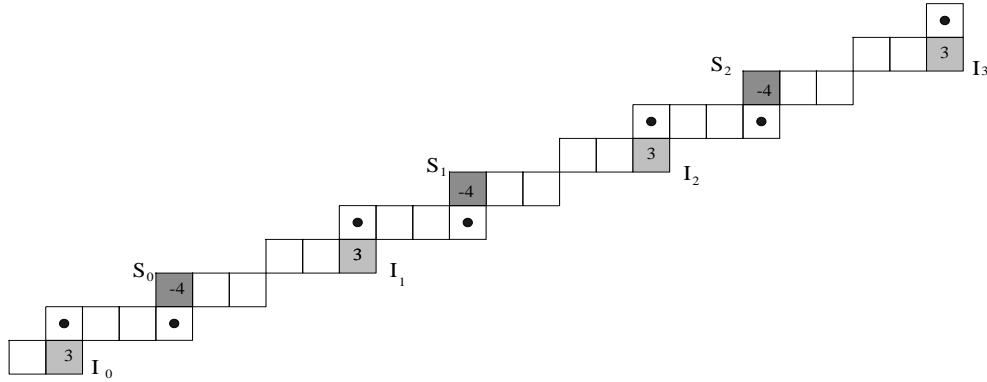
- Les points se situant le plus près possible d'une droite d'appui de  $\mathcal{D}$ . Ils vérifient l'une des conditions suivantes :

$$ax - by = \mu - 1 \text{ ou } ax - by = \mu + b$$

- Les autres points extérieurs remplissent l'une des deux conditions :

$$ax - by < \mu - 1 \text{ ou } ax - by > \mu + b$$

Les pixels satisfaisant l'équation  $ax - by = \mu - 1$  ou  $ax - by = \mu + b$  sont dits **faiblement extérieurs** à la droite  $\mathcal{D}$ . Ceux vérifiant l'une des deux conditions  $ax - by < \mu - 1$  ou  $ax - by > \mu + b$  sont dits **fortement extérieurs** à  $\mathcal{D}$ .



**Figure 3.4 :** Sur ce segment de la droite  $\mathcal{D}(3, 8, -4)$  représenté comme sur la figure 3.3, les points faiblement extérieurs ont été ajoutés ; ils sont représentés par des pixels contenant un point. Tous les autres points entiers (non représentés) sont fortement extérieurs.

Remarque : Soit  $E$  un point faiblement extérieur situé sous la droite  $\mathcal{D}$ , ses coordonnées vérifient l'équation  $ax - by = \mu + b$ . Alors  $ax - b(y + 1) = \mu$  montre que le point  $(x, y + 1)$  qui se trouve au dessus de  $E$  est un point d'appui supérieur de  $\mathcal{D}$  (cf figure 3.4). De même, les points faiblement extérieurs se situant au dessus de la droite peuvent être déduits des points d'appui inférieurs par un décalage vertical de longueur 1 (cf figure 3.4). Par conséquent, l'ensemble des points faiblement extérieurs de coordonnées  $(x_{fs}, y_{fs})$  situés au dessus de  $\mathcal{D}$  est défini par,

$$\begin{cases} x_{fs} = x_{I_0} + kb \\ y_{fs} = y_{I_0} + ka + 1 \end{cases} \quad k \in \mathbb{Z},$$

et les points faiblement extérieurs de coordonnées  $(x_{fi}, y_{fi})$  situés en dessous de  $\mathcal{D}$  sont définis de manière semblable.

### 3.2.4 Croissance d'un segment reconnu

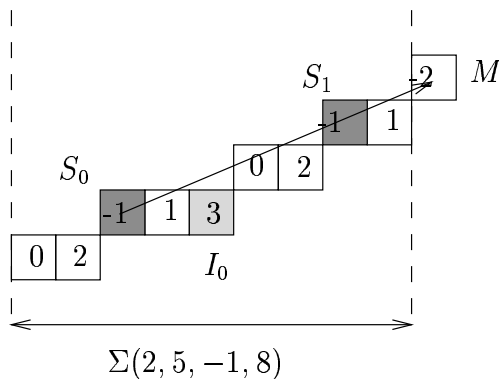
Soit  $\Sigma(a, b, \mu, l)$  un morceau reconnu de  $\mathcal{D}(a, b, \mu)$ . Supposons que le point  $M(x_M, y_M)$  (avec  $x_M = l + 1$ ) est ajouté à  $\Sigma$ ,  $\Sigma' = \Sigma \cup \{M\}$  est-il un segment de droite et, si c'est le cas, quelles sont ses caractéristiques  $a'$ ,  $b'$ ,  $\mu'$ ? La réponse à ces questions est donnée par le théorème fondamental suivant :

*Théorème 3.1 :*

Considérons  $r(M)$ , le **reste au point**  $M(x, y)$  en fonction de  $\mathcal{D}$  défini par :

$$r(M) = ax - by$$

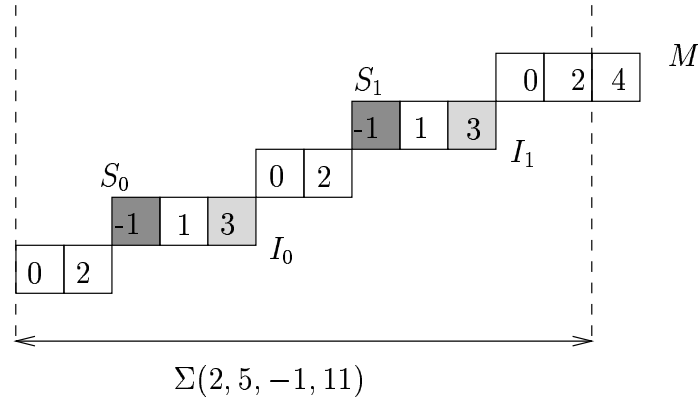
- (i) Si  $\mu \leq r(M) < \mu + b$ , alors  $M \in \mathcal{D}(a, b, \mu)$  ;  
 $\Sigma \cup \{M\}$  est un segment reconnu de la droite  $\mathcal{D}$ .
- (ii) Si  $r(M) = \mu - 1$ , alors  $M$  est faiblement extérieur à  $\mathcal{D}$  ;  
 $\Sigma \cup \{M\}$  est un segment reconnu de la droite de pente donnée par le vecteur  $S_0M$  avec  $S_0$  point d'appui principal supérieur de  $\mathcal{D}$  (cf figure 3.5).
- (iii) Si  $r(M) = \mu + b$ , alors  $M$  est faiblement extérieur à  $\mathcal{D}$  ;  
 $\Sigma \cup \{M\}$  est un segment reconnu de la droite de pente donnée par le vecteur  $I_0M$  avec  $I_0$  point d'appui principal inférieur de  $\mathcal{D}$  (cf figure 3.6).
- (iv) Si  $r(M) < \mu - 1$  ou  $r(M) > \mu + b$ , alors  $M$  est fortement extérieur à  $\mathcal{D}$  ;  
 $\Sigma \cup \{M\}$  n'est pas un segment de droite discrète.



**Figure 3.5 :** Segment de la droite discrète  $\mathcal{D}(2, 5, -1)$  de longueur 8, le point  $M$  ajouté a pour coordonnées  $(9, 4)$  et  $r(M) = -2$ ,  $M$  est faiblement extérieur à  $\mathcal{D}$  et situé au dessus de  $\mathcal{D}$ .  $\Sigma \cup \{M\}$  est un segment reconnu de la droite  $\mathcal{D}'$  de pente donnée par le vecteur  $S_0M = (7, 3)$ . Par conséquent,  $a' = 3$  et  $b' = 7$ , le point  $M$  est un point d'appui supérieur



de la nouvelle droite, alors  $\mu' = r'(M) = -1$  avec  $r'$  le reste au point  $M$  en fonction de  $\mathcal{D}'$ .  $\Sigma(2, 5, -1, 8) \cup \{M\} = \Sigma'(3, 7, -1, 9)$ .



**Figure 3.6 :** Segment de la droite discrète  $\mathcal{D}(2, 5, -1)$  de longueur 11, le point  $M$  ajouté a pour coordonnées  $(12, 4)$  et  $r(M) = 4$ ,  $M$  est faiblement extérieur à  $\mathcal{D}$  et situé en dessous de  $\mathcal{D}$ .  $\Sigma \cup \{M\}$  est un segment reconnu de la droite  $\mathcal{D}'$  de pente donnée par le vecteur  $I_0M = (8, 3)$ . Par conséquent,  $a' = 3$  et  $b' = 8$ , le point  $M$  est un point d'appui inférieur de la nouvelle droite, alors  $\mu' = r'(M) + b' - 1 = -3$  avec  $r'$  le reste au point  $M$  en fonction de  $\mathcal{D}'$ .  $\Sigma(2, 5, -1, 11) \cup \{M\} = \Sigma'(3, 8, -3, 12)$ .

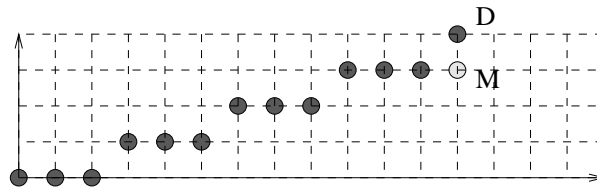
Démonstration du théorème :

Ce théorème est utilisé tel quel dans l'algorithme de reconnaissance de segments de droites discrètes qui est présenté dans le prochain paragraphe, mais la démonstration de ce résultat demande une légère reformulation :

Ajouter un point  $M(x_M, y_M)$  où  $x_M = l + 1$  à un segment  $S$  de caractéristiques  $a, b, \mu$  et  $l$  peut être considéré comme une **perturbation** du dernier point  $D(x_D, y_D)$  du segment de caractéristiques  $a, b, \mu$  et  $l + 1$  en  $M$ .

Exemple :

Considérons le segment  $\Sigma$  de caractéristiques  $a = 1, b = 3, \mu = 0$  et  $l = 11$  sur la figure 3.7, le point  $D(12, 4)$  est perturbé en  $M(12, 3)$ .



**Figure 3.7**

*Lemme 1 :*

Soit  $D(x_D, y_D)$  le point d'abscisse  $l + 1$  de la droite discrète  $\mathcal{D}(a, b, \mu)$ ,  $M(x_M, y_M)$  un point d'abscisse  $l + 1$  distinct de  $D$ ,  $\Sigma' = \Sigma(a, b, \mu, l) \cup \{M\}$  est un segment de droite discrète si et seulement si

–  $D$  est un point d'appui supérieur de  $\mathcal{D}$  et  $y_M = y_D - 1$ .

ou

–  $D$  est un point d'appui inférieur de  $\mathcal{D}$  et  $y_M = y_D + 1$ .

En effet,

• Soit  $D(x_D, y_D)$  un point d'appui supérieur alors,

$$ax_D - by_D = \mu$$

et il existe un entier  $k > 0$  tel que ,

$$D = (x_{S_0}, y_{S_0}) + k(b, a)$$

Le point  $M(x_M, y_M)$  est tel que :

$$ax_M - by_M = ax_D - b(y_D - 1) = ax_D - by_D + b = \mu + b$$

Comme  $ax_M - by_M = \mu + b$ ,  $M$  est dans le cas (iii) du théorème, alors  $\Sigma' = \Sigma \cup \{M\}$  est un segment de droite discrète de caractéristiques

$$a' = y_M - y_{I_0} = ka + y_{S_0} - y_{I_0} - 1$$

$$b' = x_M - x_{I_0} = kb + x_{S_0} - x_{I_0}$$

$$\mu' = a'x_M - b'y_M - b' + 1$$

en effet,

$$\begin{aligned} ab' - b'a &= a(kb + x_{S_0} - x_{I_0}) - b(ka + y_{S_0} - y_{I_0} - 1) \\ &= ax_{S_0} - b_{S_0} - ax_{I_0} + b_{I_0} + b \\ &= \mu - \mu - b + 1 + b \\ &= 1 \end{aligned}$$

Ceci prouve selon le théorème de Minkowski (page 24 de [25]) qu'il n'y a pas de point entier dans le parallélogramme formé par les vecteurs  $(b, a)$  et  $(b', a')$ . Par conséquent, les droites de pente  $a/b$  et  $a'/b'$  ont une même discrétisation sur l'intervalle  $[0, b']$ . Par suite, tous les points de  $\Sigma$  sont dans le segment de caractéristiques  $a', b', \mu', l$  et réciproquement, tous les points de ce segment sont dans  $\Sigma$ .

• On montre de même pour un point d'appui inférieur  $D(x_D, y_D)$  que le point  $M(x_M, y_M)$  est tel que :

$$ax_M - by_M = \mu - 1$$

$M$  est dans le cas (ii) du théorème, alors  $\Sigma' = \Sigma \cup \{M\}$  est un segment de droite discrète de caractéristiques

$$\begin{aligned} a' &= y_M - y_{S_0} = ka + y_{I_0} - y_{S_0} + 1 \\ b' &= x_M - x_{S_0} = kb + x_{I_0} - x_{S_0} \\ \mu' &= a'x_M - b'y_M \end{aligned}$$

en effet, on peut montrer que  $ab' - ba' = -1$  et par conséquent utiliser le théorème de Minkowski pour conclure.

• Si le point  $M$  ajouté n'est pas dans un des deux cas cités ci-dessus et comme  $M \neq D$  (selon les hypothèses ce n'est pas un point de la droite support du segment origine) alors  $M$  est un point fortement extérieur à  $\Sigma$ .

Supposons que  $M$  soit fortement en dessous de  $\Sigma$  et considérons le vecteur  $(u, v)$  tel que  $0 < u < b$ ,  $0 < v < a$  et  $au - bv = 1$ . En portant l'origine des axes au point d'appui principal inférieur  $I_0$ , l'équation de la droite devient  $1 - b \leq ax - by < 1$ . Si  $\Sigma$  possède deux points d'appui inférieurs, comme  $M$  est à droite du second point d'appui, on a  $x_M > b > u$ .

Si  $\Sigma$  ne possède qu'un seul point d'appui inférieur (l'origine), il possède alors deux points d'appui supérieurs de part et d'autre de l'origine. Le point  $M$  étant à droite du point d'appui supérieur  $S = (u, v + 1)$ , comme  $M$  est à droite du second point d'appui, on a  $x_M > b > u$ .

Dans ces conditions, nous allons montrer l'existence d'un point de coordonnées  $(\alpha, \beta)$  solution du système

$$\begin{cases} a\alpha - b\beta = 1 \\ y_M\alpha - x_M\beta \leq 0 \\ 0 < \alpha < x_M \end{cases}$$

Posons  $r = ax_M - by_M$ , comme  $au - bv = 1$ , nous pouvons écrire  $r = a(ru) - b(rv)$  et par conséquent il existe un entier  $k \in \mathbb{Z}$  tel que

$$x_M = ru + kb \text{ et } y_M = rv + ka.$$

L'équation  $a\alpha - b\beta = 1$  implique

$$\alpha = u + lb \text{ et } \beta = v + la \text{ pour } l \in \mathbb{Z}$$

Le point  $(\alpha, \beta)$  tel que  $x_M - b \leq \alpha < x_M$  est solution du système considéré. En effet, de  $\alpha = u + lb \geq x_M - b > u - b$ , nous déduisons  $l \geq 0$ , puis  $\alpha > 0$ . De plus, par un calcul immédiat nous obtenons  $y_M\alpha - x_M\beta = k - rl$ ; nous devons maintenant montrer que  $k - rl \leq 0$ . Partons de  $x_M - b \leq u + lb$  où nous remplaçons  $x_M$  par  $ru + kb$ , il vient :

$$(r - 1)u - b + kb \leq lb.$$

Comme  $M$  est fortement en dessous de  $\Sigma$ ,  $r \geq 2$  et nous avons alors  $-b < u - b \leq (r - 1)u - b$ . Il en résulte aussitôt  $-b + kb < lb$ , i.e.  $k \leq l$ , puis  $k - rl \leq k - l \leq 0$ . Nous avons ainsi démontré l'existence du point  $(\alpha, \beta)$  vérifiant le système considéré.

Comme  $a\alpha - b\beta = 1$ , le point  $S' = (\alpha, \beta + 1)$ , situé entre l'origine et  $M$  puisque  $0 < \alpha < M$ , est un point d'appui supérieur de la droite support de  $\Sigma$ . Il en résulte que le point  $S'$  se trouve à une distance verticale d'une unité de la droite  $(OM)$ , par conséquent, l'ensemble  $\Sigma' = \Sigma \cup \{M\}$  ne vérifie pas la propriété de la corde et ne peut donc pas être un segment de droite discrète.

Dans le cas où  $M$  est fortement au dessus de  $\Sigma$ , la démonstration est similaire. ■

Remarques :

1. Les nouvelles caractéristiques calculées  $a', b', \mu'$  sont telles que  $b'$  est le plus petit possible.
2. Une démonstration plus générale du théorème est proposée dans la publication [13].

### 3.2.5 L'algorithme incrémental

Le théorème énoncé précédemment permet d'obtenir l'algorithme incrémental de reconnaissance de segments de droites discrètes du premier octant présenté ci-après.

Une suite 8-connexe de  $N$  pixels à analyser, notée  $E$ , est donnée en entrée,  $E$  peut aussi être représentée par le code de Freeman et dans ce cas ne contenir que des 0 et des 1. Ensuite, chaque point de  $E$  va être analysé et ajouté au segment courant si cela est possible sinon, la reconnaissance s'arrêtera.

- Les coordonnées du point courant  $M$  ajouté sont  $(x_M, y_M)$  et les coordonnées des points d'appui principaux supérieurs et inférieurs sont respectivement  $(x_{S_0}, y_{S_0})$  et  $(x_{I_0}, y_{I_0})$ .
- $E$  peut être représentée par la suite des points  $(M_k)_{0 < k \leq N}$  (avec  $N > 0$ ), cette courbe se situant dans le premier octant, pour tout  $k \in ]0, N]$ , on a  $0 \leq y_{M_k} < x_{M_k}$  et  $M_0 = (0, 0)$ .

---

Algorithme de reconnaissance de segments de droites discrètes naïves  
situées dans le premier octant

```
// initialisation //                                {cf. (i)}
a = 0 ; b = 1 ;  $\mu$  = 0 ;
M = premier point de E ;
S0 = M ; I0 = M ; SEGMENT=VRAI ;
Tant que E n'a pas été entièrement parcouru et SEGMENT répéter
    M = point suivant de E
```

$r_M = ax_M - by_M$  ;  
**SEGMENT** =  $r_M \geq \mu - 1$  et  $r_M \leq \mu + b$  ;     {cf. (ii)}  
**Si**  $r_M = \mu - 1$  ou  $r_M = \mu + b$  **alors**  
    **Si**  $r_M = \mu - 1$  **alors**     {cf. (iii)}  
         $x_P = x_{S_0}$  ;  $y_P = y_{S_0}$  ;  
         $x_{I_0} = x_M - b$  ;  $y_{I_0} = y_M - a - 1$  ;  
    **Fsi**  
    **Si**  $r_M = \mu + b$  **alors**     {cf. (iv)}  
         $x_P = x_{I_0}$  ;  $y_P = y_{I_0}$  ;  
         $x_{S_0} = x_M - b$  ;  $y_{S_0} = y_M - a + 1$  ;  
    **Fsi**  
         $b = x_M - x_P$  ;  $a = y_M - y_P$  ;  
         $\mu = ax_{S_0} - by_{S_0}$  ;

**Fsi**

**Fin répéter**

---

(i) Les caractéristiques du segment courant sont initialisées par celles de la droite horizontale  $y = 0$  et les points d'appui principaux sont initialisés en conséquence.

(ii) Si le point  $M$  est fortement extérieur, la reconnaissance s'arrête ; l'ensemble des points déjà parcourus plus le point  $M$  n'est pas un segment de droite discrète.

(iii) Quand le point courant  $M$  est faiblement au dessus de la droite support du segment,  $(S_0M)$  dirige la nouvelle droite d'appui supérieure et la parallèle passant par le nouveau point d'appui principal inférieur  $I_0 = M - (b, a + 1)$  est la nouvelle droite d'appui inférieure, le point d'appui principal supérieur est inchangé. L'ensemble des points déjà parcourus plus  $M$  est un segment de la droite discrète  $\mathcal{D}'(a', b', \mu')$  avec

$$b' = x_M - x_{S_0}, a' = y_M - y_{S_0}, \mu' = a'x_M - b'y_M = a'x_{S_0} - b'y_{S_0}.$$

Remarquons que  $a'$  et  $b'$  sont premiers entre eux car les conditions  $ax_{S_0} - by_{S_0} = \mu$  et  $ax_M - by_M = \mu - 1$  impliquent  $ab' - ba' = -1$ .

(iv) Quand le point courant  $M$  est faiblement en dessous de la droite support du segment,  $(I_0M)$  dirige la nouvelle droite d'appui inférieure et la parallèle passant par le nouveau point d'appui principal supérieur  $S_0 = M + (-b, -a + 1)$  est la nouvelle droite d'appui supérieure, le point d'appui principal inférieur est inchangé. L'ensemble des points déjà parcourus plus  $M$  est un segment de la droite discrète  $\mathcal{D}'(a', b', \mu')$  avec

$$b' = x_M - x_{I_0}, a' = y_M - y_{I_0}, \mu' = a'x_M - b'y_M - b' + 1 = a'x_{S_0} - b'y_{S_0}.$$

Remarquons que  $a'$  et  $b'$  sont premiers entre eux car les conditions  $ax_{I_0} - by_{I_0} =$

$$\mu + b - 1 \text{ et } ax_M - by_M = \mu + b \text{ impliquent } ab' - ba' = 1.$$

Cet algorithme reconnaît des segments de taille maximale en donnant des caractéristiques  $a, b, \mu$  avec le plus petit  $b$  possible. Un exemple d'exécution de l'algorithme est donné dans la section suivante .

### 3.2.6 Un exemple détaillé de déroulement de l'algorithme de reconnaissance de segments de droites discrètes

Considérons la suite E de pixels de la figure 3.8 à analyser.

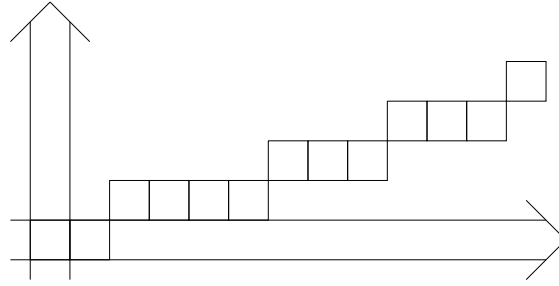


Figure 3.8

La première étape consiste à initialiser les caractéristiques :

$$a_0 = 0, b_0 = 1, \mu_0 = 0, D_0 : 0 \leq -y < 1$$

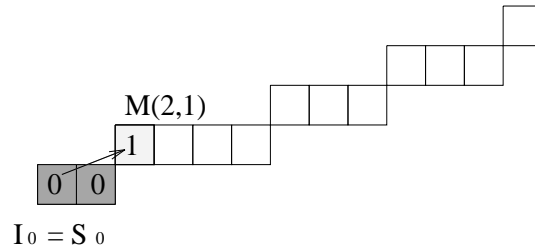
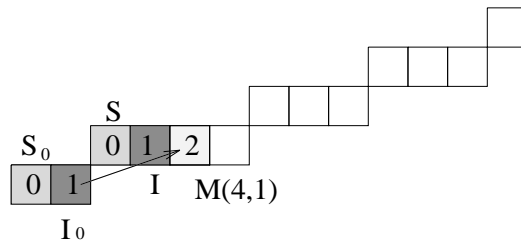


Figure 3.9 : Représentation par restes en fonction de  $D_0$  des pixels déjà parcourus.

Le pixel d'abscisse 1 appartient à  $D_0$  et a pour reste 0 en fonction de  $D_0$  mais le point de coordonnées  $(2, 1)$  de E a pour reste  $-1$  ; il est faiblement extérieur à  $D_0$ , situé au dessus (point  $\{iii\}$  de l'algorithme) par conséquent l'ensemble des points déjà parcourus plus  $M$  est un segment de la droite discrète  $D_1(a_1, b_1, \mu_1)$  définie par  $a_1 = 1, b_1 = 2, \mu_1 = 0, D_1 : 0 \leq x - 2y < 2$  et le point d'appui principal inférieur devient  $I_0 = [x - b_0, y - a_0 - 1] = [1, 0]$  et  $S_0$  reste identique.

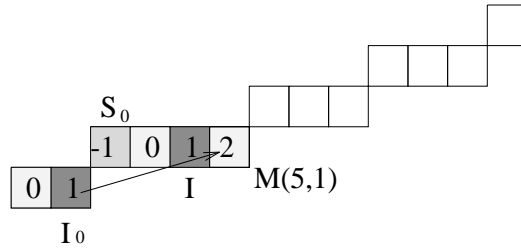


**Figure 3.10 :** Représentation par restes en fonction de  $D_1$  des pixels déjà parcourus.

Le pixel d'abscisse 3 appartient à  $D_1$ , mais le suivant est faiblement extérieur, situé en dessous de  $D_1$  (point  $\{(iv)\}$  de l'algorithme) par conséquent l'ensemble des points déjà parcourus plus  $M$  est un segment de la droite discrète  $D_2(a_2, b_2, \mu_2)$  définie par :

$$a_2 = 1, b_2 = 3, \mu_2 = -1, D_2 : -1 \leq x - 3y < 2$$

et le point d'appui principal supérieur est modifié et devient  $S_0 = [x - b_1, y - a_1 + 1] = [2, 1]$ .

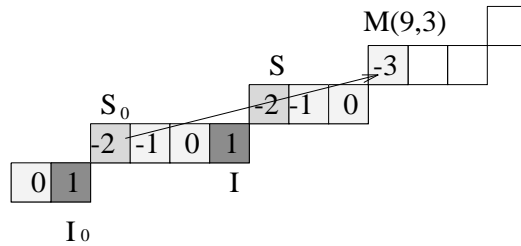


**Figure 3.11 :** Représentation par restes en fonction de  $D_2$  des pixels déjà parcourus.

La reconnaissance se poursuit et le point suivant de coordonnées (5, 1) provoque un nouveau changement qui conduit à la droite support  $D_3(a_3, b_3, \mu_3)$  :

$$a_3 = 1, b_3 = 4, \mu_3 = -2, D_3 : -2 \leq x - 4y < 2,$$

$S_0 = [2, 1]$  et  $I_0$  est inchangé.

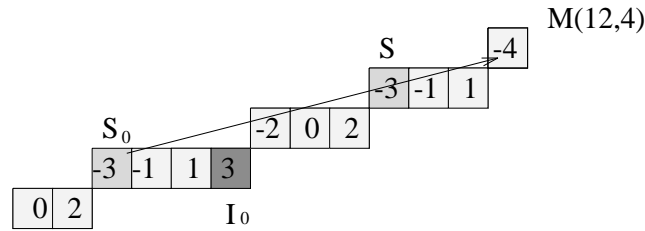


**Figure 3.12 :** Représentation par restes en fonction de  $D_3$  des pixels déjà parcourus.

Les pixels d'abscisses 5, 6, 7 appartiennent à  $D_3$  mais le point de coordonnées (9, 3)

est faiblement extérieur à  $D_3$ , situé au dessus, par conséquent l'ensemble des points déjà parcourus plus ce point est un segment de la droite discrète  $D_4(a_4, b_4, \mu_4)$  définie par :

$a_4 = 2, b_4 = 7, \mu_4 = -3, D_4 : -3 \leq 2x - 7y < 4$  et le nouveau point d'appui principal inférieur  $I_0 = [5, 1]$ .



**Figure 3.13 :** Représentation par restes en fonction de  $D_4$  des pixels déjà parcourus.

Le dernier point ajouté  $(12, 4)$  est faiblement extérieur à  $D_4$  ce qui conduit aux caractéristiques suivantes de la droite support  $D_5$  :

$$a_5 = 3, b_5 = 10, \mu_5 = -4, D_5 : -4 \leq 3x - 10y < 6, I_0 = [5, 1]$$

L'ensemble E est donc le segment de droite discrète  $\Sigma(3, 10, -4, 12)$ .

Pour obtenir ces caractéristiques avec l'algorithme de reconnaissance, 5 étapes ont été nécessaires engendrant à chaque fois une nouvelle droite support de l'ensemble des points déjà parcourus. Les points de coordonnées  $(2, 1), (4, 1), (5, 1), (9, 3)$  et  $(12, 4)$  sont à l'origine de ces changements; ils sont appelés **points d'articulation** de la reconnaissance du segment  $\Sigma(3, 10, -4, 12)$ .

### 3.3 La polygonalisation de courbes discrètes

#### 3.3.1 L'algorithme linéaire de polygonalisation de courbes discrètes 8-connexes dans tous les octants

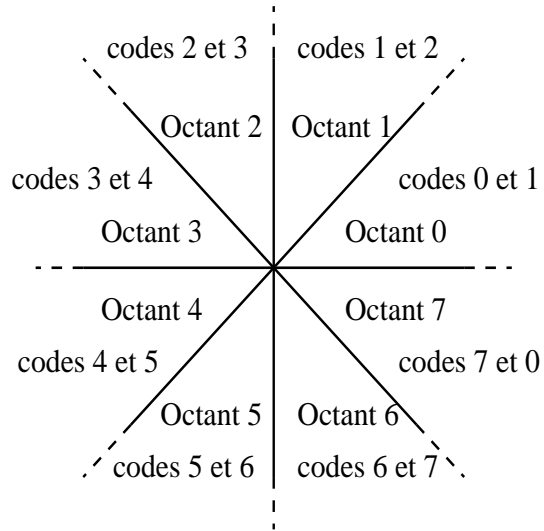
Le théorème et l'algorithme présentés dans la section précédente peuvent être utilisés très simplement pour segmenter une courbe discrète 8-connexe quelconque, donnée par son code de Freeman, en examinant incrémentalement chacun de ses points pour former des segments discrets de longueurs maximales. La linéarité de cet algorithme est claire.

De même que pour l'algorithme proposé en 2.3, nous utilisons les théorèmes de structure et les symétries des droites discrètes (2.3.6 et [50]) pour ramener toutes les opérations à un arc discret du premier octant, ceci nous permet d'obtenir un



algorithme particulièrement simple.

La détection de l'octant où se trouve le segment étudié et des changements possibles se fait grâce au code de Freeman et à son premier critère ; dans un octant donné, un segment de droite discrète est décrit avec 2 codes différents, ils sont rappelés sur le schéma suivant :



---

Algorithme de polygonalisation de courbes discrètes 8-connexes

M = Premier point de la chaîne ;  
**Tant que** la chaîne représentant la courbe est non vide **répéter**  
// initialisation des caractéristiques d'un nouveau segment SC //  
a = 0 ; b = 1 ;  $\mu = 0$  ; l = 0 ;  
S<sub>0</sub> = M ; l<sub>0</sub> = M ; SEGMENT=vrai ;  
**Tant que** le point ajouté  $M(x_M, y_M)$  est dans une des situations (i), (ii) ou (iii)  
du théorème 3.1 et SEGMENT **répéter**  
    Si nécessaire, recalculer les caractéristiques du segment et les points d'appui  
    principaux de la nouvelle droite support ;  
    Incrémenter l ;  
    **Si** l'élément suivant de la chaîne appartient au même octant que celui d'origine  
    de SC **alors**  
        ajouter l'élément symétrisé à SC ;  
    **Sinon**  
        SEGMENT = faux ;  
    **Fsi**  
**Fin répéter**  
En fonction des symétries, calculer les caractéristiques du segment de droite discrète  
dans l'octant initial ;  
**Fin répéter**

---

Remarques :

1. Le premier point d'un nouveau segment est le dernier du segment précédent.
2. Les segments obtenus sont de longueurs maximales.
3. Cet algorithme n'utilise pas directement la décomposition en paliers d'une droite discrète et évite ainsi la principale difficulté rencontrée par les algorithmes précédents.
4. Une autre méthode pour passer d'un octant à l'autre utilisant des changements de repères est proposée dans [13] et le code pascal de l'algorithme est donné.

Un exemple détaillé de déroulement de l'algorithme :

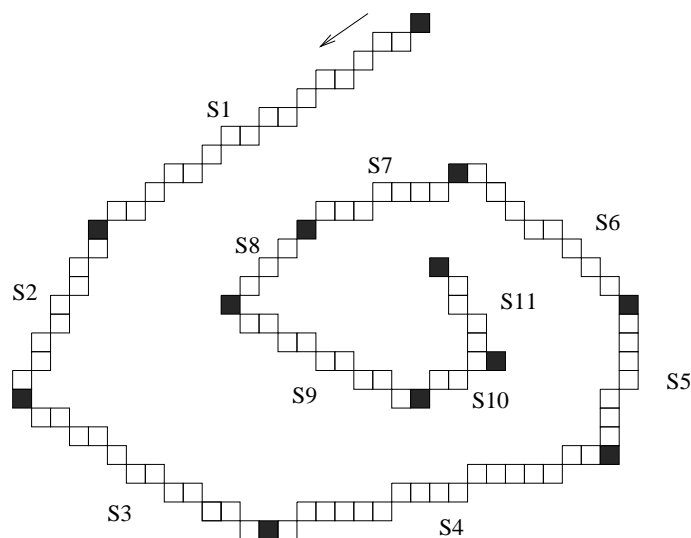


Figure 3.14

La décomposition en segments de droites de cette courbe discrète est donnée dans le tableau suivant où  $a$ ,  $b$ ,  $c$  et  $\mu$  sont les caractéristiques du segment décrit :

Nom du segment	$a$	$b$	$\mu$	$l$
S1	-5	-8	-5	17
S2	-2	-1	-1	9
S3	-5	9	-2	13
S4	2	9	-5	18
S5	5	1	-3	8
S6	7	-9	0	9
S7	-2	-7	-5	8
S8	-1	-1	0	4
S9	-1	2	0	10
S10	1	2	-1	5
S11	2	-1	0	5

Nous donnons sur les pages suivantes quelques exemples de décomposition de lignes brisées, de cercles, d'ellipse et d'hyperboles discrets.





### 3.3.2 Polygonalisation de cercles discrets

En observant de manière plus approfondie la polygonalisation des cercles de Bresenham, nous pouvons faire quelques remarques :

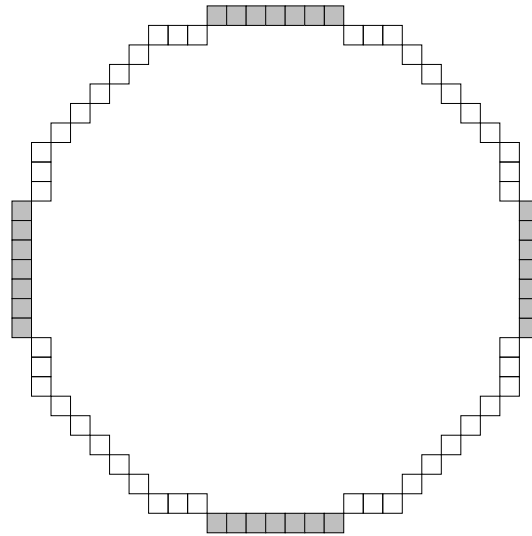
- Dans tout cercle de Bresenham de rayon  $R > 0$ , deux paliers horizontaux et deux paliers verticaux sont maximaux et de même longueur. La longueur de ces paliers peut être calculée en fonction du rayon  $R$  du cercle :

*Pour tout entier  $R > 0$ , il existe un entier  $k > 0$  tel que,*

$$(k - 1)^2 < R \leq k^2$$

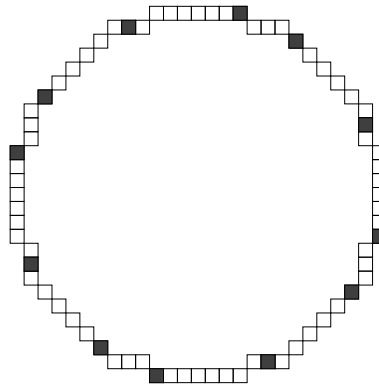
*alors, la longueur maximale des paliers du cercle de Bresenham de rayon  $R$  est égale à  $2k - 1$ .*

Exemple : Si  $R = 13$ , on a  $9 < R \leq 16$  donc  $k = 4$  et la longueur maximale de palier est égale à  $2k - 1 = 7$ .



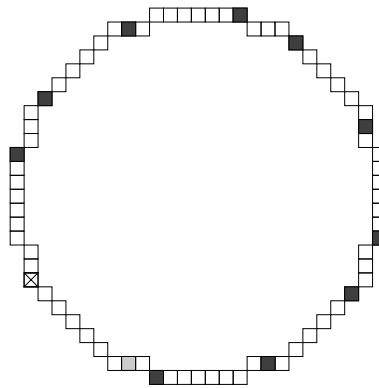
**Figure 3.15**

- Nous avons aussi remarqué que la polygonalisation d'un cercle dépend du point de départ choisi, et nous voyons apparaître des dynamiques particulières, regardons par exemple l'exemple du cercle de rayon  $R = 13$  sur la figure 3.16.



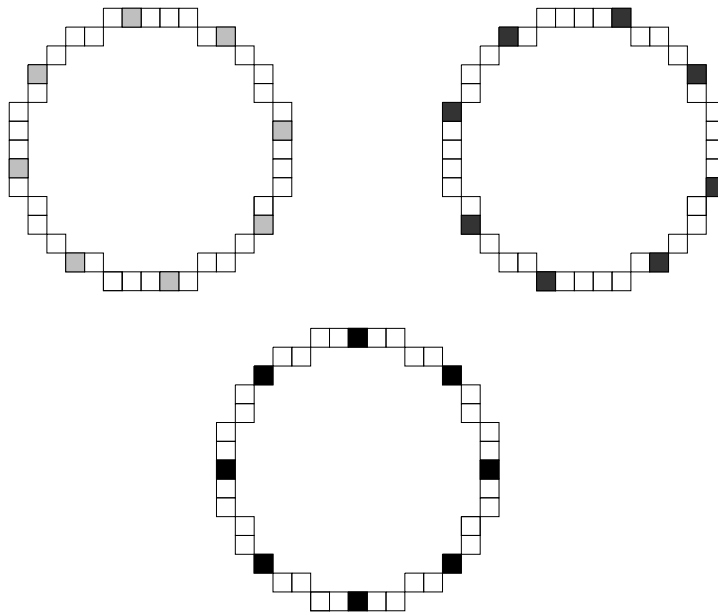
**Figure 3.16 :** *Cycle du cercle de rayon 13.*

La segmentation obtenue est optimale, nous l'appelons un *cycle*, les segments sont les plus grands possibles et le dernier segment a pour dernier point le point de départ de la polygonalisation. Si un autre point que ceux indiqués en noir est choisi comme point de départ, la segmentation "revient" dans ce cycle :



**Figure 3.17 :** *Le pixel de départ est en gris clair, le pixel extrémité d'un segment de cette polygonalisation mais n'appartenant pas au cycle est marqué d'une croix et enfin les pixels noirs sont les pixels du cycle du cercle de rayon  $R = 13$  qui sont des extrémités des segments de cette polygonalisation.*

Pour  $R = 13$  il y a un seul cycle mais ce n'est pas une généralité, par exemple pour  $R = 7$ , le cercle possède trois cycles (cf. Figure 3.17).



**Figure 3.18** : *Cycles du cercle de rayon 7.*

Si un des points du cercle n'appartenant pas à un de ces trois cycles est le pixel de départ de la polygonalisation, l'extrémité du premier segment trouvé sera dans un des trois cycles.

Nous ne sommes pas allés au delà de ces remarques mais les phénomènes observés soulèvent des questions et demandent à être approfondis.

### 3.4 La reconnaissance des segments de droites et les arbres de Stern-Brocot

La théorie des nombres, en particulier les fractions continues et les séries de Farey sont en rapport étroit avec les droites discrètes et leurs pentes [10, 50], c'est pourquoi il nous a semblé intéressant de regarder dans ce cadre le problème de la reconnaissance de segments de droites.



### 3.4.1 Définition et propriétés de l'arbre de Stern-Brocot

Une méthode pour engendrer toutes les fractions irréductibles positives  $\frac{m}{n}$  est proposée dans l'ouvrage d'Hardy et Wright [25], la structure qui en résulte est appelée l'*arbre de Stern-Brocot*.

L'idée est de démarrer avec les deux fractions  $\frac{0}{1}$ ,  $\frac{1}{0}$  et ensuite de répéter l'opération suivante autant de fois qu'on le désire :

Insérer  $\frac{m+m'}{n+n'}$  entre deux fractions adjacentes  $\frac{m}{n}$  et  $\frac{m'}{n'}$

La nouvelle fraction  $\frac{m+m'}{n+n'}$  est appelée la **médiane** de  $\frac{m}{n}$  et de  $\frac{m'}{n'}$ . Par exemple, entre  $\frac{0}{1}$  et  $\frac{1}{0}$  on construit  $\frac{1}{1}$  ainsi, on a la suite  $(\frac{0}{1}, \frac{1}{1}, \frac{1}{0})$  à la prochaine étape, deux fractions sont construites et on obtient  $(\frac{0}{1}, \frac{1}{2}, \frac{1}{1}, \frac{2}{1}, \frac{1}{0})$ , ensuite quatre fractions sont insérées  $(\frac{0}{1}, \frac{1}{3}, \frac{1}{2}, \frac{2}{3}, \frac{1}{1}, \frac{3}{2}, \frac{2}{1}, \frac{3}{1}, \frac{1}{0})$ , et ainsi de suite ...

Cette construction peut être vue comme un arbre infini dont une partie est représentée sur la figure 3.19.

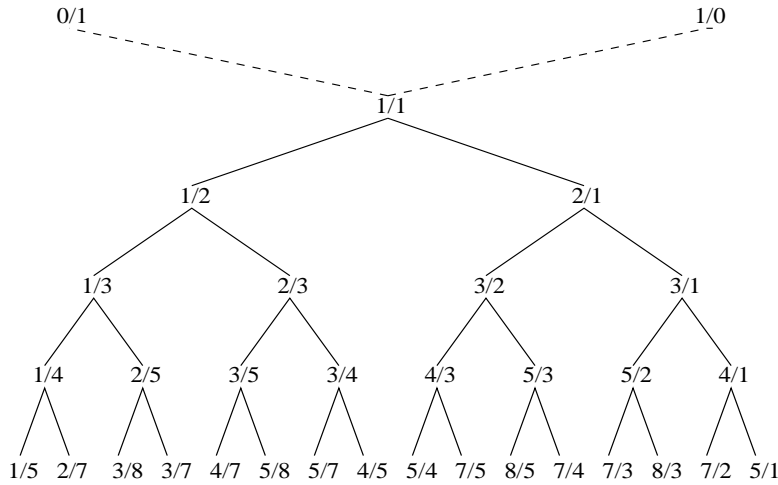


Figure 3.19 : Début de la construction de l'arbre de Stern-Brocot.

Chaque fraction  $\frac{m+m'}{n+n'}$  de l'arbre est telle que

- $\frac{m}{n}$  soit son plus proche ancêtre droit,
- $\frac{m'}{n'}$  soit son plus proche ancêtre gauche.

Cet arbre possède la propriété suivante :

Pour chaque niveau de l'arbre, si  $\frac{m}{n}$  et  $\frac{m'}{n'}$  sont deux fractions consécutives, alors  $m'n - n'm = 1$ .

Chaque noeud de l'arbre peut être représenté par une suite de déplacements :

$$R^{q_0} L^{q_1} R^{q_2} L^{q_3} \dots R^{q_{n-1}} L^{q_n}$$

avec,

- R un déplacement vers le fils droit,
- L un déplacement vers le fils gauche,
- $q_0 \geq 0$  et  $q_i > 0$  pour tout  $i > 0$ ,

en partant du noeud  $\frac{1}{1}$ .

Par exemple, le noeud  $\frac{3}{7}$  est représenté par les déplacements  $L^2 R^2$ ;  $q_0 = 0$ ,  $q_1 = 2$ ,  $q_2 = 2$ .

De plus, un noeud caractérisé par les déplacements  $R^{q_0} L^{q_1} R^{q_2} L^{q_3} \dots R^{q_{n-1}} L^{q_n}$  a la représentation en fraction continue,

$$q_0 + \frac{1}{q_1 + \frac{1}{q_2 + \frac{1}{\dots + \frac{1}{q_n + \frac{1}{1}}}}} = [q_0; q_1, q_2, \dots, q_n + 1]$$

Remarquons aussi que  $[q_0; q_1, q_2, \dots, q_n + 1] = [q_0; q_1, q_2, \dots, q_n, 1]$ .

Par exemple,

$$\frac{3}{7} = 0 + \frac{1}{2 + \frac{1}{2 + \frac{1}{1}}} = [0; 2, 3] = [0; 2, 2, 1]$$

### 3.4.2 Lien avec l'algorithme de reconnaissance de segments de droites discrètes

Après plusieurs expérimentations, nous avons remarqué que les points d'articulations  $(b_i, a_i)$  obtenus au cours de la reconnaissance d'un segment de droite de borne inférieure  $\mu = 0$  et de pente  $\frac{a}{b}$  avec  $a < b$ , correspondent à un **plongement géométrique des noeuds  $\frac{a_i}{b_i}$  de l'arbre de Stern-Brocot** ordonnés dans un parcours précis.

En effet, soient deux points d'articulation  $(b_i, a_i)$  et  $(b_{i+1}, a_{i+1})$  obtenus de manière

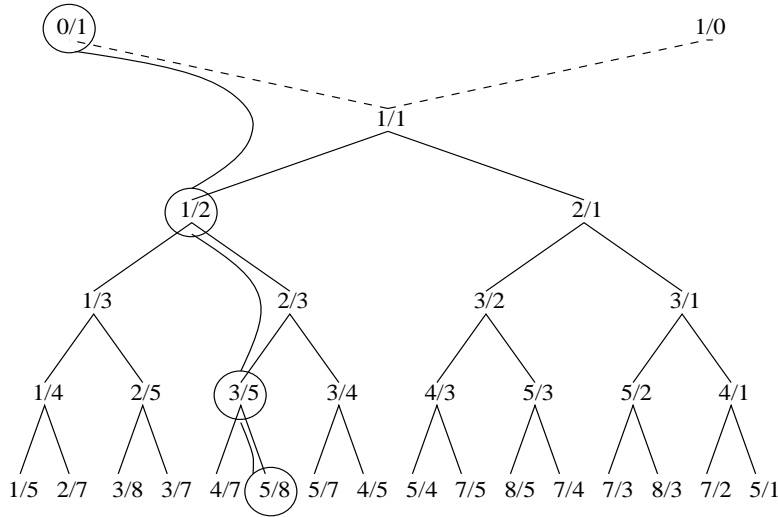
consécutives dans l'algorithme alors, on passe du noeud  $\frac{a_i}{b_i}$  au noeud  $\frac{a_{i+1}}{b_{i+1}}$  dans l'arbre de Stern-Brocot par un parcours

$$R^1 L^j \text{ avec un entier } j \geq 0.$$

Exemple :

Les points d'articulation de la droite de pente  $\frac{5}{8}$  et de borne inférieure  $\mu = 0$  sont  $(2, 1)$ ,  $(5, 3)$  et  $(8, 5)$  or, dans l'arbre on passe

- du noeud  $\frac{0}{1}$  au noeud  $\frac{1}{2}$  par les déplacements  $R^1 L^1$
- du noeud  $\frac{1}{2}$  au noeud  $\frac{3}{5}$  par les déplacements  $R^1 L^1$
- du noeud  $\frac{3}{5}$  au noeud  $\frac{5}{8}$  par les déplacements  $R^1$



**Figure 3.20 :** Déplacements dans l'arbre représentant les points d'articulation de la droite de pente  $\frac{5}{8}$ .

Soit  $C = R^{q_0+1} L^{q_1} R^{q_2} L^{q_3} \dots R^{q_{n-1}} L^{q_n}$  le chemin correspondant au noeud  $\frac{a}{b}$  de l'arbre de Stern-Brocot en partant du noeud  $\frac{0}{1}$ . Comme on suppose  $a < b$ , on a  $q_0 = 0$ .

C peut aussi s'écrire,

$$R^1 L^{q_1} \underbrace{R \dots R}_{q_2 - 1 \text{ fois}} L^{q_3} \underbrace{R \dots R}_{q_4 - 1 \text{ fois}} \dots \underbrace{R \dots R}_{q_n - 1 \text{ fois}} L^{q_n}$$

On obtient le résultat suivant :

Les points d'articulation, plongements géométriques des noeuds de l'arbre atteints par les déplacements du type  $R^1L^j$  avec un entier  $j \geq 0$  sont les points entiers de l'enveloppe convexe des points situés sous la droite réelle de pente  $\frac{a}{b}$ .

En effet, rappelons le théorème de Klein :

Soit  $r$  un réel, ayant le développement en fraction continue suivant

$$r = [q_0; q_1, q_2, q_3, \dots]$$

on considère les suites  $u_n$  et  $v_n$  telles que

$$\frac{u_n}{v_n} = [q_0; q_1, q_2, q_3, \dots, q_n]$$

alors,

– Les sommets de l'enveloppe convexe des points entiers situés en dessous de la droite de pente  $r$  sont :

$$(1, q_0), (v_2, u_2), \dots, (v_{2k}, u_{2k}).$$

– Les sommets de l'enveloppe convexe des points entiers situés au dessus de la droite de pente  $r$  sont :

$$(v_3, u_3), \dots, (v_{2k+1}, u_{2k+1}).$$

Ici,

$$\frac{a}{b} = [q_0; q_1, q_2, \dots, q_n + 1].$$

Les noeuds atteints par les déplacements  $R^1L^j$  sont représentés par

$RL^{q_1}$  qui a pour plongement dans  $\mathbb{Z}^2$  le point  $(b_1, a_1)$

$R \downarrow$

$RL^{q_1}R$  qui a pour plongement dans  $\mathbb{Z}^2$  le point  $(b_2, a_2)$

$R \downarrow$

$RL^{q_1}R^2$  qui a pour plongement dans  $\mathbb{Z}^2$  le point  $(b_3, a_3)$

$R \downarrow$

...

$R \downarrow$

$RL^{q_1}R^{q_2-1}$  qui a pour plongement dans  $\mathbb{Z}^2$  le point  $(b_{q_2}, a_{q_2})$

$RL^{q_3} \downarrow$

$RL^{q_1}R^{q_2}L^{q_3}$  qui a pour plongement dans  $\mathbb{Z}^2$  le point  $(b_{q_2+1}, a_{q_2+1})$

etc...

Or, le noeud  $RL^{q_1}R^{q_2-1}$  a un développement en fraction continue égale à  $[q_0; q_1, q_2 - 1 + 1] = [q_0; q_1, q_2] = \frac{a_{q_2}}{b_{q_2}}$  et, selon le théorème de Klein, le point  $(b_{q_2}, a_{q_2})$  est un sommet de l'enveloppe convexe des points situés sous la droite de pente  $\frac{a}{b}$ . Les noeuds le précédant, de  $RL^{q_1}$  à  $RL^{q_1}R^{q_2-2}$  ont pour développement en fraction continue  $[q_0; q_1, i]$  avec  $i$  variant de 1 à  $q_2 - 1$ . Les points représentant leurs plongements dans  $\mathbb{Z}^2$  sont sur l'enveloppe convexe des points situés sous la droite de pente  $\frac{a}{b}$ .

Un raisonnement semblable peut être fait pour tous les noeuds  $RL^{q_1} \dots R^{q_k-1}$ . ■

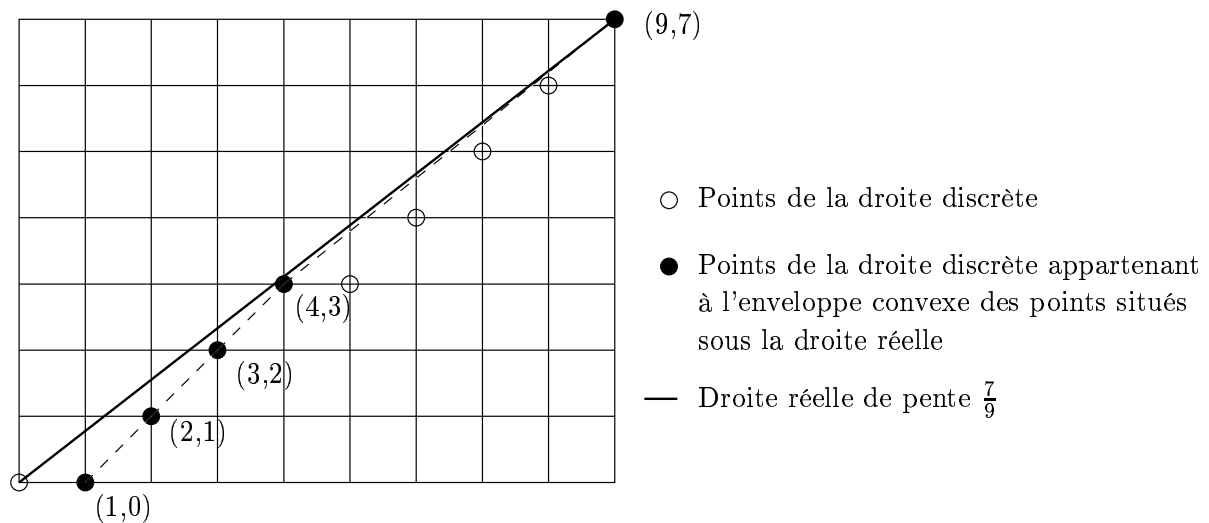
Exemple :

Le noeud  $\frac{7}{9}$  est obtenu avec le chemin  $R^1L^1R^3L^1$  donc,

$$\frac{7}{9} = [0; 1, 3, 2]$$

Les noeuds atteints par le parcours sont ,

- $RL^1$  de fraction continue  $[0; 1, 1] = \frac{1}{2}$  qui a pour plongement dans  $\mathbb{Z}^2$  le point  $(2, 1)$ .
- $RL^1R$  de fraction continue  $[0; 1, 2] = \frac{2}{3}$  qui a pour plongement dans  $\mathbb{Z}^2$  le point  $(3, 2)$ .
- $RL^1R^2$  de fraction continue  $[0; 1, 3] = \frac{3}{4}$  qui a pour plongement dans  $\mathbb{Z}^2$  le point  $(4, 3)$  qui est un sommet de l'enveloppe convexe des points situés sous la droite réelle de pente  $\frac{7}{9}$ .
- $RL^1R^3L$  de fraction continue  $[0; 1, 3, 2] = \frac{7}{9}$  qui a pour plongement dans  $\mathbb{Z}^2$  le point  $(9, 7)$  qui est un sommet de l'enveloppe convexe.



**Figure 3.21**

Ce parcours jusqu'à un noeud  $\frac{a}{b}$  avec  $a < b$  et  $\frac{a}{b} = [0; q_1, q_2, \dots, q_n]$  peut être exprimé de manière entièrement géométrique par les vecteurs  $V_i$  définis par :

$$\begin{aligned} V_0 &= (0, 1) \\ V_1 &= (1, 0) \\ V_2 &= q_1 V_1 + V_0 \\ V_3 &= q_2 V_2 + V_1 \\ &\dots \\ V_k &= q_{k-1} V_{k-1} + V_{k-2} \\ &\dots \end{aligned}$$

Les points touchés par le parcours sont alors les  $P_j$  tels que :

$$\begin{aligned} P_0 &= V_1 \\ P_1 &= P_0 + V_2; \text{ Pour } i \text{ de } 1 \text{ à } q_2 - 1 \text{ répéter } P_{i+1} = P_i + V_2 \\ P_{1+q_2} &= P_{q_2} + V_4; \text{ Pour } i \text{ de } 1 \text{ à } q_4 - 1 \text{ répéter } P_{1+q_2+i} = P_{q_2+i} + V_4 \\ P_{1+q_2+q_4} &= P_{q_2+q_4} + V_6; \text{ Pour } i \text{ de } 1 \text{ à } q_6 - 1 \text{ répéter } P_{1+q_2+q_4+i} = P_{q_2+q_4+i} + V_6 \\ &\dots \\ P_{1+q_2+q_4+\dots+q_{2k}} &= P_{q_2+q_4+\dots+q_{2k}} + V_{2(k+1)} \\ &\dots \end{aligned}$$

Exemple :

$$\frac{7}{9} = [0; 1, 3, 2] \text{ donc, } q_1 = 1, q_2 = 3, q_3 = 1.$$

$$\begin{aligned} V_0 &= (0, 1) \\ V_1 &= (1, 0) \\ V_2 &= V_1 + V_0 \\ V_3 &= 3V_2 + V_1 \\ V_4 &= V_3 + V_2 \\ \text{et,} \\ P_0 &= (1, 0) \\ P_1 &= (2, 1) \\ P_2 &= (3, 2) \\ P_3 &= (4, 3) \\ P_4 &= (9, 7) \end{aligned}$$

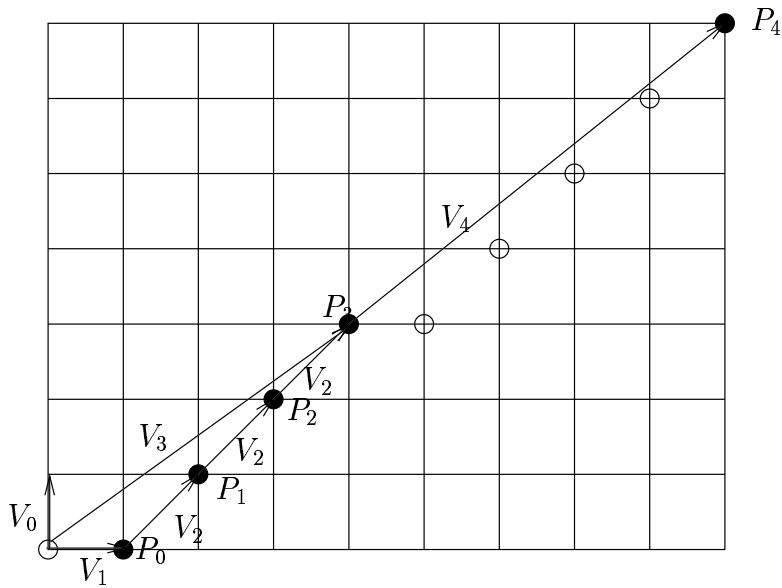


Figure 3.22

Une autre propriété importante liée aux arbres de Stern-Brocot concerne les droites discrètes contenant un segment de droite discrète donné de pente  $\frac{a}{b}$  :

*Si le noeud  $\frac{c}{d}$  se situe sur une branche partant du fils droit du noeud  $\frac{a}{b}$  alors un segment reconnu de droite discrète de pente  $\frac{a}{b}$  et de borne inférieure  $\mu = 0$  appartient à la droite discrète de pente  $\frac{c}{d}$  et de borne inférieure  $\mu = 0$ .*

En effet, le noeud  $\frac{a}{b}$  a pour plongement dans  $\mathbb{Z}^2$  le point  $(b, a)$  qui est un point d'articulation obligé dans la reconnaissance d'un segment reconnu de la droite discrète de pente  $\frac{c}{d}$  et de borne inférieure  $\mu = 0$ . Par conséquent, le segment reconnu de pente  $\frac{a}{b}$  appartient à la droite discrète de pente  $\frac{c}{d}$ . ■

### 3.5 Conclusion

L'algorithme de polygonalisation que nous proposons parcourt une seule fois la chaîne de longueur  $n$  représentant la courbe. Il est très simple et efficace, le critère de décision repose sur quatre tests et les caractéristiques du segment, recalculées à partir d'un nouvel élément ajouté, s'obtiennent immédiatement, il a une complexité en  $O(n)$ . Cet algorithme fournit des segments de droites discrètes de longueurs maximales en partant d'un point donné. Les difficultés rencontrées par la plupart des auteurs venaient des extrémités des segments, ici ce problème disparaît totalement ;

la structure de la droite est entièrement contrôlée par sa définition. Des algorithmes simples de tracé de droites discrètes naïves (cf. [50]) permettent à partir des caractéristiques des segments trouvés de reconstruire la courbe initiale.

L'algorithme de reconnaissance de segments de droites discrètes peut être étendu aux droites épaisses et aux droites naïves 3D ; ces applications sont exposées dans les deux chapitres suivants. De plus, il joue un rôle important dans la reconnaissance des morceaux de plans discrets.

Nous remercions R. Seroul pour l'intérêt qu'il a manifesté et les commentaires dont il nous a fait bénéficier à propos de cet algorithme.





## Chapitre 4

# Reconnaissance des droites connexes d'épaisseurs quelconques

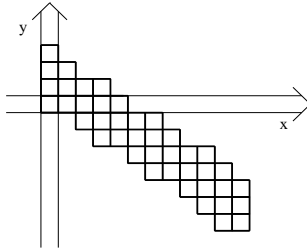
## 4.1 Introduction

Les courbes discrètes à segmenter ne sont pas forcément 8-connexes dans la pratique (traitement d'images, calligraphie). Par conséquent, il est intéressant d'étendre l'algorithme de reconnaissance exposé dans le chapitre précédent à des segments de différentes connexités. A cet effet, nous étudions les droites discrètes connexes non naïves.

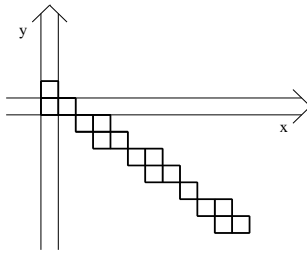
Une **droite discrète connexe non naïve**, notée  $\mathcal{D}(a, b, \mu, \omega)$ , est une droite discrète d'équation  $\mu \leq ax + by < \mu + \omega$  dont l'épaisseur  $\omega$  vérifie  $\omega > \sup(|a|, |b|)$ .

Parmi les droites connexes non naïves nous distinguons, selon leur topologie, trois catégories [50] :

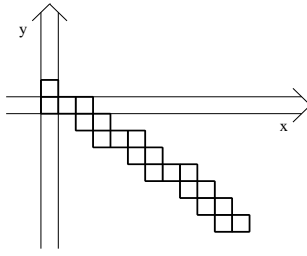
- les **droites \*-connexes** qui sont 8-connexes pour lesquelles l'épaisseur  $\omega$  vérifie  $\sup(|a|, |b|) < \omega < |a| + |b|$ ,
- les **droites dites 4-connexes** où  $\omega = |a| + |b|$ , cette épaisseur est la plus petite pour laquelle la droite est 4-connexe,
- les **droites épaisses** où  $\omega > |a| + |b|$ , elles sont 4-connexes.



**Figure 4.1 :** Segment de la droite épaisse  $\mathcal{D}(7, 10, 0, 34)$  d'équation  $0 \leq 7x + 10y < 34$ , pour  $x \in [0, 1]$ .



**Figure 4.2 :** Segment de la droite \*-connexe  $\mathcal{D}(7, 10, 0, 13)$  d'équation  $0 \leq 7x + 10y < 13$ , pour  $x \in [0, 1]$ .



**Figure 4.3 :** Segment de la droite 4-connexe  $\mathcal{D}(7, 10, 0, 17)$  d'équation  $0 \leq 7x + 10y < 17$ , pour  $x \in [0, 1]$ .

Dans la première section de ce chapitre, un algorithme incrémental de reconnaissance des segments de droites 4-connexes du septième octant et un algorithme linéaire de segmentation sont présentés ; ils constituent une extension immédiate des algorithmes donnés dans le chapitre précédent. Par contre, dans la section suivante, nous exposons les difficultés rencontrées pour reconnaître des segments d'épaisseurs quelconques et proposons un algorithme de reconnaissance de segments de droites connexes non naïves du septième octant d'épaisseurs quelconques. Cet algorithme repose sur l'analyse des différentes structures qui composent un segment de droite connexe non naïve d'épaisseur quelconque.

## 4.2 Reconnaissance des segments de droites 4-connexes et segmentation de courbes 4-connexes

Nous nous intéressons ici aux droites discrètes 4-connexes situées dans le septième octant, c'est à dire dont les points de coordonnées  $(x, y)$  vérifient  $\mu \leq ax + by < \mu + a + b$  avec  $0 \leq a \leq b$  et  $(a, b, \mu) \in \mathbb{Z}^3$ . Les notions de points d'appui, points faiblement et fortement extérieurs définies pour les droites 8-connexes s'étendent facilement aux droites 4-connexes.

Soit  $\mathcal{D}_e$  une droite 4-connexe de caractéristique  $a, b$  et  $\mu$  alors,

- les points d'appui inférieurs (resp. supérieurs) d'un segment de  $\mathcal{D}_e$ , ayant pour coordonnées  $(x, y)$ , vérifient  $ax + by = \mu$  (resp.  $ax + by = a + b - 1$ ). Le point d'appui principal inférieur (resp. supérieur) de  $\mathcal{D}_e$  est le point d'appui inférieur (resp. supérieur) possédant la plus petite abscisse.
- Les points faiblement extérieurs à  $\mathcal{D}_e$ , de coordonnées  $(x_{fs}, y_{fs})$ , situés au dessus de  $\mathcal{D}_e$  sont définis dans le septième octant par,

$$\begin{cases} x_{fs} = x_I + kb + 1 \\ y_{fs} = y_I - ka + 1 \end{cases} \quad k \in \mathbb{Z},$$

et les points faiblement extérieurs, de coordonnées  $(x_{fi}, y_{fi})$ , situés en dessous de  $\mathcal{D}_e$  sont définis par,

$$\begin{cases} x_{fi} = x_S + kb - 1 \\ y_{fi} = y_S - ka - 1. \end{cases} \quad k \in \mathbb{Z},$$

avec  $(x_I, y_I)$  et  $(x_S, y_S)$  les points d'appui principaux de  $\mathcal{D}_e$ .

- les points vérifiant l'une des deux conditions  $ax + by < \mu - 1$  ou  $ax + by > \mu + a + b$  sont dits fortement extérieurs à  $\mathcal{D}_e$ .

A partir de ces notions et en suivant le même raisonnement que dans le chapitre précédent, nous avons élaboré l'algorithme, donné ci-dessous, de reconnaissance de segments de droites discrètes 4-connexes. Une suite E de pixels 4-connexes est donnée en entrée et, à la fin du déroulement de l'algorithme, le booléen SEGMENT vaut vrai si E est un segment de droite discrète 4-connexe et faux sinon.

---

Algorithme de reconnaissance de segments de droites discrètes 4-connexes  
situés dans le septième octant

```
// initialisation // {cf. (i)}
M = premier point de E;
Si le point suivant M dans E a même abscisse alors
    a = 1; b = 1;  $\mu = 0$ ;
    S0 = M; l0 = point suivant de E;
    M = point suivant de E
Sinon
    a = 0; b = 1;  $\mu = 0$ ;
    S0 = M; l0 = M;
Fsi
SEGMENT=VRAI;
Tant que E n'a pas été entièrement parcouru et SEGMENT répéter
    M = point suivant de E
    rM = axM + byM;
    SEGMENT = rM ≥  $\mu - 1$  et rM ≤  $\mu + b + a$ ; {cf. (ii)}
    Si rM =  $\mu - 1$  ou rM =  $\mu + b + a$  alors
        Si rM =  $\mu - 1$  alors {cf. (iii)}
```

$$x_P = x_{I_0}; y_P = y_{I_0};$$

$$x_{S_0} = x_M - b + 1; y_{S_0} = y_M - a - 1;$$

**Fsi**

**Si**  $r_M = \mu + b + a$  **alors** *{cf. (iv)}*

$$x_P = x_{S_0}; y_P = y_{S_0};$$

$$x_{I_0} = x_M - b - 1; y_{I_0} = y_M - a + 1;$$

**Fsi**

$$b = x_M - x_P; a = -y_M + y_P;$$

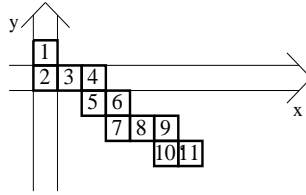
$$\mu = ax_{I_0} + by_{I_0};$$

**Fsi**

**Fin répéter**

**Un exemple de déroulement de l'algorithme de reconnaissance de segments de droites 4-connexes :**

Considérons la suite de pixels  $E$  de la figure 4.4 à analyser.



**Figure 4.4 :** Suite de pixels  $E$ , les pixels sont pondérés par leur numéro d'ordre dans la suite.

Après l'initialisation, nous avons  $a = 1$ ,  $b = 1$ ,  $\mu = 0$ ,  $S_0 = (0, 1)$  et  $I_0 = (0, 0)$ . La reconnaissance se poursuit en considérant la droite  $\mathcal{D}_{e_0}$  d'équation  $0 \leq x + y < 2$ .

Le parcours de la suite de points continue :

- au point  $(1, 0)$ ,  $x + y = 1$  donc  $(1, 0) \in \mathcal{D}_{e_0}$
- au point  $(2, 0)$ ,  $x + y = 2$  donc  $(2, 0) \notin \mathcal{D}_{e_0}$

De nouvelles caractéristiques sont calculées, le point  $(2, 0)$  est faiblement extérieur à  $\mathcal{D}_{e_0}$ , situé au dessus de  $\mathcal{D}_{e_0}$ . En suivant l'algorithme, on obtient :  $x_P = x_{S_0} = 0$ ,  $y_P = y_{S_0} = 1$ ,  $x_{I_0} = x_M - b - 1 = 0$  et  $y_{I_0} = y_M - a + 1 = 0$ . Par conséquent,  $b = 2$ ,  $a = 1$  et  $\mu = 0$ . Nous obtenons la représentation par restes donnée sur la figure 4.5 en fonction des caractéristiques obtenues pour les points déjà parcourus.

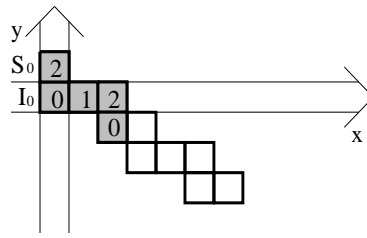


Figure 4.5

La reconnaissance se poursuit en fonction de la droite  $\mathcal{D}e_1$  d'équation  $0 \leq x + 2y < 3$  en ajoutant,

- le point  $(2, -1)$ ,  $x + 2y = 0$ ,  $(2, -1) \in \mathcal{D}e_1$ ,
- le point  $(3, -1)$ ,  $x + 2y = 1$ ,  $(3, -1) \in \mathcal{D}e_1$ ,
- le point  $(3, -2)$ ,  $x + 2y = -1$ ,  $(3, -2) \notin \mathcal{D}e_1$ .

Le point  $(3, -2)$  est faiblement extérieur situé en dessous de  $\mathcal{D}e_1$ ; de nouvelles caractéristiques sont calculées :  $x_P = x_{I_0} = 0$ ,  $y_P = y_{I_0} = 1$ ,  $x_{S_0} = x_M - b + 1 = 2$  et  $y_{S_0} = y_M - a - 1 = 0$ . Par conséquent,  $b = 3$ ,  $a = 2$  et  $\mu = 0$ . Nous obtenons sur la figure 4.6 la représentation par restes en fonction des caractéristiques obtenues pour les points déjà parcourus.

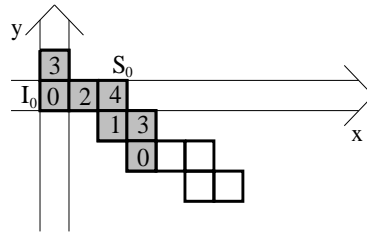


Figure 4.6

Les points suivants de  $E$ , comme nous pouvons le constater sur la figure 4.7 appartiennent à la droite  $\mathcal{D}e_2$  de caractéristiques  $b = 3$ ,  $a = 2$ ,  $\mu = 0$  et  $\omega = 5$ , par conséquent,  $E$  est un segment de la droite  $\mathcal{D}e_2$ .

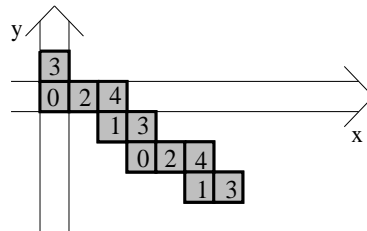
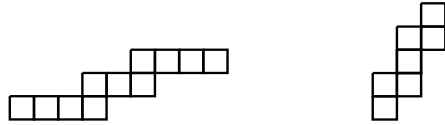


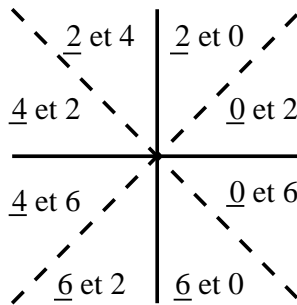
Figure 4.7

Pour coder une courbe 4-connexe, seules 4 directions du code de Freeman sont nécessaires; les directions 0, 2, 4 et 6. De plus, pour coder un segment de droite 4-connexe, deux directions parmi celles-ci interviennent dont une est dominante et l'autre sert à changer de palier.



**Figure 4.8 :** De gauche à droite, un segment se codant avec les directions 0 et 2 où 0 est la direction dominante, puis un segment se codant avec les directions 0 et 2 où 2 est la direction dominante.

Sur la figure 4.9 nous observons les directions codant un segment de droite 4-connexe selon l'octant où il se trouve, les directions dominantes sont soulignées.



**Figure 4.9**

Un algorithme linéaire de segmentation de courbes 4-connexes, similaire à celui présenté dans le chapitre précédent pour les courbes 8-connexes, se déduit simplement de l'algorithme de reconnaissance de segments de droites 4-connexes et des remarques précédentes concernant le codage d'un segment de droite 4-connexe avec le code de Freeman.



## 4.3 Reconnaissance de segments de droites connexes non naïves d'épaisseurs quelconques

### 4.3.1 Décomposition d'une droite connexe non naïve

Une droite discrète connexe non naïve est une réunion de pointillés (points entiers  $(x, y)$  vérifiant  $ax + by = \alpha$  pour  $\alpha \in [\mu, \mu + \omega]$ ) et peut être décomposée en plusieurs éléments.

Soit  $\mathcal{D}(a, b, \mu, \omega)$  une droite discrète connexe non naïve, on appelle

- **droite inférieure de  $\mathcal{D}$** , la droite discrète dont les points vérifient l'équation

$$\mu \leq ax + by < \mu + \text{sup}(|a|, |b|),$$

- **droites internes de  $\mathcal{D}$** , les droites discrètes dont les points vérifient les équations

$$\mu + k\text{sup}(|a|, |b|) \leq ax + by < \mu + (k + 1)\text{sup}(|a|, |b|),$$

pour  $k$  variant de 1 à  $\lceil \frac{\omega}{\text{sup}(|a|, |b|)} \rceil - 1$ .

- **droite externe de  $\mathcal{D}$** , la droite dont les points vérifient l'équation

$$\mu + \lceil \frac{\omega}{\text{sup}(|a|, |b|)} \rceil \text{sup}(|a|, |b|) \leq ax + by < \mu + \omega.$$

Remarques :

- Les droites internes existent si  $\lceil \frac{\omega}{\text{sup}(|a|, |b|)} \rceil > 1$ .
- La droite externe est vide s'il existe un entier  $k > 1$  tel que  $\omega = k\text{sup}(|a|, |b|)$ .

Exemple :

Un segment de la droite épaisse  $\mathcal{D}(7, 10, 0, 34)$  est décomposé sur la figure 4.10 où nous voyons la représentation par restes d'un segment de  $\mathcal{D}$  (la valeur  $ax + by$  pondère chaque point de coordonnées  $(x, y)$  de  $\mathcal{D}$ ).

Plusieurs types de *parcours* peuvent être réalisés dans un segment de droite discrète d'épaisseur quelconque. Nous avons choisi, pour faciliter l'algorithme de reconnaissance, de parcourir d'abord la droite naïve inférieure puis, si elles existent, les droites internes et enfin la droite externe.



En appliquant l'algorithme de reconnaissance de segments de droites naïves sur la courbe 8-connexe inférieure de  $S$  (marquée en gris foncé), nous obtenons les caractéristiques  $a = 2$ ,  $b = 5$  et  $\mu = 0$ . En observant la représentation par restes en fonction de  $a$  et  $b$  sur la figure 4.12, nous constatons l'absence du point de coordonnées  $(6, -1)$ , de reste 7, pour que  $S$  soit un segment de la droite de caractéristiques  $a = 2$ ,  $b = 5$ ,  $\mu = 0$  et d'épaisseur  $\omega = 8$ .

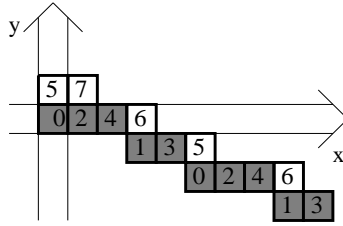


Figure 4.12

Après cette constatation, nous ne pouvons pas conclure sur la nature de  $S$ .

2. Considérons la courbe connexe  $S$  de la figure 4.13.

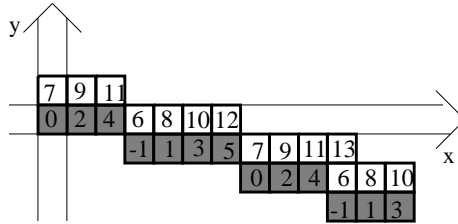


Figure 4.13

Le segment de la droite inférieure naïve reconnue par l'algorithme de reconnaissance de segments de droites naïves a comme caractéristiques  $a = 2$ ,  $b = 7$  et  $\mu = -1$ . Le point de coordonnées  $(10, -1)$  ayant pour reste 13 apparaît au dessus de la deuxième période de la droite naïve inférieure sans être présent dans la première ; nous ne pouvons donc pas conclure sur la nature de  $S$ .

Dans les deux exemples présentés ci-dessus, nous ne pouvons pas appliquer uniquement le principe utilisé dans la reconnaissance des droites naïves ou 4-connexes, en effet, l'épaisseur de segments à reconnaître est inconnue. Dans la sous-section suivante, nous montrons comment résoudre ce problème.

### 4.3.3 Un algorithme de reconnaissance de segments de droites connexes non naïves d'épaisseurs quelconques

Considérons un ensemble de pixels  $S$  à analyser, pour obtenir une équation de droite connexe non naïve correspondant à  $S$ , nous devons “anticiper” les changements de la droite naïve inférieure en fonction des pixels trouvés sur la droite externe. En effet la droite externe d'un segment de droite discrète nous informe sur les changements qui ont lieu dans les périodes suivantes non visibles au niveau de la droite naïve inférieure. Rappelons le résultat donné dans le chapitre précédent qui est utilisé dans l'algorithme exposé par la suite :

*Les points faiblement extérieurs à la droite naïve  $\mathcal{D}(a, b, \mu)$  de coordonnées  $(x_{fs}, y_{fs})$  situés au dessus de la droite naïve  $\mathcal{D}$  sont définis dans le septième octant par,*

$$\begin{cases} x_{fs} = x_I + kb \\ y_{fs} = y_I - ka + 1 \end{cases} \quad k \in \mathbb{Z},$$

*et les points faiblement extérieurs de coordonnées  $(x_{fi}, y_{fi})$  situés en dessous de  $\mathcal{D}$  sont définis par,*

$$\begin{cases} x_{fi} = x_S + kb \\ y_{fi} = y_S - ka - 1. \end{cases} \quad k \in \mathbb{Z},$$

*avec  $(x_I, y_I)$  et  $(x_S, y_S)$  les points d'appui principaux de  $\mathcal{D}$  vérifiant respectivement dans le septième octant  $ax_I + by_I = \mu$  et  $ax_S + by_S = \mu + b - 1$ .*

Les points faiblement extérieurs sont les seuls points de changements possibles en cas de modification de pente.

Soit  $S$  un ensemble connexe de pixels, situés dans le septième octant, l'algorithme de reconnaissance s'articule autour de trois étapes :

- La première consiste à isoler le code de Freeman de la courbe 8-connexe inférieure et tester avec l'algorithme de reconnaissance de segments de droites naïves si cette courbe est un segment de droite naïve, noté  $S_{inf}$ , si c'est le cas, les caractéristiques obtenues servent à initialiser celles du segment en cours de reconnaissance.
- Dans la seconde étape, les pixels des segments 8-connexes situés au dessus de  $S_{inf}$  sont parcourus parallèlement à  $S_{inf}$ . Le nombre de segments complets 8-connexes “empilés” au dessus de  $S_{inf}$  sert à déterminer l'épaisseur de l'éventuel segment de droite.
- L'ensemble des pixels déconnectés situé au dessus de l'ensemble déjà parcouru, noté  $S_{ext}$ , permet, dans la troisième étape, de modifier si nécessaire les

caractéristiques obtenues à l'étape 1. En effet, nous regardons les pixels de  $S_{ext}$  situés au dessus de la première période de la droite naïve de  $S_{inf}$ , ils doivent se retrouver périodiquement au dessus des autres périodes présentes dans  $S_{inf}$ . Si ce n'est pas le cas, ou si des pixels supplémentaires apparaissent, on en déduit un changement de pente non décelé sur  $S_{inf}$  ou une discontinuité. Si un point du dernier pointillé est manquant, le point de changement, de coordonnées  $(x_c, y_c)$ , est un point faiblement extérieur situé en dessous de la droite support de  $S_{inf}$ . Si un point du pointillé immédiatement supérieur au dernier pointillé apparaît en supplément, le point de changement est un point faiblement extérieur situé au dessus de la droite support de  $S_{inf}$ . A partir de ces points de changement, de nouvelles caractéristiques sont calculées et testées.

---

Algorithme de reconnaissance de segments de droites discrètes connexes non naïves situés dans le septième octant

Entrée :  $S$  un ensemble de pixels connexe, soit  $l$  l'abscisse maximale de cet ensemble et  $h$  l'épaisseur maximale de  $S$  ( $h = \max(|y_i - y_j|)$  pour tout couple de points  $(x, y_i)$  et  $(x, y_j)$  de  $S$ ).

Sortie : Faux si  $S$  n'est pas un segment de droite discrète et Vrai sinon (de plus, les caractéristiques du segment sont calculées).

Etape 1 : // Droite naïve inférieure //

Isoler puis reconnaître la courbe 8-connexe inférieure avec l'algorithme de reconnaissance de segments de droites naïves.

**Si** l'algorithme retourne Faux **alors**  $S$  n'est pas un segment de droite, retourner Faux.

**Sinon** l'algorithme reconnaît un segment de droite naïve noté  $S_{inf}$  de caractéristiques  $a, b, \mu$  avec les points d'appui principaux inférieurs et supérieurs ayant respectivement pour coordonnées  $(x_I, y_I)$  et  $(x_S, y_S)$ .

**Fsi**

Etape 2 : // Droites internes //

Parcourir les droites internes (segments 8-connexes parallèles à  $S_{inf}$ ) présentes dans  $S$ , s'arrêter dès qu'un point est manquant par rapport à ceux de  $S_{inf}$ , soit  $n$  le nombre de droites parcourues.

**Si**  $n = h$  **alors**  $S$  est un segment de la droite de caractéristiques  $a, b, \mu$  et d'épaisseur  $\omega = (h + 1)b$ , retourner Vrai.

**Sinon**

**Si**  $n \neq h - 1$  **alors**  $S$  n'est pas un segment de droite discrète, retourner Faux. **Fsi**

**Fsi**

Etape 3 : // Parcours et interprétation de la droite externe //

**Tant que  $b < l$  répéter**

1. Soit  $n_{ext}$  le nombre de points de  $S$  d'abscisse  $x \in [0, b[$ , qui n'appartiennent ni à  $S_{inf}$ , ni aux droites internes; ces points forment un ensemble déconnecté, noté  $S_{ext[0,b[}$ .

**Si** il existe un point de  $S_{ext[0,b[}$  de coordonnées  $(x, y)$  qui ne vérifie pas  $\mu + (n + 1)b \leq ax + by < \mu + (n + 1)b + n_{ext}$  **alors**  $S$  n'est pas un segment de droite discrète, retourner Faux.

**Fsi**

2. Soit  $S_{ext[b,l[}$  l'ensemble des points de  $S$  n'appartenant ni à  $S_{inf}$ , ni aux droites internes, ni à  $S_{ext[0,b[}$ . Posons  $\omega = (n + 1)b + n_{ext}$ .

**Si** exactement tous les points de  $S_{ext[0,b[}$  se retrouvent périodiquement dans  $S_{ext[b,l[}$  **alors**  $S$  est un segment de la droite de caractéristiques  $a, b, \mu$  et d'épaisseur  $\omega$ , retourner Vrai.

**Sinon** trois cas sont possibles :

– **Si** un point de coordonnées  $(x, y)$  tel que  $ax + by < \omega - 1$  manque ou tel que  $ax + by > \omega$  est en trop dans  $S_{ext[b,l[}$  **alors**  $S$  n'est pas un segment de droite, retourner Faux.

**Fsi**

– **Si** un point de coordonnées  $(x, y)$  tel que  $ax + by = \omega + \mu - 1$  manque dans  $S_{ext[b,l[}$  **alors**

$$x_c = x_S + ([l/b] + 1)b; y_c = y_S - ([l/b] + 1)a - 1;$$

$$a = -y_c + y_I; b = x_c - x_I; \mu = ax_c + by_c;$$

**Si**  $b > l$  **alors** soit  $d_{max}$  la valeur maximale prise par  $ax + by$  pour  $(x, y)$  un point de  $S$ ,  $S$  est un segment de la droite de caractéristiques  $a, b, \mu$  et d'épaisseur  $\omega = d_{max} + 1$ , retourner Vrai.

**Fsi**

**Fsi**

– **Si** un point de coordonnées  $(x, y)$  tel que  $ax + by = \omega + \mu$  est en supplément dans  $S_{ext[b,l[}$  **alors**

$$x_c = x_I + ([l/b] + 1)b; y_c = y_I - ([l/b] + 1)a + 1;$$

$$a = -y_c + y_S; b = x_c - x_S; \mu = ax_c + by_c - b + 1;$$

**Si**  $b > l$  **alors** soit  $d_{max}$  la valeur maximale prise par  $ax + by$  pour  $(x, y)$  un point de  $S$ ,  $S$  est un segment de la droite de caractéristiques  $a, b, \mu$  et d'épaisseur  $\omega = d_{max} + 1$ , retourner Vrai.

**Fsi**

**Fsi**

**Fsi**

**Fin répéter**

---

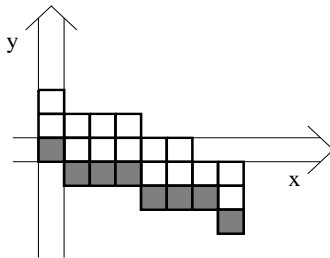
Remarques :

– Au plus deux itérations sont nécessaires dans l'étape 3 pour obtenir  $b < l$ .

- Segmenter des courbes connexes d'épaisseurs quelconques avec cet algorithme non incrémental, nécessite de faire tourner l'algorithme précédent autant de fois qu'il y a de points sur la courbe 8-connexe inférieure. A chaque fois qu'un point de la courbe 8-connexe est ajouté avec les points de la courbe situés au dessus de lui au segment courant, l'algorithme de reconnaissance de segments de droites d'épaisseurs quelconques doit reparcourir et reconnaître tout le segment plus le point.

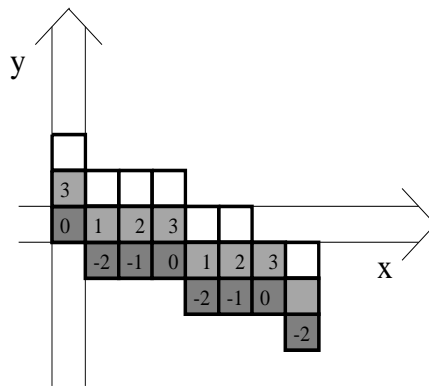
**Exemples de déroulement de l'algorithme de reconnaissance de segments de droites connexes d'épaisseurs quelconques :**

1. Considérons l'ensemble  $S$  de pixels de la figure 4.14. L'algorithme décrit ci-dessus parcourt dans un premier temps la courbe 8-connexe inférieure de  $S$  marquée en gris foncé sur la figure 4.14. L'algorithme de reconnaissance des segments de droites naïves reconnaît cette courbe qui est un segment de la droite naïve de caractéristiques  $a = 1$ ,  $b = 3$  et  $\mu = -2$  et les points d'appui principaux inférieur et supérieur ont respectivement pour coordonnées  $(1, -1)$  et  $(0, 0)$ .



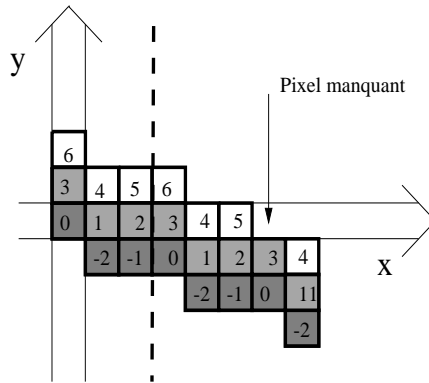
**Figure 4.14**

Un segment 8-connexe complet parallèle au segment précédemment reconnu est parcouru dans la deuxième étape de l'algorithme en calculant en chaque pixel  $(x, y)$  la valeur du reste  $x + 3y$ ,  $n = 1$  or  $h = 2$  par conséquent nous ne pouvons pas encore conclure.



**Figure 4.15 :** Chaque pixel de  $S$  parcouru jusqu'à l'étape 2 est pondéré par la valeur  $x + 3y$ .

Dans la troisième étape, l'ensemble de pixels déconnectés restant est étudié. Nous regardons d'abord les pixels situés au dessus de la première période du segment reconnu à l'étape 1. Trois points se trouvent dans cette période et ils vérifient  $4 \leq x + 3y < 7$  (cf. Figure 4.16 les pixels blancs),  $\omega = 9$ . Nous devons retrouver ces points au dessus des autres périodes, or, le point de coordonnées  $(6, 0)$  qui a pour reste 6 est manquant.



**Figure 4.16**

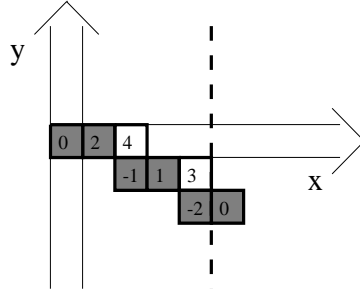
Le premier point de bifurcation possible non visible sur le segment présent dans  $S$  de  $\mathcal{D}(1, 3, -2)$  est un point faiblement extérieur situé en dessous de  $\mathcal{D}$ , il a pour coordonnées  $(9, -4)$  et ce point de changement engendre les caractéristiques  $a = 3$ ,  $b = 8$  et  $\mu = -5$ . La représentation par restes de  $S$  en fonction de ces caractéristiques est donnée sur la figure 4.17.





Nous pouvons constater sur la figure 4.19 qu'il n'y a pas de point situé au dessus de la première période du segment de la droite naïve  $\mathcal{D}(1, 2, 0)$  mais les points de coordonnées  $(2, 0)$  et  $(4, -1)$  ayant pour reste 2 apparaissent au dessus des périodes suivantes.

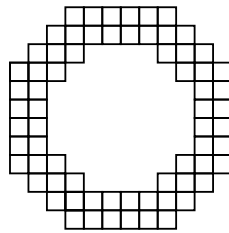
Le premier point de bifurcation possible non visible sur le segment présent dans  $S$  de  $\mathcal{D}(1, 2, 0)$  est un point faiblement extérieur situé au dessus de  $\mathcal{D}$ , il a pour coordonnées  $(6, -2)$  et ce point de changement engendre les caractéristiques  $a = 2, b = 5$  et  $\mu = -2$ . Voici la représentation par restes de  $S$  en fonction de ces caractéristiques :



**Figure 4.20**

Une période entière plus un point de  $\mathcal{D}(2, 5, -2)$  sont présents dans  $S$ , les points de  $S$  situés au dessus de  $\mathcal{D}$  vérifient  $3 \leq 2x + 5y < 5$ , par conséquent,  $S$  est un morceau de la droite épaisse de caractéristiques  $a = 2, b = 5, \mu = -2$  et d'épaisseur  $\omega = 7$ .

## 4.4 Conclusion



**Figure 4.21**

Nous avons présenté dans ce chapitre un algorithme linéaire de segmentation de courbes discrètes 4-connexes directement déduit de l'algorithme de segmentation de courbes 8-connexes présenté dans le chapitre précédent. De plus, un algorithme de reconnaissance de segments de droites connexes d'épaisseurs quelconques a été exposé.

La segmentation des courbes connexes d'épaisseurs quelconques demande encore du travail pour être performante ; il faut rendre incrémental l'algorithme présenté dans ce chapitre et ensuite définir une notion correcte de courbe discrète épaisse. En effet, dans l'exemple donné sur la figure 4.21, la courbe 8-connexe inférieure possède moins de pixels que la courbe 8-connexe supérieure, certains pixels ne peuvent pas être inclus dans des segments de droites.

# Chapitre 5

## Les droites discrètes 3D et la segmentation de courbes discrètes 26-connexes

## 5.1 Introduction

Comme la droite discrète 2D, une définition formelle d'une droite discrète 3D est nécessaire dans de nombreuses applications utilisant des objets formés de voxels [64, 32, 30, 31]. Par exemple, le tracé de droites 3D entre deux points de l'espace discret est primordial en lancer de rayon discret. B. Vidal [64], dans sa thèse, a fait la synthèse des différentes méthodes utilisées :

- les méthodes “par projections”, où sont calculées indépendamment deux projections du segment 3D sur deux plans de base (algorithme d'Amanatides et Woo [1])
- les méthodes “par calcul direct” obtenues par extension des algorithmes 2D en 3D (algorithme de Bresenham, DDA, algorithme de Loeff)

Plusieurs auteurs ont proposé des définitions de droites 3D 26-connexes [34, 59] et, comme en 2D, il nous a semblé intéressant de proposer une définition arithmétique générale de droite discrète 3D regroupant les différentes structures obtenues par les algorithmes de tracé connus.

Dans la première section de ce chapitre, nous donnons une définition arithmétique de droite 3D ainsi qu'un théorème de structure permettant de contrôler la connexité des droites discrètes 3D ainsi définies. De plus, un algorithme de parcours très simple est proposé, il est équivalent à ceux déjà connus utilisant des méthodes “par projections” et repose uniquement sur des calculs en nombres entiers. Ensuite, dans la deuxième section, un algorithme de reconnaissance de segments de droites naïves 3D est exposé ; il repose sur la propriété de fonctionnalité d'une droite naïve 3D et l'algorithme de reconnaissance de segments de droites naïves 2D donné dans le chapitre précédent. Enfin, un algorithme de segmentation de courbes discrètes 3D 26-connexes est proposé dans la dernière section ainsi que des exemples sur différentes courbes 26-connexes.

## 5.2 Définition et premières propriétés

### 5.2.1 La droite discrète 3D

Pour des facilités d'écriture, nous donnons une définition de droite discrète 3D ayant pour vecteur directeur  $(a, b, c) \in \mathbb{Z}^3$  avec  $a \geq b \geq c$ . Les définitions pour des coefficients différemment ordonnés s'obtiennent en permutant  $x, y, z$  ainsi que les coefficients. Nous supposons pour la suite que  $a \geq b \geq c$ , condition à laquelle il est facile de se ramener par symétries.

La **droite discrète 3D**, notée  $\mathcal{D}$ , de vecteur directeur  $V(a, b, c)$ , avec  $(a, b, c) \in \mathbb{Z}^3$ , tel que  $a \geq b \geq c$  est définie par l'ensemble des points  $(x, y, z)$  de  $\mathbb{Z}^3$  vérifiant les

inéquations diophantiennes suivantes :

$$\mathcal{D} \begin{cases} \mu \leq cx - az < \mu + e & (1) \\ \mu' \leq bx - ay < \mu' + e' & (2) \end{cases}$$

avec  $\mu, \mu', e, e' \in \mathbb{Z}$ .  $\mu$  et  $\mu'$  sont les *bornes inférieures* de  $\mathcal{D}$ , de plus,  $e$  et  $e'$  sont appelées les *épaisseurs arithmétiques* de  $\mathcal{D}$ . Une telle droite est notée  $\mathcal{D}(a, b, c, \mu, \mu', e, e')$ .

Remarques :

1. Pour un vecteur directeur donné, il existe un grand nombre de représentations différentes en fonction des bornes inférieures et des épaisseurs choisies
2. Dans le plan  $Oxz$  (resp.  $Oxy$ ), la double inéquation (1) (resp. (2)) représente une droite arithmétique 2D de borne inférieure  $\mu$  (resp.  $\mu'$ ) et d'épaisseur arithmétique  $e$  (resp.  $e'$ ).
3. Les doubles inéquations diophantiennes (1) et (2) sont les équations de deux plans discrets de  $\mathbb{Z}^3$  (cf. Chapitre 6).

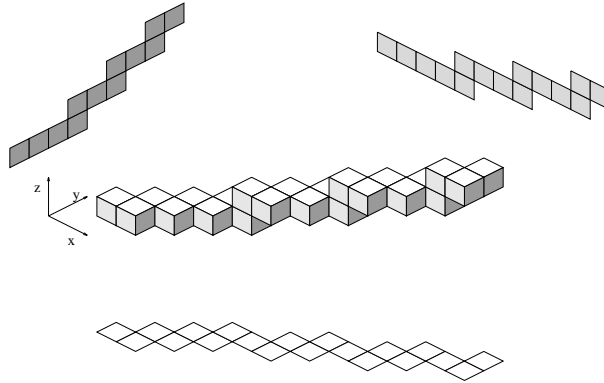


Figure 5.1 :  $\mathcal{D}(10, 7, 3, 0, -9, 13, 17)$ , droite 6-connexe,

$$\begin{cases} 0 \leq 3x - 10z < 13 \\ -9 \leq 7x - 10y < 8 \end{cases}$$

Nous appelons *droites naïves 3D*, les droites discrètes 3D dont les épaisseurs vérifient  $e = e' = a$ , elles sont notées  $\mathcal{D}(a, b, c, \mu, \mu')$ .

*Théorème 5.1 : Une droite naïve 3D est fonctionnelle.*

En effet, considérons la droite naïve  $\mathcal{D}(a, b, c, \mu, \mu', e)$ ,  $\mathcal{D}$  est définie par

$$\begin{cases} \mu \leq cx - az < \mu + e \\ \mu' \leq bx - ay < \mu' + e \end{cases}$$

ce qui est équivalent à

$$\begin{cases} z = \left\lfloor \frac{cx - \mu}{a} \right\rfloor \\ y = \left\lfloor \frac{bx - \mu'}{a} \right\rfloor \end{cases}$$

■

## 5.2.2 Relation entre les épaisseurs arithmétiques et la connexité de la droite discrète 3D

Pour mieux visualiser la notion de droite discrète 3D voici un théorème de structure qui établit le lien entre les épaisseurs arithmétiques et la connexité d'une droite discrète 3D.

*Théorème 5.2 :*

*Soit  $\mathcal{D}(a, b, c, \mu, \mu', e, e')$  avec  $a > b > c$ ,*

*Si  $e \geq a + c$  et  $e' \geq a + b$ ,  $\mathcal{D}$  est 6-connexe.*

*Si  $e \geq a + c$  et  $a \leq e' < a + b$  ou  $e' \geq a + b$  et  $a \leq e < a + c$ ,  $\mathcal{D}$  est 18-connexe.*

*Si  $a \leq e < a + c$  et  $a \leq e' < a + b$ ,  $\mathcal{D}$  est 26-connexe.*

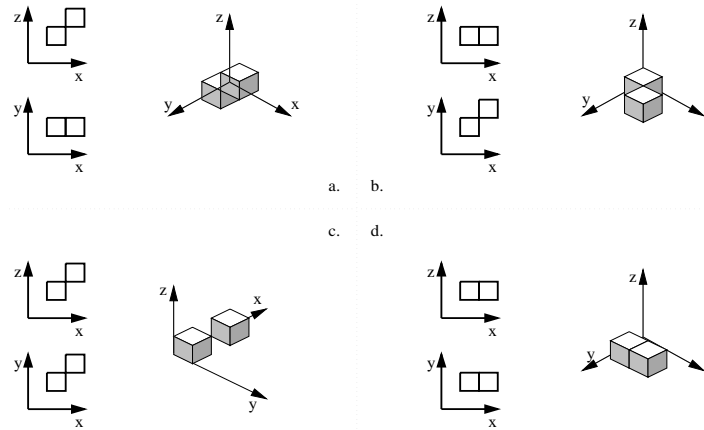
*Si  $e < a$  ou  $e' < a$ ,  $\mathcal{D}$  est déconnectée.*

Pour démontrer ces résultats, nous utilisons le fait que les projections de  $\mathcal{D}$  dans les plans  $Oxy$  et  $Oxz$  sont des droites discrètes 2D dont la connexité est connue [50].

Par exemple, considérons le troisième cas ;  $a \leq e < a + c$  et  $a \leq e' < a + b$ , dans le plan  $Oxz$ , le passage d'un pixel au suivant dans la droite discrète  $\mu \leq cx - az < \mu + e$  est fait par un saut 4-connexe ou 8-connexe, de même dans le plan  $Oxy$  pour la droite  $\mu' \leq bx - ay < \mu' + e'$ . Toutes les combinaisons possibles entre deux pixels des plans  $Oxy$  et  $Oxz$  pour engendrer deux voxels de  $\mathbb{Z}^3$  ont été étudiées sur la figure 5.2 :

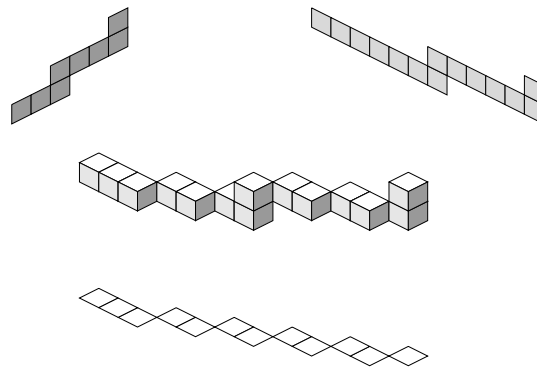
- a. un saut 8-connexe dans le plan  $Oxz$  et un saut 4-connexe dans le plan  $Oxy$  engendrent deux voxels liés par un saut 18-connexe,
- b. un saut 4-connexe dans le plan  $Oxz$  et un saut 8-connexe dans le plan  $Oxy$  engendrent deux voxels liés par un saut 18-connexe,

- c. un saut 8-connexe dans le plan  $Oxz$  et un saut 8-connexe dans le plan  $Oxy$  engendrent deux voxels liés par un saut 26-connexe,
  - d. un saut 4-connexe dans le plan  $Oxz$  et un saut 4-connexe dans le plan  $Oxy$  engendrent deux voxels liés par un saut 26-connexe.
- Ainsi, nous montrons que la droite  $\mathcal{D}$  est 26-connexe.



**Figure 5.2**

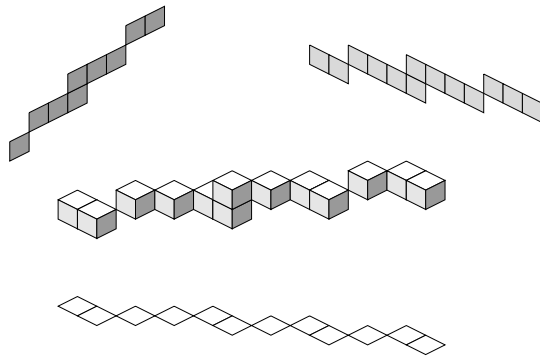
Les autres résultats se démontrent de manière similaire en regardant toutes les combinaisons de deux voxels obtenues à partir de celles possibles de deux pixels dans les plans  $Oxz$  et  $Oxy$ . ■



**Figure 5.3** :  $\mathcal{D}(11, 5, 2, 0, 0, 13, 11)$ , droite 18-connexe,

$$\begin{cases} 0 \leq 2x - 11z < 13 \\ 0 \leq 5x - 11y < 11 \end{cases}$$

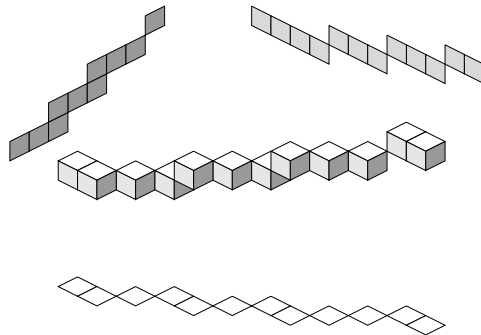




**Figure 5.4** :  $\mathcal{D}(10, 7, 3, -5, -2, 11, 10)$ , droite 26-connecte,

$$\begin{cases} -5 \leq 3x - 10z < 6 \\ -2 \leq 7x - 10y < 8 \end{cases}$$

Les droites naïves sont 26-connexes et le passage d'un palier au suivant se fait uniquement par une arête ou par un sommet, nous pouvons le constater sur les figures 5.5 et 5.6.



**Figure 5.5** :  $\mathcal{D}(10, 7, 3, 0, 0)$

$$\begin{cases} 0 \leq 3x - 10z < 10 \\ 0 \leq 7x - 10y < 10 \end{cases}$$

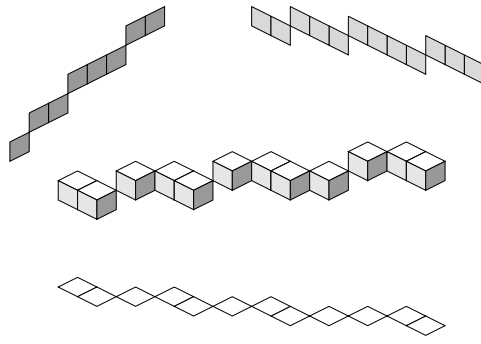


Figure 5.6 :  $\mathcal{D}(10, 7, 3, -5, 0)$

$$\begin{cases} -5 \leq 3x - 10z < 5 \\ 0 \leq 7x - 10y < 10 \end{cases}$$

### 5.2.3 Un algorithme de parcours de droites discrètes naïves 3D

Considérons  $A(x_A, y_A, z_A)$  et  $B(x_B, y_B, z_B)$  deux points de  $\mathbb{Z}^3$ . L'algorithme donné ci-dessous trace un segment de droite naïve entre les deux points  $A$  et  $B$  de vecteur directeur  $\vec{AB} = (v_x, v_y, v_z)$ . Selon le choix de la borne inférieure, des structures différentes sont obtenues. En particulier, pour  $\mu = \mu' = -\lceil \frac{\max(|v_x|, |v_y|, |v_z|)}{2} \rceil + 1$ , les voxels générés sont les mêmes que ceux définis par l'algorithme de Brésenham étendu au 3D.

Le principe de l'algorithme consiste à calculer parallèlement dans les plans  $Oxy$  et  $Oxz$  les pixels des droites naïves 2D qui sont les projections de la droite 3D dans ces deux plans. Nous reconstituons ainsi les trois coordonnées  $(x, y, z)$  du voxel correspondant.

---

Algorithme de construction d'un segment de droite discrète naïve de bornes inférieures  $\mu$  et  $\mu'$  entre deux points  $A(x_A, y_A, z_A)$  et  $B(x_B, y_B, z_B)$  de  $\mathbb{Z}^3$  tels que  $x_B - x_A > y_B - y_A$  et  $y_B - y_A > z_B - z_A$ .

$v_x = x_B - x_A$  ;  $v_y = y_B - y_A$  ;  $v_z = z_B - z_A$  ;

$r_1 = v_z * x_A - v_x * z_A - \mu$  ;

$r_2 = v_y * x_A - v_x * y_A - \mu'$  ;

$x = x_A$  ;  $y = y_A$  ;  $z = z_A$  ;

**Tant que**  $x < x_B$  **répéter**

```

x = x + 1 ;
r1 = r1 + vz ;
r2 = r2 + vy ;
Si r1 < 0 ou r1 ≥ vx alors
    z = z + 1 ;
    r1 = r1 - vx ;
Fsi
Si r2 < 0 ou r2 ≥ vx alors
    y = y + 1 ;
    r2 = r2 - vx ;
Fsi
Afficher le point (x, y, z) ;
Fin répéter

```

---

Cet algorithme est linéaire et très simple par rapport à ceux existants déjà dans la littérature.

### 5.3 Algorithme de reconnaissance de segments de droites naïves 3D

La reconnaissance des segments de droites naïves 3D est une conséquence directe de la reconnaissance des segments de droites naïves 2D. En effet, un segment de droite 3D se projette bijectivement dans au moins deux plans de coordonnées en deux droites discrètes naïves 2D. Par conséquent, reconnaître un segment de droite naïve 3D consiste à reconnaître les deux segments de droites naïves 2D de ses projections fonctionnelles.

---

#### Algorithme de reconnaissance de segments de droites naïves 3D

Entrée : Ensemble 26-connexe de voxels à analyser, noté  $\mathcal{S}$

Sortie : Vrai si  $\mathcal{S}$  est un segment de droite discrète 3D, Faux sinon

**Si** les voxels de  $\mathcal{S}$  ne se projettent pas bijectivement sur au moins deux plans de coordonnées pour former deux courbes de pixels  $C_1$  et  $C_2$  **alors**  $\mathcal{S}$  n'est pas un segment de droite 3D, retourner Faux ; **Fsi**

Appliquer l'algorithme de reconnaissance de segments de droites naïves 2D sur  $C_1$  et  $C_2$  ;

**Si**  $C_1$  et  $C_2$  sont deux segments de droites discrètes **alors**  
     $\mathcal{S}$  est un segment de droite 3D, retourner Vrai ;  
**Sinon**  $\mathcal{S}$  n'est pas un segment de droite 3D, retourner Faux ;  
**Fsi**

---

Cet algorithme est linéaire et tous les calculs nécessaires se font exclusivement avec des nombres entiers.

Les caractéristiques du segment de droite reconnu sont déterminées en fonction des caractéristiques des segments de droites 2D obtenus. Par exemple, supposons que  $\mathcal{S}$  se projette bijectivement dans le plan  $Oxy$  en deux courbes  $C_1$  et  $C_2$ . L'algorithme de reconnaissance de segments de droites naïves 2D reconnaît  $C_1$  comme un segment de la droite  $\mathcal{D}(c_1, a_1, \mu_1)$  et  $C_2$  comme un segment de la droite  $\mathcal{D}(b_2, a_2, \mu_2)$ . Considérons  $m$  le plus petit multiple commun de  $a_1$  et  $a_2$  tel que  $m = k_1 a_1 = k_2 a_2$  alors  $\mathcal{S}$  est un segment de la droite naïve 3D de vecteur directeur  $V(m, k_2 b_2, k_1 c_1)$  et de bornes inférieures  $\mu_1$  et  $\mu_2$ .

## 5.4 Segmentation de courbes discrètes 26-connexes

Segmenter une courbe discrète 26-connexe, notée  $\mathcal{C}$ , consiste à “découper”  $\mathcal{C}$  en segments de droites naïves 3D. Pour cela, nous utilisons le même principe que l'algorithme présenté dans la section précédente, mais, pour atteindre une complexité linéaire, la recherche des plans de projections fonctionnelles n'est pas effectuée de la même manière. Nous procédons incrémentalement ; trois courbes discrètes 2D sont construites et segmentées au cours de la reconnaissance, elles correspondent aux projections du morceau de la courbe déjà parcouru dans les trois plans de coordonnées.

L'algorithme donné ci-dessous utilise les mêmes fondements que l'algorithme de segmentation de courbes discrètes 2D présenté dans le chapitre 3. Une suite  $\mathcal{C}$  de voxels 26-connexes est donnée en entrée.

---

Algorithme de segmentation de courbes discrètes 26-connexes

$M$  = premier voxel de  $\mathcal{C}$

**Tant que**  $\mathcal{C}$  n'a pas été entièrement parcourue **répéter**

    Dans chaque plan de coordonnées  $Oxy$ ,  $Oxz$ ,  $Oyz$ , initialiser les caractéristiques et les points d'appui principaux de nouveaux segments  $S_1$ ,  $S_2$  et  $S_3$  ;

SEGMENT=vrai ;

**Tant que** le voxel ajouté  $M$  possède au moins deux projections se trouvant dans une des situations (i), (ii) ou (iii) du théorème 3.1 et SEGMENT **répéter**

Si nécessaire, recalculer les caractéristiques et les points d'appui principaux des droites supports des segments 2D concernés ;

**Si** les projections du voxel suivant de  $\mathcal{C}$  dans les deux plans de coordonnées utilisés appartiennent aux mêmes octants que les segments 2D courants **alors**

$M$  = point suivant de  $\mathcal{C}$  ;

Ajouter en les symétrisant, les pixels projetés de  $M$ , aux segments 2D concernés ;

**Sinon**

SEGMENT = faux ;

**Fsi**

**Fin répéter**

En fonction des symétries, calculer les caractéristiques des droites supports des deux segments 2D obtenus dans les octants initiaux, puis en déduire le vecteur directeur du segment de droite naïve 3D reconnu ;

**Fin répéter**

---

Remarque : Le premier point d'un nouveau segment est le dernier du segment précédent.

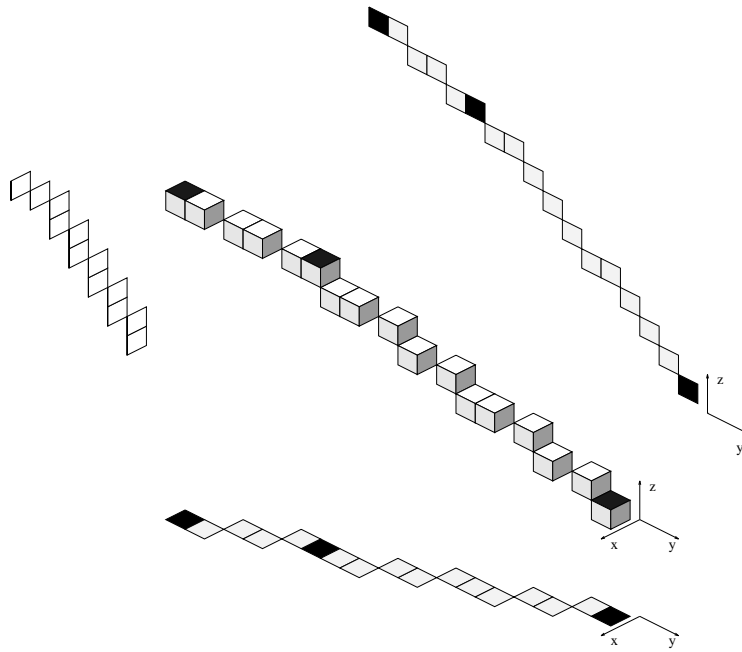
Sur la figure 5.7 nous voyons un exemple de segmentation de courbe discrète, les voxels de fin et de début de segment sont en noir. Les plans de projections utilisés sont  $Oxy$  et  $Oyz$ , les segments résultant de la reconnaissance 2D sont marqués en gris clair sur les projections dans ces deux plans de l'ensemble de voxels considéré.

Deux segments de droites naïves sont obtenus :

Le segment 1 est de longueur 11, voici les équations des droites supports des segments 2D obtenus :

$$\begin{cases} -4 \leq -4y - 5z < 1 \\ -2 \leq -2y - 5x < 3 \end{cases}$$

Par conséquent, la droite naïve 3D support du segment 1 a pour vecteur directeur  $V(2, -5, 4)$ .



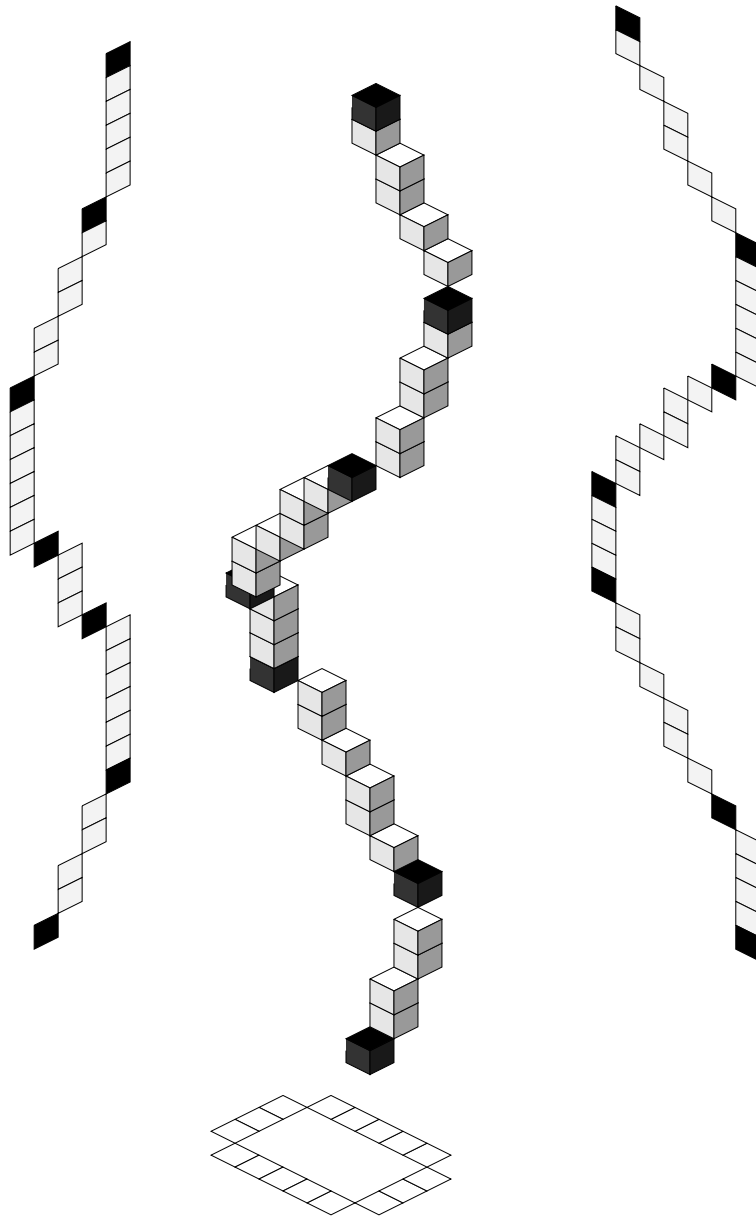
**Figure 5.7**

Le segment 2 est de longueur 5, les équations des droites supports des segments 2D obtenus sont les suivantes :

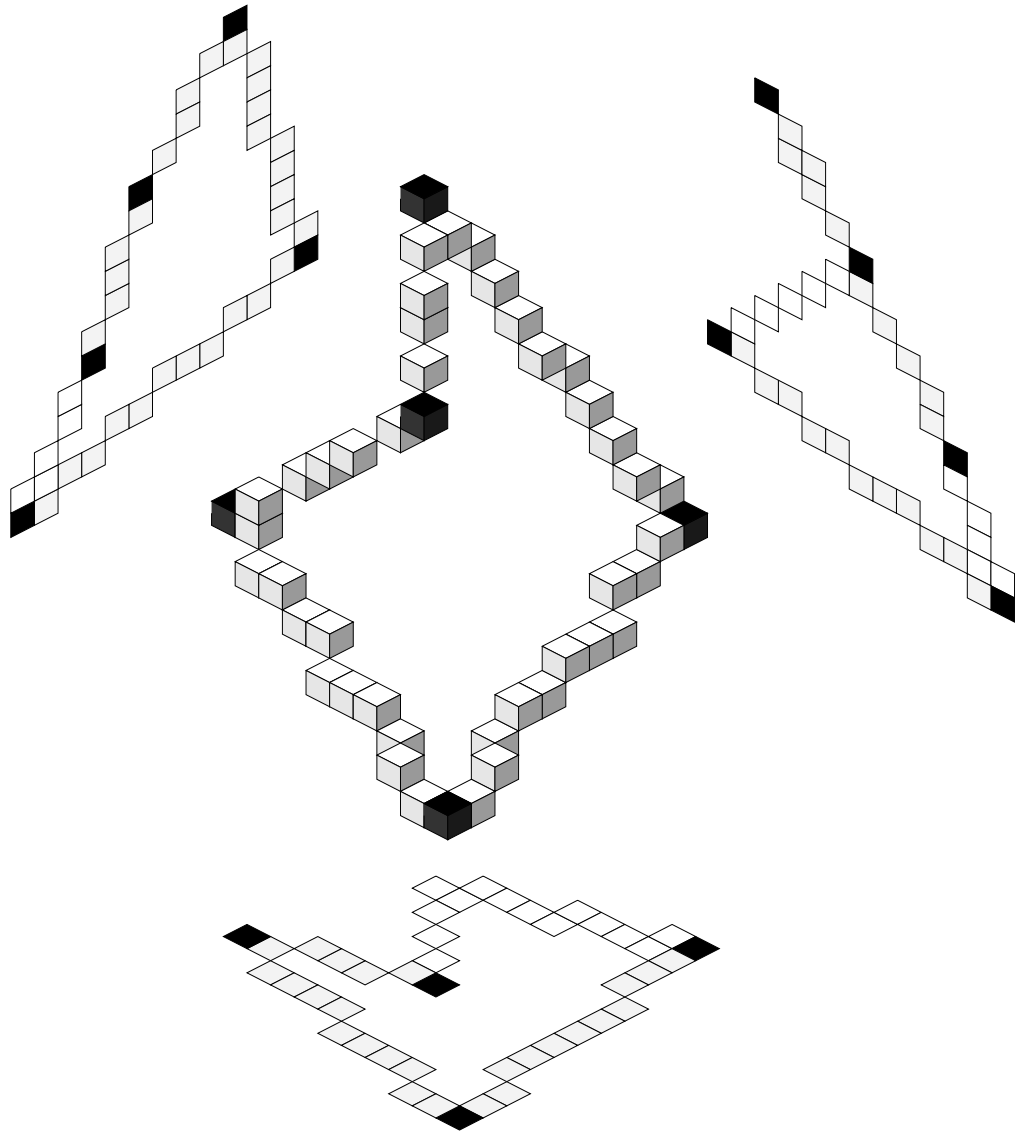
$$\begin{cases} 0 \leq y - 2z < 2 \\ 0 \leq y - 2x < 2 \end{cases}$$

Par conséquent, la droite naïve 3D support du segment 2 a pour vecteur directeur  $V(1, -2, 1)$ .

Deux autres exemples de segmentation de courbes 26-connexes sont donnés dans les pages suivantes.



**Figure 5.8 :** *Segmentation d'une spirale discrète ; les voxels noirs indiquent les débuts et les fins de segments. Deux des trois plans de projection ont été utilisés pour reconnaître la courbe (les courbes 2D segmentées sont en gris clair).*



**Figure 5.9** : *Les voxels noirs indiquent les débuts et les fins de segments. Pour la segmentation de cette courbe 26-connexe, les trois plans de projection ont été utilisés pour reconnaître la courbe (les courbes 2D segmentées sont en gris clair).*

## 5.5 Conclusion

Nous avons énoncé dans ce chapitre une définition arithmétique de droite discrète 3D, grâce à la richesse et à la multiplicité des structures pouvant être engendrées à



partir d'un vecteur directeur donné, elle recouvre la plupart des définitions données dans la littérature. De plus, la propriété de projection bijective d'un segment de droite 3D sur deux plans de coordonnées en deux segments de droites 2D nous a permis d'obtenir très facilement un algorithme de parcours de droite 3D et un algorithme très efficace de segmentation de courbes discrètes 26-connexes en segments de droites naïves 3D.

Ce chapitre est une introduction à l'étude générale des droites discrètes 3D ; beaucoup de questions restent en suspens. Par exemple, deux points d'un plan discret naïf étant donnés, existe-t-il une droite naïve 3D appartenant à ce plan qui passe par ces deux points ? Une définition plus générale, incluant la notion de droite 3D donnée dans ce chapitre, est en cours d'étude par O. Figueiredo [19].

# Chapitre 6

## Généralités sur les plans discrets naïfs

## 6.1 Introduction

Les plans discrets ont été étudiés par plusieurs auteurs (cf. [20, 35, 38, 59, 62, 63, 23, 21, 22]) pour répondre à des problèmes particuliers, par exemple en imagerie médicale, ou encore en espérant construire une version discrète de la géométrie euclidienne. De notre point de vue, les difficultés rencontrées dans certaines de ces études proviennent de la définition usuelle des plans discrets comme discrétisation des plans euclidiens *irrationnels*. Mais, dans la pratique, les plans rationnels sont suffisants, et leur discrétisation est plus simple ; c'est l'ensemble des points entiers satisfaisant une double inégalité diophantienne,

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

où *tous les paramètres sont entiers*. Cette façon de penser est radicalement nouvelle car cela évite de mélanger des réels et des entiers et ainsi, le problème difficile de l'approximation de vecteurs 3D irrationnels par des rationnels est détourné.

Ainsi définis, sans aucune référence aux notions euclidiennes (réelles), les plans discrets ont de nombreuses propriétés intéressantes développées dans ce chapitre.

La définition des plans discrets naïfs est énoncée dans la première section de ce chapitre ainsi que des propriétés de symétrie qui permettent de réduire leur étude aux plans dont le vecteur normal est situé dans un 48ième d'espace donné. De plus, la propriété de fonctionnalité permet de donner plusieurs types de codage bidimensionnel d'un plan discret naïf. La seconde section utilise la présence des droites discrètes naïves 2D dans un plan naïf pour en déduire des théorèmes de structure et des algorithmes de parcours de plans et d'intersection très simples. Enfin, dans la dernière partie, nous montrons comment une séquence modulaire, sous-jacente à la définition arithmétique, décrit la structure interne des plans discrets naïfs.

## 6.2 Définitions et premières propriétés

### 6.2.1 Le plan discret naïf

La définition arithmétique du plan discret a été introduite par E. Andres [2] ; elle est la généralisation de la définition arithmétique d'une droite discrète 2D.

Un *plan discret*

- ◇ de caractéristiques  $a, b, c$  telles que  $a \wedge b \wedge c = 1$
- ◇ de borne inférieure  $\mu$
- ◇ d'épaisseur arithmétique  $\omega$

est l'ensemble des points  $(x, y, z)$  de  $\mathbb{Z}^3$  satisfaisant la double inéquation diophantienne

$$\mu \leq ax + by + cz < \mu + \omega \quad (1)$$

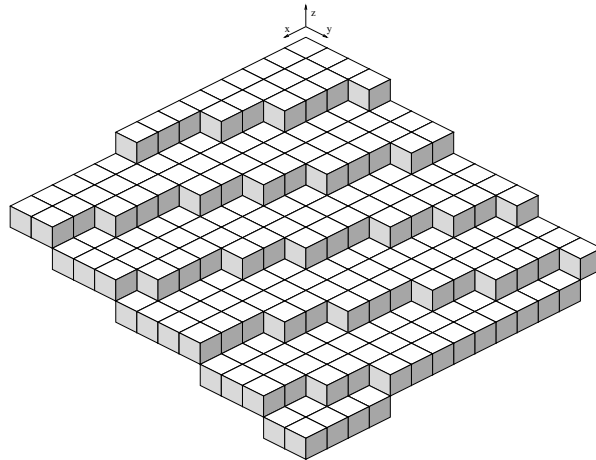
avec tous les paramètres entiers.

Notre étude des plans discrets porte sur les plans 18-connexes sans trou 6-connexe [2] pour lesquels :

$$\omega = \max(|a|, |b|, |c|)$$

Ces plans discrets sont appelés *plans discrets naïfs*.

Le plan discret naïf de caractéristiques  $a, b, c$  et de borne inférieure  $\mu$  sera noté  $P(a, b, c, \mu)$  ou plus simplement  $P$  s'il n'y a pas de confusion possible sur le vecteur normal et la borne inférieure.



**Figure 6.1 :** Représentation en voxels d'un morceau du plan discret  $P(7, 17, 57, 0)$  pour  $0 \leq x \leq 13$  et  $0 \leq y \leq 13$ .

Nous nommerons **réseau d'indice**  $\alpha$  de  $P(a, b, c, \mu)$  pour  $\alpha \in [\mu, \mu + \max(|a|, |b|, |c|)]$ , l'ensemble des points  $(x, y, z)$  de  $\mathbb{Z}^3$  vérifiant l'équation

$$ax + by + cz = \alpha.$$

Par la suite, il sera noté  $\mathbf{res}(\mathbf{a}, \mathbf{b}, \mathbf{c})(\alpha)$  ou  $\mathbf{res}(\alpha)$ .

Un plan discret naïf est donc constitué de la réunion de ses réseaux, d'indices variant de  $\mu$  à  $\mu + \max(|a|, |b|, |c|) - 1$  (cf. Figure 6.2).

*Théorème 6.1 :*

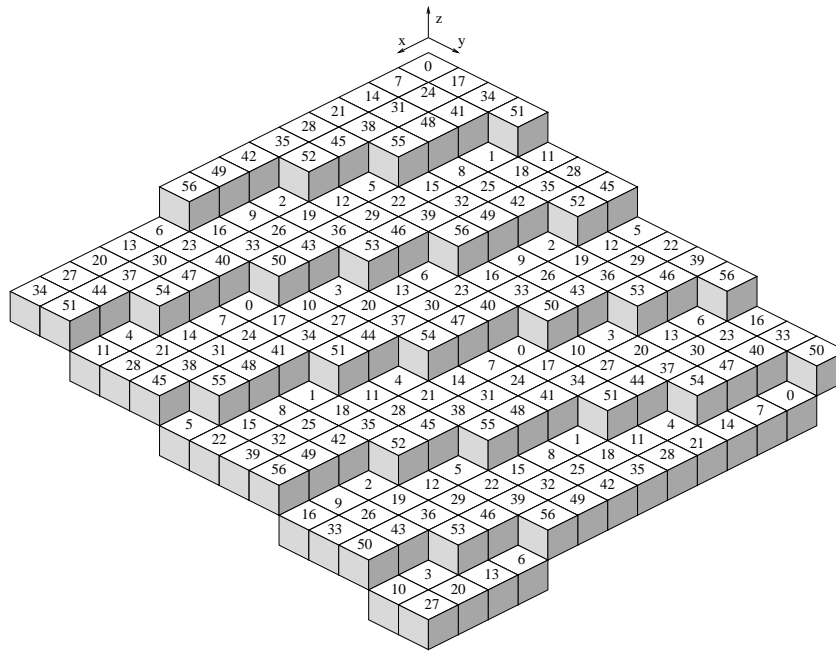
*Soit  $P(a, b, c, \mu)$  un plan discret, pour tout  $\alpha$  et  $\beta$  de l'intervalle  $[\mu, \mu + \max(|a|, |b|, |c|)]$ , il existe un vecteur  $u$  tel que la translation de vecteur  $u$  transforme le réseau d'indice  $\alpha$  de  $P$  en le réseau d'indice  $\beta$  de  $P$ .*

En effet, supposons  $0 \leq a \leq b \leq c$  et prenons par exemple  $u = (l, m, 0)$  avec  $l$  et  $m$  de  $\mathbb{Z}$  vérifiant  $la + mb = \beta - \alpha$ . Nous avons,

$$\begin{aligned} & (x, y, z) \in \mathbf{res}(\alpha) \\ \Leftrightarrow & ax + by + cz = \alpha \\ \Leftrightarrow & a(x + l) + b(y + m) + cz = \beta \\ \Leftrightarrow & (X = x + l, Y = y + m, Z = z) \in \mathbf{res}(\beta) \blacksquare \end{aligned}$$

Dans l'étude des plans discrets, pour un plan  $P$  donné, l'indice du réseau auquel appartient chaque point de  $P$  joue un rôle primordial, c'est pourquoi nous introduisons la définition suivante :

Soit  $M(x, y, z)$  un point du plan  $P(a, b, c, \mu)$ , la valeur  $ax + by + cz$  sera appelée **reste de  $\mathbf{M}$  par rapport à  $\mathbf{P}$**  et notée  $\mathbf{r}(\mathbf{a}, \mathbf{b}, \mathbf{c})(\mathbf{M})$  ou plus simplement  $\mathbf{r}(\mathbf{M})$ .



**Figure 6.2 :** Morceau du plan discret  $P(7, 17, 57, 0)$  pour  $0 \leq x \leq 13$  et  $0 \leq y \leq 13$  où chaque voxel est pondéré par l'indice du réseau de  $P$  auquel il appartient.

## 6.2.2 Symétries

Il existe une relation très simple entre les plans discrets et le groupe  $\mathcal{G}$  des isométries (d'ordre 48) du cube unité. Un domaine fondamental de  $\mathcal{G}$  est le sous-ensemble de  $\mathbb{R}^3$  contenant les points satisfaisant les conditions  $0 \leq x \leq y \leq z \leq 1$  (i.e. le simplexe standard). Le résultat suivant se déduit de manière immédiate des définitions.

*Théorème 6.2 :*

*Le groupe  $\mathcal{G}$  opère simplement et transitivement sur l'ensemble des plans discrets naïfs.*

**Ce théorème permet de réduire l'étude des plans discrets à ceux dont le vecteur normal  $(a, b, c)$  est contenu dans le cône standard défini par  $0 \leq a \leq b \leq c$ .**

Ce simplexe est un 48ième d'espace, les symétries induites par ce groupe permettent de ramener les 47 autres sous-espaces à celui-ci.

Les transformées d'un plan discret  $P(a, b, c, \mu)$  par les symétries les plus courantes sont données sur la page suivante.

La transformée par la symétrie par rapport

- à  $O(0, 0, 0)$  est  $P(-a, -b, -c, \mu)$  ou  $P(a, b, c, -c + 1 - \mu)$
- au plan de coordonnées
  - $Oxy$  est  $P(a, b, -c, \mu)$
  - $Oxz$  est  $P(a, -b, c, \mu)$
  - $Oyz$  est  $P(-a, b, c, \mu)$
- au plan bissecteur
  - $x = y$  est  $P(b, a, c, \mu)$
  - $x = z$  est  $P(c, b, a, \mu)$
  - $y = z$  est  $P(a, c, b, \mu)$
- à la droite
  - $x = y = 0$  est  $P(-a, -b, c, \mu)$
  - $x = z = 0$  est  $P(-a, b, -c, \mu)$
  - $y = z = 0$  est  $P(a, -b, -c, \mu)$

### 6.2.3 Représentations d'un plan discret naïf par lignes de niveaux

La propriété suivante va permettre de “visualiser” les plans naïfs en dimension 2 ce qui facilitera l'étude de ces objets 3D.

*Théorème 6.3 :*

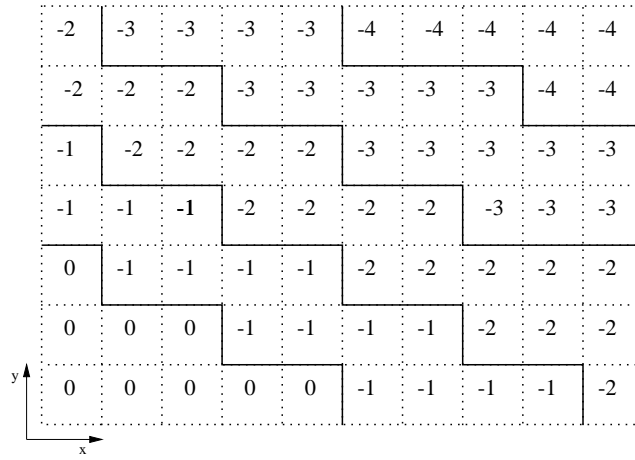
*Un plan discret naïf  $P(a, b, c, \mu)$  est fonctionnel.*

En effet, supposons  $0 \leq a \leq b \leq c$  alors

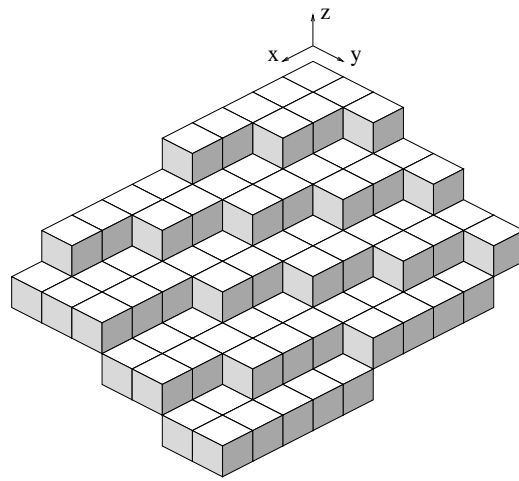
$$\{(x, y, z) \in \mathbb{Z}^3 \mid \mu \leq ax + by + cz < \mu + c\}$$

$$\iff \left\{ \left( x, y, z = - \left\lceil \frac{ax + by - \mu}{c} \right\rceil \right) \text{ avec } (x, y) \in \mathbb{Z}^2 \right\} \quad \blacksquare$$

Nous pouvons alors en déduire que chaque voxel d'un plan  $P(a, b, c, \mu)$  avec  $0 \leq a \leq b \leq c$  a une unique projection dans le plan  $Oxy$ .  $P$  peut donc être **codé** par ses lignes de niveaux, c'est à dire par des pixels de coordonnées  $(x, y)$ , numérotés par  $z$  comme dans l'exemple ci-dessous.



**Figure 6.3 :** Représentation par lignes de niveaux du plan  $P(6, 13, 27, 0)$  dans la fenêtre  $[0, 9] * [0, 6]$ , l'origine est le pixel le plus proche du repère.



**Figure 6.4 :** Représentation en voxels du même morceau de plan dans le parallélépipède  $[0, 9] * [0, 6] * [0, -4]$ , l'origine est le voxel le plus proche du repère.

## 6.2.4 Plans discrets équivalents

*Théorème 6.4 :*

Deux plans discrets naïfs  $P(a, b, c, \mu)$  et  $P'(a', b', c', \mu')$  sont égaux à une translation près si et seulement s'ils ont le même vecteur normal  $(a, b, c) = (a', b', c')$ .



En effet, à un réseau de niveau  $\alpha$  de  $P$  correspond par translation un réseau de  $P'$  de même structure. Par suite, ces deux réseaux ont une base identique et par conséquent  $a = a'$ ,  $b = b'$  et  $c = c'$ .

Réciproquement, considérons le point  $M$  de  $P$  ayant pour coordonnées  $(x, y, z)$  telles que  $ax + by + cz = \mu$  et le point  $M'$  de  $P'$  ayant pour coordonnées  $(x', y', z')$  telles que  $a'x' + b'y' + c'z' = \mu'$ ; ces deux réseaux étant les plus “bas” de  $P$  et  $P'$  ont une structure identique. Par conséquent la translation de vecteur  $M\vec{M}'$  transforme  $P$  en  $P'$ . ■

Le théorème suivant est une première conséquence de ce résultat.

*Théorème 6.5 :*

*Si  $(\mathcal{P})$  est un plan euclidien d'équation  $\alpha x + \beta y + \gamma z = \delta$  avec  $(x, y, z) \in \mathbb{R}^3$  et  $\alpha, \beta, \gamma, \delta$  des nombres rationnels avec  $\max(|\alpha|, |\beta|, |\gamma|) = |\gamma|$ , alors la discrétisation de  $(\mathcal{P})$  par les points entiers les plus proches est donnée par le plan naïf  $P(a, b, c, \mu)$  où,*

$$\alpha = \frac{a}{c}, \beta = \frac{b}{c}, \gamma = 1, \delta = \frac{d}{c},$$

*et la borne inférieure  $\mu = d - \left[\frac{1}{2}c\right]$ .*

Le plan  $P(a, b, c, \mu)$  défini ci-dessus correspond aux définitions de plan discret données dans [35, 38, 63].

## 6.3 Sections par les plans de coordonnées et conséquences

Dans tout ce paragraphe, nous considérons des plans discrets naïfs  $P(a, b, c, \mu)$  dont les caractéristiques vérifient  $0 \leq a \leq b \leq c$ .

### 6.3.1 Présence de droites discrètes 2D dans un plan naïf

Le théorème suivant fournit une propriété très intéressante pour l'étude des plans discrets.

*Théorème 6.6 :*

*L'intersection du plan  $P(a, b, c, \mu)$  avec le plan*

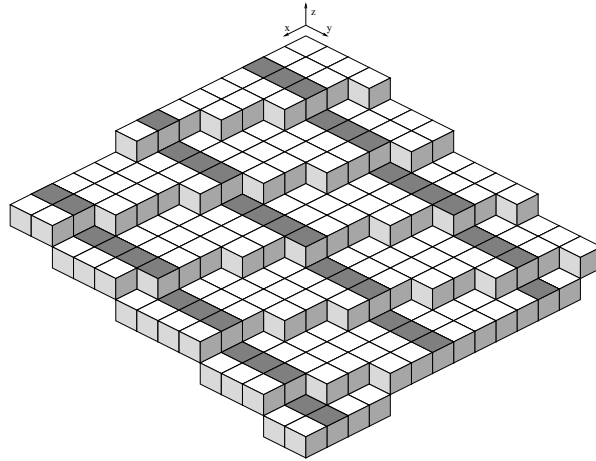
- (i)  $z = z_0, z_0 \in \mathbb{Z}$ , a pour projection dans le plan  $Oxy$  la droite discrète épaisse  $\mathcal{D}(a, b, \mu - cz_0, c)$  et pour toutes les valeurs entières  $z_0$  de  $\mathbb{Z}$  ces droites discrètes sont équivalentes, i.e. égales à une translation près,
- (ii)  $y = y_0, y_0 \in \mathbb{Z}$ , a pour projection dans le plan  $Oxz$  la droite discrète naïve  $\mathcal{D}(a, c, \mu - by_0, c)$  et pour toutes les valeurs entières  $y_0$  de  $\mathbb{Z}$  ces droites discrètes sont équivalentes,
- (iii)  $x = x_0, x_0 \in \mathbb{Z}$ , a pour projection dans le plan  $Oyz$  la droite discrète naïve  $\mathcal{D}(b, c, \mu - ax_0, c)$  et pour toutes les valeurs entières  $x_0$  de  $\mathbb{Z}$  ces droites discrètes sont équivalentes.

En effet, ces trois points sont évidents, regardons en détail (i), l'intersection de  $P$  avec le plan  $z = z_0$  est l'ensemble des points vérifiant la double inéquation

$$\mu - cz_0 \leq ax - by < \mu - cz_0 + c$$

ce qui est, par définition, la droite  $\mathcal{D}(a, b, \mu - cz_0, c)$  dans le plan  $Oxy$ .

De plus, si  $z_0$  varie dans  $\mathbb{Z}$ , seule la borne inférieure  $\mu$  change et par conséquent selon [50], les droites obtenues sont équivalentes. ■



**Figure 6.5 :** Sections du plan discret  $P(7, 17, 57, 0)$  pour  $x = 2, x = 7$  et  $x = 12$ . Pour  $x = 2$ , la section obtenue dans le plan  $Oyz$  est la droite d'équation

$$-14 \leq 17y + 57z < 43.$$

Nous pouvons déduire de ce théorème une condition nécessaire que doit vérifier un ensemble de voxels pour appartenir à un plan discret naïf.

**CN1 :**

*Soit  $V$  un ensemble de points de  $\mathbb{Z}^3$  se projetant de manière bijective dans le plan  $Oxy$ .*

*Si  $V$  est un morceau de plan discret naïf alors toutes les sections en  $x$  et les sections en  $y$  de  $V$  sont des droites discrètes naïves et les sections en  $z$  des droites discrètes pouvant être épaisses.*

Pratiquement, les algorithmes de reconnaissance des droites discrètes développés dans les chapitres précédents doivent s'exécuter avec succès sur les sections de  $V$ .

La présence des droites discrètes  $2D$  dans un plan discret et les travaux déjà effectués sur les droites ([50], [29]) vont nous permettre dans les sections suivantes d'obtenir très simplement plusieurs propriétés intéressantes des plans discrets.

### 6.3.2 Paliers d'un plan naïf

Les paliers d'une droite discrète  $2D$  vérifient la propriété suivante [50] :

*Dans une droite discrète  $\mathcal{D}(a, b, \mu, \omega)$ , avec  $0 \leq a \leq b$ ,*

*– les longueurs des paliers horizontaux sont  
les entiers consécutifs  $\left\lfloor \frac{\omega}{a} \right\rfloor$  et  $\left\lfloor \frac{\omega}{a} \right\rfloor + 1$  si  $\left\{ \frac{\omega}{a} \right\} \neq 0$*

*ou*

*l'entier  $\left\lfloor \frac{\omega}{a} \right\rfloor$  si  $\left\{ \frac{\omega}{a} \right\} = 0$*

*– les longueurs des paliers verticaux sont*

*les entiers consécutifs  $\left\lfloor \frac{\omega}{b} \right\rfloor$  et  $\left\lfloor \frac{\omega}{b} \right\rfloor + 1$  si  $\left\{ \frac{\omega}{b} \right\} \neq 0$*

*ou*

*l'entier  $\left\lfloor \frac{\omega}{b} \right\rfloor$  si  $\left\{ \frac{\omega}{b} \right\} = 0$*

Cette propriété et le théorème du paragraphe 6.3.1 ont des conséquences immédiates sur les plans discrets :

Soit  $P(a, b, c, \mu)$  un plan discret naïf tel que  $0 \leq a \leq b \leq c$ , un **palier horizontal** (resp. **vertical**) de  $P$  est une suite "maximale" de points  $(x, y, z)$  de  $P$  consécutifs en  $x$  (resp. en  $y$ ) ayant même  $y$  (resp. même  $x$ ) et même  $z$ ; i.e. la projection d'un palier horizontal (resp. vertical) de  $P$  dans le plan  $Oxz$  (resp.  $Oyz$ ) est un palier horizontal (resp. vertical) de la droite  $\mathcal{D}(a, c, \mu - by, c)$  (resp.  $\mathcal{D}(b, c, \mu - ax, c)$ ).

Par conséquent,

- les longueurs des paliers horizontaux de  $P$  sont les entiers consécutifs  $\left[\frac{c}{a}\right]$  et  $\left[\frac{c}{a}\right] + 1$  si  $\left\{\frac{c}{a}\right\} \neq 0$   
ou  
l'entier  $\left[\frac{c}{a}\right]$  si  $\left\{\frac{c}{a}\right\} = 0$   
de même,
- les longueurs des paliers verticaux de  $P$  sont les entiers consécutifs  $\left[\frac{c}{b}\right]$  et  $\left[\frac{c}{b}\right] + 1$  si  $\left\{\frac{c}{b}\right\} \neq 0$   
ou  
l'entier  $\left[\frac{c}{b}\right]$  si  $\left\{\frac{c}{b}\right\} = 0$

En étendant la notion de paliers dans un plan discret à celle de paliers dans un ensemble de voxels, nous pouvons déduire de la propriété précédente une condition nécessaire que doit vérifier un ensemble de voxels pour appartenir à un plan discret naïf :

**CN2 :**

*Soit  $V$  un ensemble de points de  $\mathbb{Z}^3$  se projetant de manière bijective dans le plan  $Oxy$ .*

*Si  $V$  est un morceau de plan discret naïf alors  $V$  possède au plus deux longueurs de paliers horizontaux et au plus deux longueurs de paliers verticaux.*

### 6.3.3 Algorithmes de parcours

Un plan discret étant fonctionnel, il suffit pour le représenter de calculer la composante  $z$  en fonction de  $x$  et  $y$ , ces derniers variant dans le domaine  $[x_0, x_1] \times [y_0, y_1]$ . Un algorithme immédiat consiste donc à calculer  $z$  en utilisant la relation

$$z = - \left[ \frac{ax + by - \mu}{c} \right]$$

La division entière de cette formule rend l'algorithme assez lent et nous pouvons optimiser la recherche de  $z$  en utilisant la présence des droites discrètes 2D dans un plan discret. Le théorème du paragraphe 6.3.1 nous permet de construire un plan discret par ses sections, or nous disposons dans [50] de nombreux algorithmes de parcours de droites discrètes 2D à partir desquels nous pouvons élaborer des algorithmes de construction de plans discrets naïfs dont les plus efficaces sont présentés ci-après.

Un premier algorithme rapide consiste à construire les sections "horizontales" d'un plan  $P(a, b, c, \mu)$ , c'est à dire les droites naïves  $\mu - by_0 \leq ax + cz < \mu - by_0 - c$

pour  $y_0$  appartenant à  $\mathbb{Z}$ .

---

Un algorithme de construction du plan  $P(a, b, c, \mu)$  par “sections en  $y$ ” pour  $x$  et  $y$  dans le domaine  $[x_0, x_1] \times [y_0, y_1]$

```
zini = - ⌊  $\frac{a*x_0 + b*y_0 - \mu}{c}$  ⌋ ; z = zini ;
rini = a * x0 + b * y0 + c * z - μ ; r = rini ;
pour y = y0 à y1 répéter
  pour x = x0 à x1 répéter
    construire le voxel correspondant au point (x, y, z) ;
    r = r + a ;
    si r ≥ c alors faire
      z = z - 1 ;
      r = r - c ;
    fsi
  fpour
    rini = rini + b ;
    si rini ≥ c alors faire
      zini = zini - 1 ;
      rini = rini - c ;
    fsi
  z = zini ; r = rini ;
fpour
```

---

Bien sûr, l'algorithme qui construit les sections “verticales” est similaire; il suffit d'échanger  $x$  avec  $y$  et  $a$  avec  $b$ . Cet algorithme est rapide et a une complexité similaire aux algorithmes de Brésenham.

Pour construire les droites naïves correspondant aux sections horizontales de  $P$ , nous pouvons aussi utiliser l'algorithme de construction par paliers longs et courts [50] qui est plus rapide que le précédent, l'algorithme correspondant est présenté ci-dessous.

---

Un algorithme de construction du plan  $P(a, b, c, \mu)$  par paliers des “sections en  $y$ ” pour  $x$  et  $y$  dans le domaine  $[x_0, x_1] \times [y_0, y_1]$

```

zini = -  $\left\lceil \frac{a*x_0 + b*y_0 - \mu}{c} \right\rceil$ ; z = zini;
rini = a * x0 + b * y0 + c * z -  $\mu$ ; r = rini;
pour y = y0 à y1 répéter
  pour x = x0 à x1 répéter
    si x = x0 alors faire
      tant que r < c répéter
        r = r + a;
        x = x + 1;
        construire le voxel correspondant au point (x, y, z);
      ftantque
        r = r - c;
    sinon
      si r <  $\left\{ \frac{c}{a} \right\}$  alors faire
        tracer un palier long de longueur  $\left\lceil \frac{c}{a} \right\rceil + 1$ ;
        x = x +  $\left\lceil \frac{c}{a} \right\rceil + 1$ ;
        r = r + a -  $\left\{ \frac{c}{a} \right\}$ ;
      sinon
        tracer un palier court de longueur  $\left\lceil \frac{c}{a} \right\rceil$ ;
        x = x +  $\left\lceil \frac{c}{a} \right\rceil + 1$ ;
        r = r -  $\left\{ \frac{c}{a} \right\}$ ;
    fsi
  fsi
  z = z - 1;
fpour
  rini = rini + b;
  si rini ≥ c alors faire
    zini = zini - 1;
    rini = rini - c;
  fsi
z = zini; r = rini;
fpour

```

---

Remarque : Le tracé d'un palier sous-entend de tester si la borne  $x_1$  n'est pas atteinte.

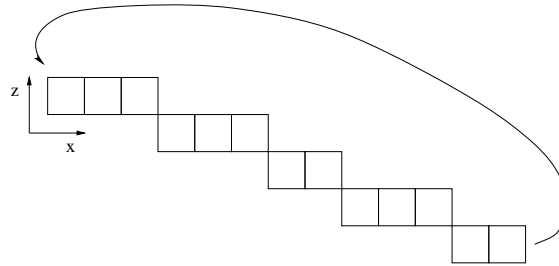
Un troisième algorithme encore plus performant consiste à construire le premier

niveau en  $y$  du plan  $P(a, b, c, \mu)$  avec une des méthodes déjà proposées et à stocker sur une période la structure de la droite naïve correspondante. Par suite, un plan discret étant un “assemblage” de droites naïves de même pente donc de même structure, il suffit de reconstruire cette structure génératrice en tenant compte du décalage constant entre deux niveaux  $y$  successifs. Ce décalage correspond au nombre  $\alpha \in [0, c]$  solution de l'équation

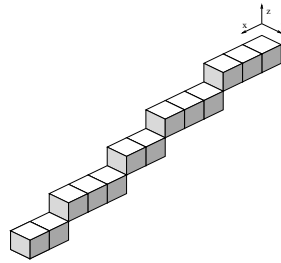
$$\alpha a + \beta c = b$$

Construisons avec cette méthode le plan  $P(5, 7, 13, 0)$  pour  $x$  et  $y$  dans le domaine  $[0, 12] \times [0, 12]$ .

En construisant le niveau  $y = 0$  nous obtenons la structure génératrice :



**Figure 6.6 :** Structure génératrice du plan  $P(5, 7, 13, 0)$  sur une période.



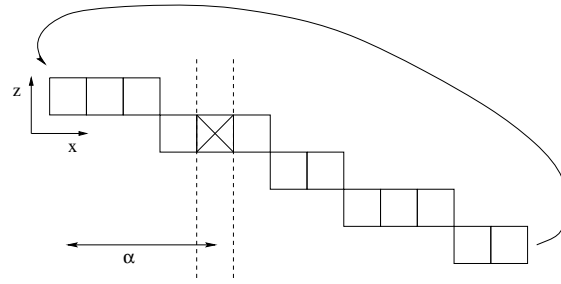
**Figure 6.7 :** Niveau  $y = 0$  de  $P(5, 7, 13, 0)$  pour  $x \in [0, 12]$

L'ensemble des solutions de l'équation  $\alpha a + \beta c = b$  est l'ensemble des couples  $(\alpha, \beta)$  vérifiant

$$\begin{cases} \alpha = -35 + 13k & \forall k \in \mathbb{Z} \\ \beta = 14 + 7k \end{cases}$$

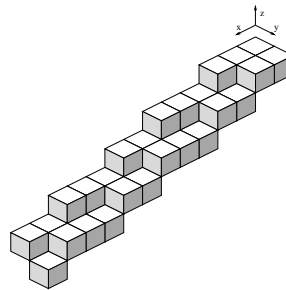
de plus, pour  $k = 3$ ,  $\alpha$  vaut 4 et appartient donc à l'intervalle  $[0, 13[$ . Le décalage  $\alpha$  entre le niveau  $y = 0$  et  $y = 1$  correspond au numéro de l'élément de la structure génératrice qui sera le premier point du niveau  $y = 1$  qui est marqué par une croix

sur la figure ci-dessous.



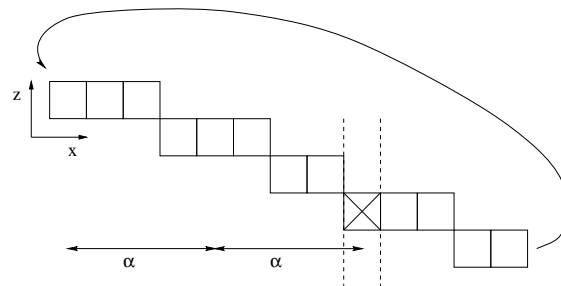
**Figure 6.8**

Le calcul de la composante en  $z$  du point  $(0, 1, z)$  se fait selon la même méthode que dans les algorithmes décrits précédemment, on obtient ainsi la construction suivante :



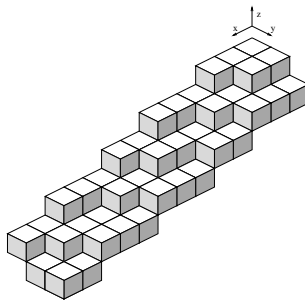
**Figure 6.9** : Niveau  $y = 0$  et  $y = 1$  de  $P(5, 7, 13, 0)$  pour  $x \in [0, 12]$ .

On itère le processus pour construire le niveau  $y = 2$  en se déplaçant de  $\alpha$  dans la structure initiale, vous voyez sur la figure suivante l'élément correspondant au premier point du niveau  $y = 2$ .



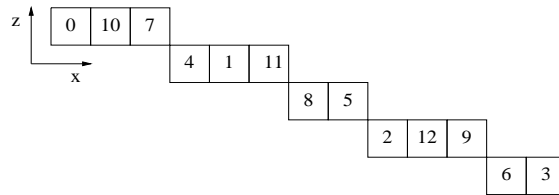
**Figure 6.10**



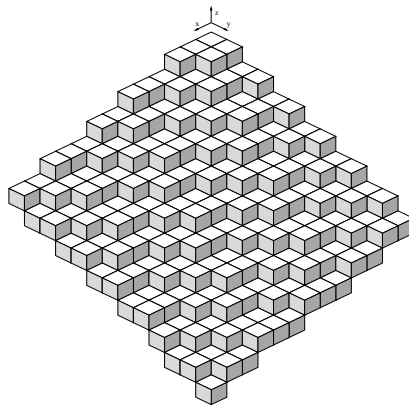


**Figure 6.11 :** Niveau  $y = 0$ ,  $y = 1$  et  $y = 2$  de  $P(5, 7, 13, 0)$  pour  $x \in [0, 12]$ .

Les autres niveaux en  $y$  de  $P$  s'obtiennent de manière identique en repérant très simplement grâce à l'entier  $\alpha$  où se trouve dans la structure génératrice du plan le premier élément de chaque niveau  $y$ .



**Figure 6.12 :** Les pixels de la structure génératrice sont pondérés par le niveau  $y$  pour lesquels ils sont premier élément.



**Figure 6.13 :**  $P(5, 7, 13, 0)$  pour  $x$  et  $y$  dans le domaine  $[0, 12] \times [0, 12]$ .

L'algorithme correspondant à cette méthode est donné ci-dessous,  $|x_1| \geq c$  :

---

Un algorithme de construction du plan  $P(a, b, c, \mu)$  par “assemblage” des structures des “sections en  $y$ ” pour  $x$  et  $y$  dans le domaine  $[0, x_1] \times [0, y_1]$

```

zini = -  $\left\lfloor \frac{a*x_0 + b*y_0 - \mu}{c} \right\rfloor$  ;
rini = a * x0 + b * y0 + c * z -  $\mu$  ;
calculer le nombre  $\alpha \in [0, c[$  solution de  $\alpha a + \beta c = b$  ;
construire le niveau  $y = 0$  et stocker la structure génératrice ;
position = 0 ; // position de l'élément de la structure correspondant
                au premier point du niveau courant en  $y$  à construire //
pour  $y = 0$  à  $y_1$  répéter
    rini = rini + b ;
    si rini  $\geq c$  alors faire
        zini = zini - 1 ;
        rini = rini - c ;
    fsi
    position =  $\left\lfloor \frac{\text{position} + \alpha}{c} \right\rfloor$  ;
    construire à partir du point  $(0, y, z_{ini})$  pour  $y$  constant,  $\left\lfloor \frac{x_1}{c} \right\rfloor$ 
    périodes plus  $\left\lfloor \frac{x_1}{c} \right\rfloor$  points de la structure génératrice à partir
    de l'élément d'abscisse position ;
fpour

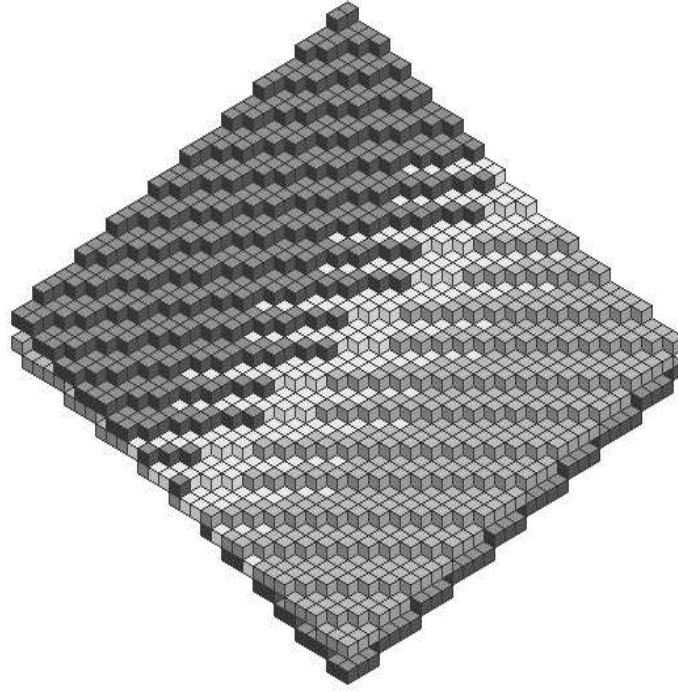
```

---

Des algorithmes de construction par niveaux  $z$  sont également possibles et reposent sur les algorithmes de construction de droites épaisses mais les algorithmes décrits précédemment sont plus efficaces.

### 6.3.4 Intersection de deux plans naïfs

Comme pour les droites discrètes, l'intersection de 2 plans discrets naïfs est dans la majorité des cas très distincte de l'intersection de deux plans réels. Les propriétés d'intersection des plans discrets et euclidiens sont radicalement différentes ; dans le cas discret, ces intersections peuvent être non connexes, vides ou former des “bandes” communes de points plus ou moins larges. Nous en voyons un exemple sur la figure suivante.



**Figure 6.14 :** Les voxels de  $P_1(2, 2, 7, 0)$  sont en gris clair, ceux de  $P_2(2, 3, 7, 0)$  sont en gris foncé et ceux de  $P_1 \cap P_2$  en gris très clair.

Considérons les deux plans discrets naïfs  $P_1$  et  $P_2$  définis par les inéquations

$$\mu \leq ax + by + cz < \mu + c \quad (P_1)$$

$$\nu \leq dx + ey + fz < \nu + f \quad (P_2)$$

avec  $0 \leq a \leq b \leq c$  tels que  $a \wedge b \wedge c = 1$  et  $0 \leq d \leq e \leq f$  tels que  $d \wedge e \wedge f = 1$ .

Nous avons les relations suivantes ,

$$(P_1) \Leftrightarrow cfz \leq afx + bfy - \mu < cf(z + 1)$$

$$(P_2) \Leftrightarrow cfz \leq dcx + ecy - \nu < cf(z + 1)$$

nous pouvons en déduire que l'intersection des plans  $P_1$  et  $P_2$  est déterminée par

l'ensemble des pavés d'indices  $(z, z)$  pour  $z \in \mathbb{Z}$ , notés  $p_{(z,z)}$ , de l'application quasi-affine  $\phi$  définie comme suit.

$$\phi : \begin{pmatrix} x \\ y \end{pmatrix} \mapsto \left\{ \begin{array}{l} \left[ \frac{afx+bfy-\mu}{cf} \right] \\ \left[ \frac{dcx+ecy-\nu}{cf} \right] \end{array} \right\}$$

$p_{(z,z)} = \left\{ (x, y) \in \mathbb{Z}^2 \mid \phi(x, y) = (z, z) \right\}$  et on a

$$P_1 \cap P_2 = \left\{ (x, y, z) \in \mathbb{Z}^3 \mid (x, y) \in p_{(z,z)} \right\}.$$

Dans sa thèse, Marie-Andrée Jacob [29] propose un algorithme de parcours des pavés d'une application quasi-affine que nous utilisons pour parcourir l'intersection de deux plans discrets.

Une autre caractérisation de l'intersection de deux plans discrets est possible en utilisant des matrices unimodulaires [14].

## 6.4 Rôle de la suite des restes

### 6.4.1 "Erreur" commise

Soit  $P(a, b, c, \mu)$  un plan discret naïf tel que  $0 \leq a \leq b \leq c$  et  $a \wedge b \wedge c = 1$ . Comme

$$P = \left\{ \left( x, y, z = - \left[ \frac{ax + by - \mu}{c} \right] \right) \text{ avec } (x, y) \in \mathbb{Z}^2 \right\}$$

et

$$ax + by - \mu = \left[ \frac{ax + by - \mu}{c} \right] c + \left\{ \frac{ax + by - \mu}{c} \right\} \quad (*)$$

la valeur

$$\rho(x, y) = \left\{ \frac{ax + by - \mu}{c} \right\}$$

exprime exactement la différence ou l'erreur entre l'approximation  $z = - \left[ \frac{ax+by-\mu}{c} \right]$  et la valeur rationnelle exacte.

De plus, (\*) peut s'écrire

$$ax + by - \mu = -cz + \rho(x, y)$$

et par suite nous avons l'identité

$$\rho(x, y) = ax + by + cz - \mu$$

ou encore

$$\boxed{\rho(x, y) = r(a, b, c)(x, y, z) - \mu}$$

avec  $r$  le reste du point  $(x, y, z)$  par rapport à  $P$ .

## 6.4.2 Représentation par restes

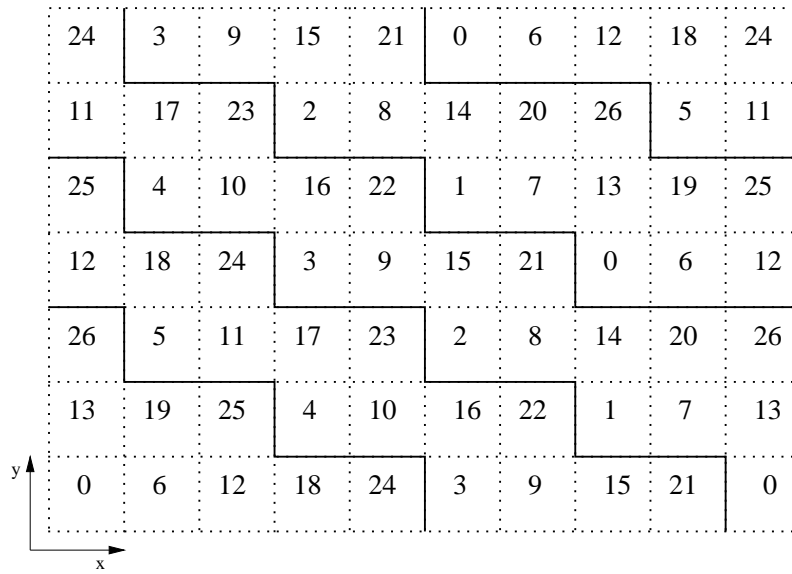
La suite des restes des points de  $P$  nous fournit des informations géométriques sur  $P$ . En effet, soient  $M_0(x_0, y_0, z_0)$  et  $M_1(x_1, y_1, z_1)$  deux points de  $P(a, b, c, \mu)$ , dans ces conditions,

si  $x_0 = x_1$ ,  $y_0 \leq y_1$  et la fonction  $\rho(x, y)$  est croissante pour  $y$  variant dans  $[y_0, y_1]$  et  $x = x_0 = x_1$ , nous avons  $z_0 = z_1$ ,

de même,

si  $y_0 = y_1$ ,  $x_0 \leq x_1$  et la fonction  $\rho(x, y)$  est croissante pour  $x$  variant dans  $[x_0, x_1]$  et  $y = y_0 = y_1$ , nous avons  $z_0 = z_1$ .

Par conséquent en utilisant la relation  $\rho(x, y) = r(a, b, c)(x, y, z) - \mu$  démontrée dans le paragraphe précédent, un plan discret  $P(a, b, c, \mu)$  peut être codé d'une autre façon en pondérant pour tout  $M(x, y, z)$  dans  $P$  le pixel  $(x, y)$  par son reste au point  $M$  par rapport à  $P$ ,  $r(M) = ax + by + cz$ .



**Figure 6.15 :** Représentation par restes du plan  $P(6, 13, 27, 0)$  dans la fenêtre  $[0, 9] * [0, 6]$ .

Nous noterons également par  $r(x, y)$  la valeur du reste attachée au pixel de coordonnées  $(x, y)$ . Les lignes de niveaux de  $P$  s'obtiennent à partir du codage par les restes en isolant les séquences croissantes de la suite  $r(x, y)$  suivant  $x$  et  $y$ . Nous pouvons observer ce principe sur la figure précédente.

Cette représentation par reste d'un plan discret naïf est particulièrement importante dans l'étude des plans discrets car elle fournit à la fois des informations géométriques et arithmétiques.

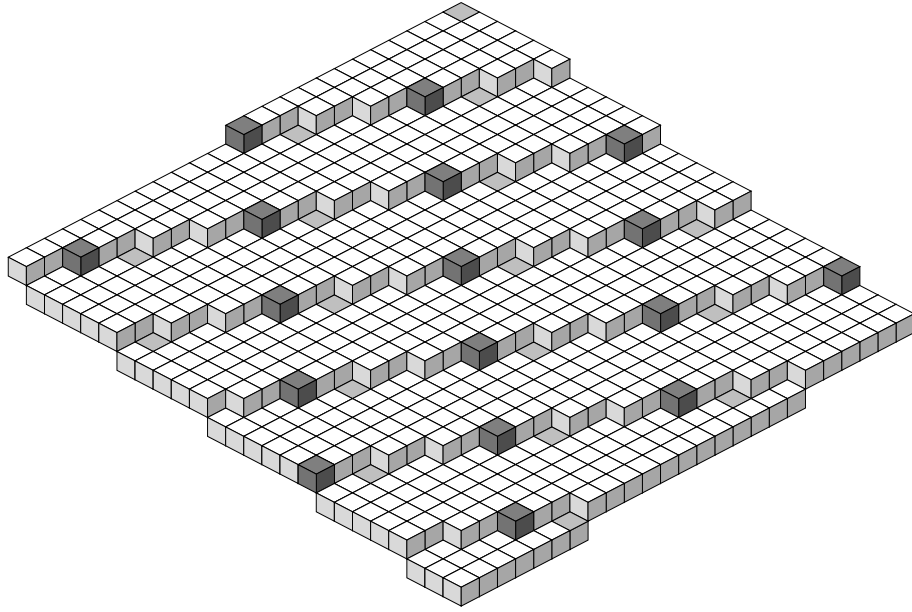
### 6.4.3 Points d'appui et plans d'appui

Soit  $P(a, b, c, \mu)$  un plan discret naïf, les plans réels  $ax + by + cz = \mu$  et  $ax + by + cz = \mu + \max(|a|, |b|, |c|) - 1$  sont appelés les **plans d'appui du plan discret  $P$** . Un voxel du réseau correspondant est appelé un **point d'appui**.

Remarques :

1. Le plan discret  $P$  est l'ensemble des voxels qui sont compris entre les deux plans réels  $ax + by + cz = \mu$  et  $ax + by + cz = \mu + \max(|a|, |b|, |c|) - 1$ .

2. Si  $0 \leq a \leq b \leq c$ , nous dirons que le plan  $ax + by + cz = \mu$  est le **plan d'appui inférieur** de  $P$  et  $ax + by + cz = \mu + c - 1$  le **plan d'appui supérieur** de  $P$ , ils sont notés respectivement  $(P_i)$  et  $(P_s)$ .



**Figure 6.16 :** Points d'appui de  $P(3, 7, 37, 0)$  ; en gris foncé les points d'appui supérieurs et en gris clair les points d'appui inférieurs.

*Théorème 6.7 :*

La distance entre un plan discret naïf  $P(a, b, c, \mu)$  et ses plans d'appui  $(P_i)$  et  $(P_s)$  est inférieure à 1.

Démonstration :

Supposons  $0 < a \leq b \leq c$ ,  $P$ ,  $(P_i)$  et  $(P_s)$  sont fonctionnels, par conséquent, pour tout couple  $(x, y)$  de  $\mathbb{Z}^2$ ,

$$\exists! z \in \mathbb{Z} \text{ tel que } \mu \leq ax + by + cz < \mu + c \quad (*)$$

$$\exists! z_i \in \mathbb{Z} \text{ tel que } ax + by + cz_i = \mu \quad (**)$$

$$\exists! z_s \in \mathbb{Z} \text{ tel que } ax + by + cz_s = \mu + c - 1$$

Par suite,

$$(*) + (**) \Rightarrow \mu - \mu \leq ax + by + cz - (ax + by + cz_i) < \mu + c - \mu$$

$$\Rightarrow 0 \leq c(z - z_i) < c$$

$$\Rightarrow 0 \leq z - z_i < 1$$

ce qui prouve que les points de  $P$  sont situés sur  $(P_i)$  ou au dessus à une distance inférieure à 1.

De manière identique nous démontrons que la distance de  $P$  à  $(P_s)$  est inférieure à 1. ■

*Propriétés :*

Soit  $P(a, b, c, \mu)$  tel que  $0 \leq a \leq b \leq c$ ,

- Les points d'appui supérieurs de  $P$  sont des fins de palier de  $P$
- Les points d'appui inférieurs de  $P$  sont des débuts de palier de  $P$
- La projection d'un point d'appui  $(x_a, y_a, z_a)$  de  $P$  dans le plan  $Oxz$  (resp.  $Oyz$ ) est un point d'appui de la droite naïve correspondante  $\mathcal{D}(a, c, \mu - by_a, c)$  (resp.  $\mathcal{D}(b, c, \mu - ax_a, c)$ ).

Ces propriétés se déduisent immédiatement des propriétés des points d'appui des droites naïves 2D.

#### 6.4.4 Périodicité

Comme

$$\rho(x, y) = \left\{ \frac{ax + by - \mu}{c} \right\},$$

il est clair que la fonction  $\rho$  est doublement périodique,

$$\rho(x + kc, y + lc) = \rho(x, y)$$

nous pouvons alors en déduire qu'un plan discret possède également une propriété de pseudo-périodicité.

remarque :  $\rho$  vérifie aussi  $\rho(x + \alpha, y + \beta) = \rho(x, y)$  pour tout couple  $(\alpha, \beta)$  de  $\mathbb{Z}^2$  tel que  $\alpha a + \beta b = c$ .

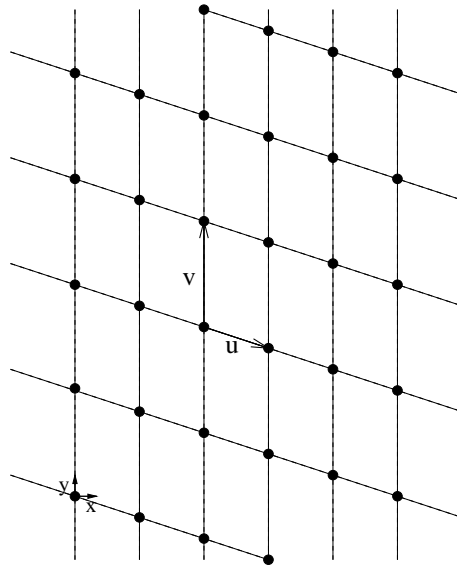
Cette pseudo-périodicité permet d'obtenir plusieurs propriétés de régularité vérifiées par un plan discret. La notion de base d'un réseau d'un plan définie ci-dessous va permettre d'énoncer ces propriétés.

Une **base d'un réseau d'indice  $\alpha$**  d'un plan  $P(a, b, c, \mu)$  est formée par deux vecteurs  $u$  et  $v$  de  $\mathbb{Z}^3$  non colinéaires, présents dans le réseau et tels que dans la projection sur le plan  $Oxy$  de tout parallélogramme engendré par ces deux vecteurs on ne trouve pas d'autre point projeté du réseau d'indice  $\alpha$  de  $P$  mis à part les sommets des parallélogrammes qui sont des points du réseau.

Nous noterons  $\mathcal{B}_{res\alpha}(P)$  une telle base.

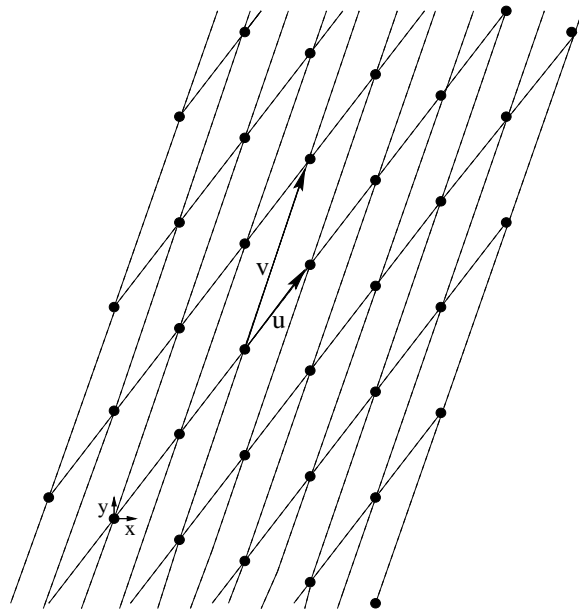


Exemple : Considérons le plan  $P(2, 6, 15, -1)$ , les vecteurs  $u = (3, -1, 0)$  et  $v = (0, 5, -2)$  forment une base du réseau d'indice 0 de  $P$  appelée  $\mathcal{B}_1$ .



**Figure 6.17 :** *Projection dans le plan  $Oxy$  du "treillis" engendré par les vecteurs de la base  $\mathcal{B}_1$  sur les points du réseau d'indice 0 de  $P$ .*

Une autre base  $\mathcal{B}_2$  du réseau d'indice 0 de  $P$  est formée par les vecteurs  $u = (3, 6, -2)$  et  $v = (3, 11, -4)$  qui engendrent un "treillis" différent.



**Figure 6.18 :** Projection dans le plan  $Oxy$  du “treillis” engendré par les vecteurs de la base  $\mathcal{B}_2$  sur les points du réseau d’indice 0 de  $P$ .

*Propriété :*

Une base du réseau d’indice  $\alpha$  de  $P$  est une base de tout réseau de  $P$ .

Cette propriété est une conséquence immédiate du théorème 1 énoncé en 6.2.1. Par suite, une base d’un réseau d’indice  $\alpha$  de  $P$  peut être appelée plus simplement **base de  $P$**  et notée  $\mathcal{B}(P)$ .

*Théorème 6.8 :*

Soit  $\mathcal{B}$  une base du plan  $P(a, b, c, \mu)$  constituée des vecteurs  $u(u_x, u_y, u_z)$  et  $v(v_x, v_y, v_z)$  alors,

$$|c| = \begin{vmatrix} u_x & v_x \\ u_y & v_y \end{vmatrix}$$

$$|b| = \begin{vmatrix} u_x & v_x \\ u_z & v_z \end{vmatrix}$$

$$|a| = \begin{vmatrix} u_y & v_y \\ u_z & v_z \end{vmatrix}$$

En effet, soit  $A$  un point du réseau d’indice  $\alpha$  de  $P$ , on a

$$ax_A + by_A + cz_a = \alpha \tag{1}$$

et par hypothèse,

$$a(x_A + u_x) + b(y_A + u_y) + c(z_a + u_z) = \alpha \quad (2)$$

$$a(x_A + v_x) + b(y_A + v_y) + c(z_a + v_z) = \alpha \quad (3)$$

Or,

$$(2) - (1) \Leftrightarrow \frac{a}{c}u_x + \frac{b}{c}u_y = u_z$$

$$(3) - (1) \Leftrightarrow \frac{a}{c}v_x + \frac{b}{c}v_y = v_z$$

et la résolution du système obtenu par les deux équations précédentes conduit au résultat. ■

*Théorème 6.9 :*

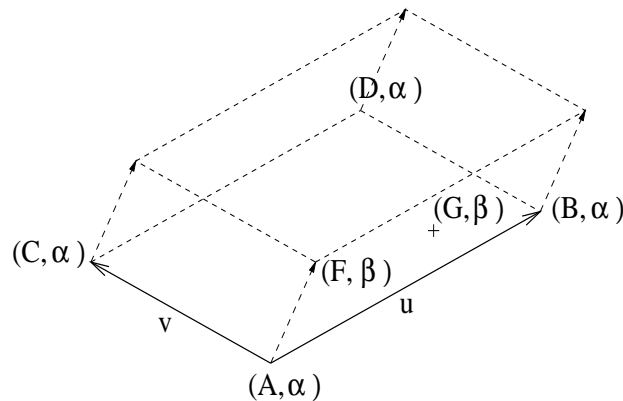
*Soient*

- *un plan  $P(a, b, c, \mu)$  tel que  $0 \leq a \leq b \leq c$ ,*
- *une base  $\mathcal{B}$  de  $P$  formée par les vecteurs  $u$  et  $v$ ,*
- *$A$  un point du réseau d'indice  $\alpha$  de  $P$ ,*

*la projection dans le plan  $Oxy$  de tout parallélogramme engendré à partir de  $A$  par les vecteurs de  $\mathcal{B}$  contient un et un seul point projeté du réseau d'indice  $\beta$  de  $P$  avec  $\beta \neq \alpha$  et appartenant à l'intervalle  $[\mu, \mu + c]$ .*

En effet, soient  $B = (t_u(A))_{Oxy}$ ,  $C = (t_v(A))_{Oxy}$ ,  $D = (t_{u+v}(A))_{Oxy}$ .

- Supposons qu'il existe dans le parallélogramme  $A, B, C, D$  deux points projetés d'indice  $\beta$ , les points  $F$  et  $G$ . Translatons le parallélogramme  $A, B, C, D$  par le vecteur  $\vec{AF}$ , on obtient un parallélogramme formé de quatre points du réseau d'indice  $\beta$  de  $P$  (cf. théorème 1 de 6.2.1) dont nous voyons une illustration sur la figure 6.19. Le point  $G$  d'indice  $\beta$  est à l'intérieur de ce parallélogramme formé par les vecteurs d'une base du réseau d'indice  $\beta$  de  $P$ , ce qui est en contradiction avec la définition d'une base d'un réseau du plan.



**Figure 6.19**

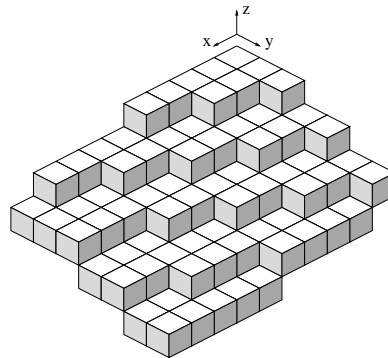
- Dans le parallélogramme  $A, B, C, D$ , il y a  $c$  points car selon le théorème précédent,

$$|c| = \begin{vmatrix} u_x & v_x \\ u_y & v_y \end{vmatrix}$$

or il y a exactement  $c$  indices de réseau de  $P$  possibles variant de  $\mu$  à  $\mu + c - 1$ , par conséquent chaque indice a exactement un représentant dans le parallélogramme  $A, B, C, D$ . ■

### 6.4.5 Un algorithme de construction des faces visibles d'un plan naïf

Pour représenter par des voxels un ensemble de points de  $\mathbb{Z}^3$ , nous associons à chaque point un cube unité. Pour construire cet ensemble de voxels nous devons utiliser des algorithmes qui n'affichent que les faces visibles des voxels et les projettent dans le plan de l'écran ou encore des algorithmes qui construisent et projettent tous les voxels de l'ensemble dans un ordre tenant compte de la position de l'observateur. La représentation d'un plan discret ne nécessite pas l'utilisation de tels algorithmes, en effet nous pouvons "prédire" pour chaque point du plan quelles seront les faces visibles de son voxel associé.

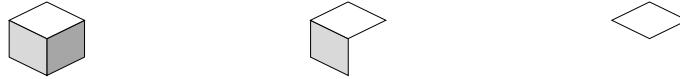


**Figure 6.20 :** Représentation en voxels du plan  $P(6, 13, 27, 0)$  dans le parallélépipède  $[0, 9] * [0, 6] * [0, -4]$ , l'origine est le point le plus proche du repère.

Quand nous observons un plan discret (par exemple celui de la figure précédente), nous voyons trois sortes de voxels dépendant de leur position dans le morceau de plan à représenter :

- les voxels de coin (par exemple le voxel de coordonnées  $(4, 0, 0)$  de  $P(6, 13, 27, 0)$ )
- les voxels de côté (par exemple le voxel de coordonnées  $(1, 3, -1)$  de  $P(6, 13, 27, 0)$ )
- les voxels de milieu (par exemple le voxel de coordonnées  $(1, 2, -1)$  de  $P(6, 13, 27, 0)$ )

Nous allons les représenter selon cette classification ; nous voyons trois faces des voxels de coin, deux des voxels de côté et une seule pour les voxels de milieu.



**Figure 6.21** : De gauche à droite un voxel de coin, un voxel de côté et un voxel de milieu.

La valeur du reste en un point par rapport au plan à construire permet de déterminer très simplement par quel type de voxel ce point doit être représenté. Ce résultat est formalisé par le théorème donné à la page suivante.

*Théorème 6.10 :*

Soit  $P(a, b, c, \mu)$  un plan discret tel que  $0 \leq a \leq b \leq c$ ,  $r(x, y)$  le reste associé au voxel  $V$  dont les deux premières coordonnées sont  $(x, y)$ . Alors,

- $V$  est un voxel de coin si et seulement si  $c - a \leq r(x, y) - \mu < c$
- $V$  est un voxel de côté si et seulement si  $c - b \leq r(x, y) - \mu < c - a$
- $V$  est un voxel de milieu si et seulement si  $0 \leq r(x, y) - \mu < c - b$

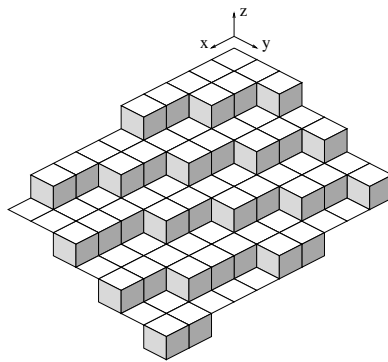
Nous déduisons de ce théorème un algorithme qui est une modification du premier algorithme de parcours donné en 6.3.3.

---

Un algorithme de construction des faces visibles du plan  $P(a, b, c, \mu)$

```
zini = -  $\left\lfloor \frac{a*x_0 + b*y_0 - \mu}{c} \right\rfloor$  ; z = zini ;
rini = a * x0 + b * y0 + c * z -  $\mu$  ; r = rini ;
pour y = y0 à y1 répéter
  pour x = x0 à x1 répéter
    si r < c - b alors Construire le voxel de milieu (x, y, z) ;
    sinon
      si r < c - a alors Construire le voxel de côté (x, y, z) ;
      sinon
        Construire le voxel de coin (x, y, z) ;
        z = z - 1 ;
        r = r - c ;
      fsi
    fsi
    r = r + a ;
  fpour
  si rini  $\geq$  c - b alors faire
    zini = zini - 1 ;
    rini = rini - c ;
  fsi
  rini = rini + b ; z = zini ; r = rini ;
fpour
```

---



**Figure 6.22 :** Construction du morceau de plan  $P(6, 13, 27, 0)$  dans le parallélépipède  $[0, 9] * [0, 6] * [0, -4]$  avec l'algorithme précédent.

Pour obtenir la représentation de la figure 6.20, des tests doivent être ajoutés à l'algorithme précédent pour représenter complètement les voxels bornant le domaine.

## 6.5 Conclusion

Nous avons donné dans ce chapitre plusieurs propriétés des plans discrets naïfs arithmétiques à partir desquelles nous avons déduit des algorithmes très simples et performants de parcours et de construction de plans discrets. La notion de plan discret naïf, analogue à celle des droites discrètes naïves 2D (cf. Chapitres 2 et 3), se révèle tout aussi riche et le codage bidimensionnel d'un plan naïf en facilite beaucoup l'étude. De plus, comme pour les droites, la suite des restes décrit exactement la structure interne d'un plan discret, elle a un rôle primordial dans la reconnaissance des plans discrets, ceci est développé dans les chapitres suivants.

Une étude approfondie des plans discrets d'épaisseurs quelconques reste à réaliser en étendant les propriétés des plans discrets naïfs développées dans ce chapitre aux plans discrets d'épaisseurs quelconques. La notion d'hyperplan discret en dimension quelconque est aussi à déterminer sans aucune référence aux plans réels [59, 62].

## Chapitre 7

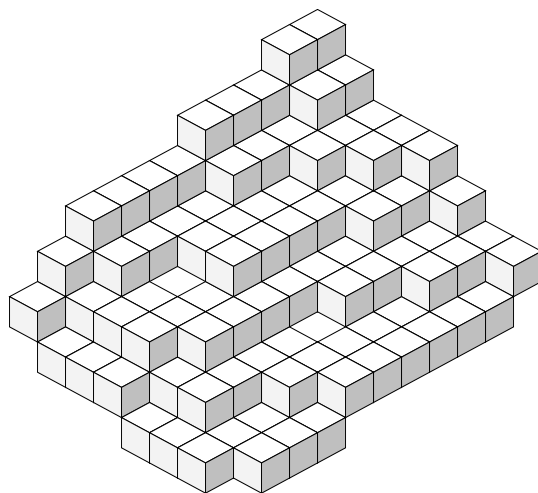
Historique du problème et premier  
algorithme arithmétique de  
reconnaissance de morceaux de plans  
discrets



## 7.1 Introduction

La discrétisation d'un plan ou d'une surface produit d'énormes quantités de voxels. Il est intéressant de pouvoir retrouver les caractéristiques analytiques qui leur ont donné naissance. Cela permet de contrôler ces données *au voxel près* et de leur appliquer des traitements spécifiques : lissage, squelettisation, opérations topologiques, compression . . .

Cette question a intéressé plusieurs auteurs (cf. [20, 35, 38, 59, 62, 63]) et consiste à déterminer si un ensemble de voxels comme celui de la figure 7.1 est ou non un morceau de plan discret.



**Figure 7.1 :** *Est-ce un morceau de plan discret ?*

Les méthodes proposées sont des généralisations des méthodes 2D de reconnaissance de droites discrètes ou bien des méthodes d'analyse numérique.

Dans une première section, sont exposées les méthodes utilisant la construction de l'enveloppe convexe 3D de l'ensemble de points entiers à reconnaître [35, 38, 59]. Ensuite les travaux de P. Velaert [62, 63] utilisant une généralisation de la propriété de régularité (“evenness”) des droites discrètes de Hung [28] sont résumés. Enfin, dans la section 3, nous avons interprété le premier algorithme proposé par Kim [35] en utilisant la définition arithmétique d'un plan discret. Ceci nous a conduit à détecter des cas particuliers négligés dans [35] et élaborer un premier algorithme arithmétique de reconnaissance de morceaux de plans discrets exposé dans ce paragraphe.

## 7.2 Algorithmes reposant sur la construction de l'enveloppe convexe 3D d'un ensemble de points entiers (Kim, Stojmenović et Tošić)

En 1982, dans “Convex digital solid” [36], Kim et Rosenfeld énoncent la **propriété du “triangle corde”** qui est une extension de la propriété de la corde de Rosenfeld [55] :

*Soient  $\omega(x, y, z)$  et  $\omega'(x', y', z')$  deux points de l'espace euclidien de dimension 3. Les deux points sont dits **proches** l'un de l'autre si  $\max\{|x-x'|, |y-y'|, |z-z'|\} < 1$ . Soit  $S$  un solide discret, un point  $\omega$  est dit proche de  $S$  s'il existe un point de  $S$  dont  $\omega$  soit proche.*

*Soient  $d_1, d_2$  et  $d_3$  des points de  $S$ , le triangle de sommets  $d_1, d_2$  et  $d_3$  est appelé “triangle corde” de  $S$ . Un “triangle corde”  $T$  est dit proche de  $S$  si chaque point de  $T$  est proche de  $S$ .  $S$  possède la propriété du “triangle corde” si chaque “triangle corde” de  $S$  est proche de  $S$ .*

Kim, en 1984 dans “Three-dimensional digital planes” [35] donne une définition de plan discret à partir de celle d'une surface discrète :

*Considérons  $S \cap D_{x=i}, S \cap D_{y=j}, S \cap D_{z=k}$ , les ensembles des points entiers situés respectivement sur les plans ( $x = i$ ), ( $y = j$ ) et ( $z = k$ ) avec  $i, j, k$  entiers. Un sous-ensemble 26-connexe de  $\mathbb{Z}^3$  est une **surface discrète** si pour chaque point  $d = (i, j, k)$  de  $S$ , on a les propriétés suivantes :*

*Dans au moins deux des sous-ensembles  $S \cap D_{x=i}, S \cap D_{y=j}, S \cap D_{z=k}$ , le point  $d$  possède au plus deux 8-voisins. Lorsqu'il en possède deux, ceux-ci ne sont pas 8-voisins entre eux. Si pour l'un des deux, disons  $S \cap D_{z=k}$ , le point  $d$  possède plus ou exactement deux 8-voisins qui sont 8-voisins entre eux alors aucun des deux points  $d' = (i, j, k - 1)$  et  $d'' = (i, j, k + 1)$  n'appartient à  $S$ .*

*Soit  $p$  un plan euclidien, chaque fois que  $p$  intersecte une ligne de coordonnée, le point discret le plus proche de l'intersection devient un point de l'image discrète de  $p$ . Si deux points sont possibles, celui situé sur le côté opposé au vecteur normal de  $p$  devient le discrétisé.*

*Un morceau de surface discrète  $R$  est un **morceau de plan discret** s'il existe un sous-ensemble connexe  $g$  d'un plan dont la discrétisation est  $R$ .*

Il montre que la propriété du “triangle corde” n'est ni une condition nécessaire ni une condition suffisante pour qu'un morceau de surface discrète soit un morceau de plan discret. Par contre, il démontre le théorème 7.1 qui est le support de l'algorithme de reconnaissance de morceaux de plans discrets qu'il propose.

*Théorème 7.1 :*

*Une surface discrète finie est un morceau de plan discret si et seulement s'il existe une face de l'enveloppe convexe de la surface discrète telle que la distance entre la surface discrète et le plan support de la face soit inférieure à 1.*

NB : La distance d'un ensemble  $E$  de  $\mathbb{Z}^3$  à un plan  $p$  est définie par la relation suivante :

$$dist(E, p) = \min\{\max_{M \in E}\{dist_x(M, p)\}, \max_{M \in E}\{dist_y(M, p)\}, \max_{M \in E}\{dist_z(M, p)\}\}$$

où,

la  $x$ \_distance d'un point  $M(x, y, z)$  à un plan  $p$  est définie par  $dist_x = |x - x'|$  avec  $(x, y, z')$  un point de  $p$ ,

la  $y$ \_distance et la  $z$ \_distance sont définies de manière similaire à la  $x$ \_distance.

Il en déduit alors l'algorithme suivant qui détermine si un ensemble fini de points discrets est ou non un morceau convexe de plan discret.

---

#### Algorithme *CONVEX\_PLANE(S)*

Soit  $S$  un ensemble fini de points discrets, l'algorithme renvoie vrai si  $S$  est un élément de plan convexe (ses projections sur les plans de coordonnées sont convexes) et faux sinon.

Etape 1 : Tester s'il existe un plan de coordonnées pour lequel la projection de  $S$  est bijective. Si ce n'est pas le cas, arrêt, retourner faux.

Etape 2 : Sur chaque plan de coordonnées, projeter  $S$ , déterminer si la projection est une région discrète convexe, si ce n'est pas le cas, arrêt, retourner faux.

Etape 3 : Construire  $\mathcal{H}(S)$  l'enveloppe convexe de  $S$ .

Etape 4 : Trouver une face de  $\mathcal{H}(S)$  telle que la distance entre  $S$  et  $F$ , le plan support de la face, soit inférieure à 1. Si une telle face existe retourner vrai sinon retourner faux.

---

La complexité de cet algorithme est détaillée et si  $S$  contient  $n^2$  éléments,

- l'étape 1 est en  $O(n^2)$ ,
- le test de la convexité de l'étape 2 est aussi en  $O(n^2)$ ,
- la construction de l'enveloppe convexe des points de  $S$  est en  $O(n^2 \log n)$  en utilisant l'algorithme donné dans [49] par F.P. Preparata et S.J. Hong, la complexité de l'étape 4 est donc en  $O(n^2 \log n)$ ,
- pour chaque face de l'enveloppe convexe construite à l'étape 3, il faut déterminer la distance entre  $S$  et le plan support de la face, ceci est réalisé en  $O(n^2)$ ;

il y a au plus  $n^2$  faces, par conséquent l'étape 4 est réalisée en  $O(n^4)$ .  
 L'algorithme de reconnaissance de morceaux de plans discrets de Kim est donc en  $O(n^4)$ .

En 1991, avec I. Stojmenović [38] ils améliorent l'étape 4 et obtiennent ainsi une complexité en  $O(n^2 \log n)$ . De plus, ils proposent une version parallélisée de leur algorithme.

La même année, avec R. Tošić, I. Stojmenović [59] donne de nouvelles méthodes de discrétisation et de reconnaissance des droites discrètes et hyperplans dans une dimension quelconque. En dimension trois, deux algorithmes de reconnaissance de morceaux de plans discrets sont exposés :

- Une méthode reposant sur la construction de deux enveloppes convexes 3D, de complexité en  $O(N \log N)$ , avec  $N$  le nombre de points de l'ensemble testé.
- Une méthode reposant sur la programmation linéaire en dimension trois de complexité en  $O(N)$ .

Ces deux algorithmes sont issus des définitions et du théorème donnés ci-dessous.

*Un plan  $p$  dans  $\mathbb{R}^3$  est donné par l'équation  $ax + by + cz + d = 0$ , de plus, si  $0 < |a| \leq |b| \leq |c|$ , l'axe  $z$  est appelé l'**axe majeur** de  $p$ .*

La première méthode de discrétisation exposée est équivalente à celle donnée par Kim [35] :

*Considérons  $p$  un plan vérifiant  $0 < |a| \leq |b| \leq |c|$  et  $P(r_1, r_2, p_3)$  le point d'intersection entre  $p$ , et la droite  $d$  parallèle à l'axe majeur de  $p$  d'équation  $x = r_1, y = r_2$  avec  $r_1$  et  $r_2$  entiers. Le point discret  $P' = (r_1, r_2, r_3)$  où  $p_3 - 0.5 < r_3 \leq p_3 + 0.5$  est le point le plus proche de  $P$ .  $P'$  est appelé l'image discrète de  $P$  et l'ensemble des discrétisés des points d'intersection de  $p$  avec les droites parallèles à l'axe majeur de  $p$  est l'**image discrète de  $p$**  et notée  $p'$ .*

Une autre méthode de discrétisation est proposée et énoncée à la page suivante.

*L'image discrète d'un plan  $p$  est l'ensemble de tous les points discrets situés en dessous (le long de l'axe majeur de  $p$ ) des points d'intersection entre le plan  $p''$  ( $p''$  est obtenu en translatant  $p$  de 0.5 le long de l'axe majeur de  $p$ ) et les droites parallèles à l'axe majeur de  $p$  d'équations  $x = r_1, y = r_2$  avec  $(r_1, r_2) \in \mathbb{Z}^2$ .*

*Théorème 7.2 :*

*Un ensemble  $S$  de points de  $\mathbb{Z}^3$  est un morceau de plan discret si et seulement s'il existe un plan  $p''$  qui sépare  $S$  de l'ensemble  $S''$ , avec  $S''$  obtenu en translatant  $S$  de 1 le long de l'axe majeur de  $p''$ .*

Les deux algorithmes ont un “corps” commun donné ci-dessous.

---

Algorithme *Digital\_Hyperplane\_Segments(S)*

Entrée : S un ensemble de n points

Sortie : Vrai si S est un morceau de plan discrétisé et faux sinon

Etape1 : Trouver l'axe candidat pour être l'axe majeur de S

Etape2 : Translater S de 1 le long de son axe majeur pour obtenir l'ensemble S''

Etape3 : Déterminer si S et S'' sont séparables par un plan. S'il existe un tel plan p'' alors S est l'image discrète d'un plan p obtenu en translatant p'' de -0.5 le long de l'axe majeur de S, retourner vrai, sinon faux.

---

Deux différentes méthodes sont proposées pour l'étape 3, la première utilise la construction des enveloppes convexes de S et S'' [49] et un algorithme de détection d'intersection de polyèdres [15] :

---

Procédure *Separability\_ch(S, S'')*

Etape3.1 : Trouver les enveloppes convexes  $\mathcal{H}(S)$  et  $\mathcal{H}(S'')$  de S et S''

Etape3.2 : Déterminer si  $\mathcal{H}(S)$  et  $\mathcal{H}(S'')$  sont séparables par un plan en calculant s'ils s'intersectent ou non. (S et S'' sont séparables si et seulement si  $\mathcal{H}(S)$  et  $\mathcal{H}(S'')$  n'ont pas de point commun.)

---

L'algorithme résultant est en  $O(n \log n)$ .

La seconde méthode pour l'étape 3 utilise des techniques de programmation linéaire et est présentée sur la page suivante.

---

Procédure *Separability\_ch(S, S'')*

Etape3' : Utiliser la programmation linéaire pour tester la séparabilité de S et S''.

[Plusieurs conditions doivent être vérifiées par les points de S :

Supposons que le plan désiré p'' ait pour équation  $ax + by + cz + d = 0$  où  $a, b, c$  et  $d$  doivent être déterminés,

(1) Les points  $(r_1, r_2, r_3)$  de  $S$  sont situés sous  $p''$ , par conséquent ils doivent vérifier

$$ar_1 + br_2 + cr_3 + d \leq 0$$

(2) Les points  $(r_1, r_2, r_3 + 1)$  de  $S''$  sont situés au dessus de  $p''$ , ils doivent donc vérifier

$$ar_1 + br_2 + c(r_3 + 1) + d > 0$$

Les  $2n$  conditions obtenues peuvent être considérées comme une forme spéciale de problème de programmation linéaire à trois variables.]

---

En utilisant les méthodes de résolution proposées dans [18], un algorithme en  $O(n)$  est obtenu.

Les algorithmes proposés ont une complexité assez faible, néanmoins, ils ont un inconvénient majeur pour les applications en n'étant pas incrémentaux, de plus, les méthodes utilisées reposent sur des concepts non triviaux et assez lourds.

### 7.3 Le critère de régularité de P. Veelaert

En 1993, dans “On the flatness of digital hyperplanes” [62], P. Veelaert généralise aux hyperplans de dimensions quelconques la propriété de régularité (“evenness”) des droites discrètes énoncée par Hung [27]. Pour cela, il introduit les notions de planarité (“flatness”) et de régularité en dimension 3 pour caractériser un plan discret. Dans “Digital planarity of rectangular surface segments” [63], il précise les propriétés des plans en dimension 3 et donne un algorithme de reconnaissance de morceaux de plans discrets de forme rectangulaire. Le critère de régularité à l'origine de cet algorithme s'articule sur les définitions qui suivent.

*Soit  $S$  un ensemble de  $\mathbb{Z}^3$ ,  $S$  est dit **planaire** (“flat”) si et seulement s'il existe trois réels  $\gamma_0, \gamma_1, \gamma_2$  tels que  $a_z = \lceil \gamma_0 + \gamma_1 a_x + \gamma_2 a_y \rceil$ , pour tout  $a = (a_x, a_y, a_z) \in S$ .*

*Si un ensemble de points discrets est à la fois planaire et une surface discrète au sens de Kim [35] alors il est appelé **plan discret**.*

Cette définition est équivalente à celle de Kim et il propose cette reformulation plus mathématique :

*Soit  $S$  un ensemble de  $\mathbb{Z}^3$ ,  $S$  est dit **planaire** si et seulement s'il existe trois réels  $\gamma_0, \gamma_1, \gamma_2$  tels que  $|\gamma_0 + \gamma_1 a_x + \gamma_2 a_y - a_z| < 0.5$ , pour tout  $a = (a_x, a_y, a_z) \in S$ .*

Pour déterminer si un ensemble de points est ou non un plan discret, cette définition conduit à établir pour chaque point de l'ensemble une inégalité, et la méthode de Fourier-Motzkin [58] permet de déterminer l'existence des paramètres  $\gamma_0$ ,  $\gamma_1$  et  $\gamma_2$  satisfaisant toutes ces inégalités.

Le critère de régularité de Hung est étendu aux sous-ensembles de  $\mathbb{Z}^3$  :

Soit  $S$  un ensemble de  $\mathbb{Z}^3$ ,  $S$  est dit **régulier** si et seulement si

- a. La projection de  $S$  dans le plan  $Oxy$  est bijective.
- b. Tout quadruplet  $(a, b, c, d)$  de points de  $S$  tels que,  $(a_x - b_x, a_y - b_y) = (c_x - d_x, c_y - d_y)$  vérifie  $|(a_z - b_z) - (c_z - d_z)| \leq 1$ .

P. Veelaert prouve l'équivalence entre la propriété de régularité et celle de planarité pour des ensembles de points ayant une projection particulière dans le plan  $Oxy$ .

Soit  $S$  un sous-ensemble de  $\mathbb{Z}^3$  et  $\mathcal{P}_{xy}(S)$  sa projection dans le plan  $Oxy$ ,

- a. Si  $\mathcal{P}_{xy}(S) = \{(x, y, 0) : r \leq x \leq s \text{ et } u \leq y \leq v\}$  alors  $\mathcal{P}_{xy}(S)$  est appelé un rectangle.
- b. Si  $\mathcal{P}_{xy}(S) = \{(x, y, 0) : r \leq x \leq s \text{ et } u \leq y \leq v\} \cup \{(x, y, 0) : x = s+1 \text{ et } u' \leq y \leq v'\}$ , avec  $u \leq u' \leq v' \leq v$  alors  $\mathcal{P}_{xy}(S)$  est appelé un rectangle partiellement étendu. Bien sûr, en incluant les lignes de points  $x = r - 1$  ou  $y = v + 1$  ou encore  $y = u - 1$  nous obtenons aussi un rectangle partiellement étendu.

*Théorème 7.3 :*

Soit  $S$  un sous-ensemble de  $\mathbb{Z}^3$  tel que  $\mathcal{P}_{xy}(S)$  soit un rectangle ou un rectangle partiellement étendu.  $S$  est planaire si et seulement si  $S$  est régulier.

Le critère de régularité peut être évalué en  $O(n^2)$ , avec  $n$  le nombre de points de  $S$ , par conséquent l'algorithme *Evenness(S)* détermine si un ensemble de points  $S$  ayant une projection en rectangle ou en rectangle partiellement étendu est ou non un morceau de plan discret.

---

### Algorithme *Evenness(S)*

Pour tout point a de S répéter

Pour tout point b de S répéter

$$f_{min}[a_x - b_x, a_y - b_y] = MaxInteger ; f_{max}[a_x - b_x, a_y - b_y] = -MaxInteger ;$$

Fin pour

```

Fin pour
Pour tout point a de S répéter
  Pour tout point b de S répéter
    Si  $f_{min}[a_x - b_x, a_y - b_y] > a_z - b_z$  alors  $f_{min}[a_x - b_x, a_y - b_y] = a_z - b_z$ ; Fsi
    Si  $f_{max}[a_x - b_x, a_y - b_y] < a_z - b_z$  alors  $f_{max}[a_x - b_x, a_y - b_y] = a_z - b_z$ ; Fsi
  Fin pour
Fin pour
plan = Vrai ;
Pour tout point a de S répéter
  Pour tout point b de S répéter
    Si  $f_{max}[a_x - b_x, a_y - b_y] - f_{min}[a_x - b_x, a_y - b_y] > 1$  alors plan = Faux ; Fsi
  Fin pour
Fin pour

```

---

Cet algorithme est très simple et, malgré sa complexité en  $O(n^2)$ , il peut être une solution acceptable pour les petits ensembles ( $n < 100$  selon l'auteur).

## 7.4 Une première approche arithmétique

### 7.4.1 Utilisation de l'algorithme de Kim

Certains des algorithmes précédents utilisent des critères géométriques, en particulier, les algorithmes de Kim et Stojmenović ont besoin de l'enveloppe convexe 3D des ensembles de points étudiés pour trouver une face qui engendre le plan support de l'ensemble considéré. Ceci nous a semblé pouvoir être reformulé plus simplement grâce aux plans discrets définis arithmétiquement (cf. chapitre 6). En effet, un morceau de plan discret étant un objet assez plat, la face retenue dans l'enveloppe convexe par l'algorithme de Kim est nécessairement une face de l'enveloppe convexe supérieure s'appuyant sur des débuts de paliers horizontaux et verticaux ou bien une face de l'enveloppe convexe inférieure s'appuyant sur des fins de paliers horizontaux et verticaux. Or dans un plan discret, grâce à la suite des restes, les débuts et les fins de paliers sont facilement repérables, par conséquent, une première exploration dans cette voie nous a paru utile.

De plus, rappelons le lien entre les plans discrets de Kim et les plans discrets arithmétiques naïfs (cf. chapitre 6, théorème 6.5) :

*Si  $(\mathcal{R})$  est un plan euclidien d'équation  $\alpha x + \beta y + \gamma z = \delta$  avec  $(x, y, z) \in \mathbb{R}^3$  et  $\alpha,$*



$\beta, \gamma, \delta$  des nombres rationnels avec  $\max(|\alpha|, |\beta|, |\gamma|) = |\gamma|$ , alors la discrétisation de  $(\mathcal{R})$  par les points entiers les plus proches est donnée par le plan naïf  $P(a, b, c, \mu)$  où,

$$\alpha = \frac{a}{c}, \beta = \frac{b}{c}, \gamma = 1, \delta = \frac{d}{c},$$

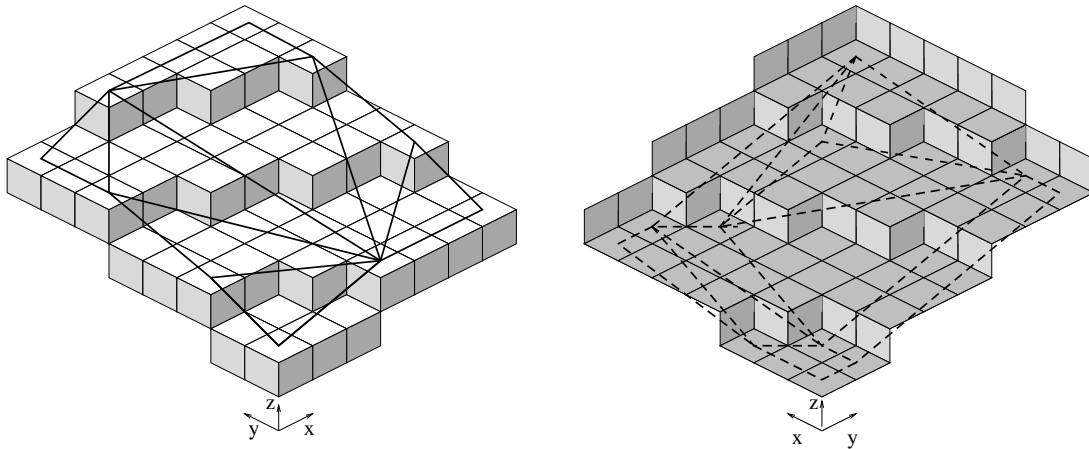
et la borne inférieure  $\mu = d - \lfloor \frac{1}{2}c \rfloor$ .

A la suite de ces constatations, un premier algorithme a été élaboré. Cet algorithme teste tous les triplets de voxels d'un ensemble convexe  $S$  qui sont débuts (resp. fins) de paliers horizontaux et verticaux et regarde si les plans engendrés par ces triplets peuvent avoir pour discrétisation l'ensemble  $S$  de points. En étudiant tous les triplets de tels points, les faces de l'enveloppe convexe supérieure et inférieure sont nécessairement testées. Mais, après de nombreux essais, nous avons constaté que dans certains morceaux de plans discrets il n'existait pas de triplet de voxels, donc de face sur l'enveloppe convexe de  $S$ , tel que la distance entre  $S$  et le plan support de la face soit inférieure à 1.

### 7.4.2 Un exemple de cas négligé par Kim

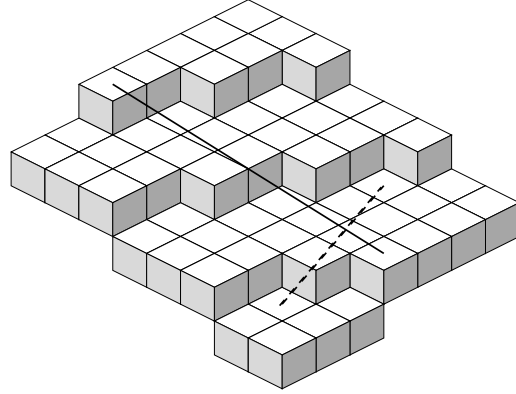
Considérons le plan discret de Kim issu du plan euclidien d'équation  $5x + 9y - 29z = 0$ , il correspond au plan discret naïf  $P(5, 9, -29, -14)$ . Le morceau  $S$  tel que  $0 \leq x \leq 6$  et  $0 \leq y \leq 7$  est étudié.

Sur la partie gauche de la figure 7.2, nous voyons la représentation par voxels vue du dessus de ce morceau de plan discret avec l'enveloppe convexe supérieure de ses points. A droite, sa vue du dessous avec l'enveloppe convexe inférieure.



**Figure 7.2**

Aucune face sur l'enveloppe convexe supérieure ou inférieure des points du morceau n'est telle que la distance entre  $S$  et le plan support de la face soit inférieure à 1. Mais en examinant toutes les arêtes présentes sur l'enveloppe convexe, deux d'entre elles, une sur l'enveloppe supérieure et l'autre sur l'enveloppe inférieure forment une base du plan  $P(5, 9, -29, -14)$  (cf. Figure 7.3).



**Figure 7.3**

En général, si une face de l'enveloppe convexe ne convient pas il faut chercher deux arêtes pour répondre au problème posé.

### 7.4.3 Reformulation du théorème de Kim et premier algorithme arithmétique de reconnaissance de morceaux de plans discrets

Suite à l'exemple donné dans la sous-section précédente, nous sommes amenés à reformuler le théorème 7.1 de Kim :

- S*, une surface discrète finie, est un morceau de plan discret si et seulement si
- il existe une face  $F$  de l'enveloppe convexe de la surface discrète telle que la distance entre  $S$  et le plan support de  $F$  soit inférieure à 1.
  - ou
  - il existe 2 arêtes appartenant à l'enveloppe convexe de la surface discrète telles que la distance entre  $S$  et le plan engendré par les deux arêtes soit inférieure à 1.

Nous obtenons alors l'algorithme, présenté ci-dessous, de reconnaissance de morceaux convexes de plans discrets naïfs dans les 48ièmes d'espace tels que  $0 \leq |a| \leq |b| \leq |c|$ . Cet algorithme cherche tous les vecteurs distincts ayant deux extrémités sur deux débuts (ou deux fins) de paliers horizontaux et verticaux. Puis, chaque couple de deux de ces vecteurs est testé comme base possible d'un plan support de l'ensemble S.

---

Algorithme *Rec\_Morc\_Plan(S)*

Entrée : un ensemble convexe S de voxels

Sortie : Vrai si S est un morceau de plan discret naïf (de plus les caractéristiques du plan discret naïf support sont calculées) et Faux sinon

Etape 0 : initialisation des caractéristiques de l'éventuel plan support,

$$a_S = 0; b_S = 0; c_S = \text{MaxInteger}; \mu_S = 0;$$

Etape 1 : extraire tous les vecteurs distincts entre deux débuts de paliers

extraire tous les vecteurs distincts entre deux fins de paliers

Etape 2 : pour tout couple de vecteurs obtenus  $(u, v)$ ,

– calculer les caractéristiques  $a, b, c$  engendrées par  $(u, v)$  (cf. chapitre 6.) :

$$|c| = \begin{vmatrix} u_x & v_x \\ u_y & v_y \end{vmatrix}$$

$$|b| = \begin{vmatrix} u_x & v_x \\ u_z & v_z \end{vmatrix}$$

$$|a| = \begin{vmatrix} u_y & v_y \\ u_z & v_z \end{vmatrix}$$

– vérifier si les caractéristiques obtenues correspondent à un plan support de l'ensemble S en calculant en chaque point  $(x, y, z)$  de S la valeur  $ax + by + cz$ ; la valeur minimale  $min$  et la valeur maximale  $max$  obtenues doivent vérifier la relation  $min + |c| - 1 = max$ . Si c'est le cas, les caractéristiques correspondent à S.

Si  $|c| < |c_S|$  alors  $a_s = a; b_s = b; c_s = c; \mu_s = min$ ; Fsi

Etape 3 : Si  $c_S \neq \text{MaxInteger}$  retourner Vrai; sinon retourner Faux; Fsi

---

Cet algorithme détermine si un ensemble de voxels est un morceau de plan discret et, si c'est le cas, il donne les caractéristiques  $a, b, c, \mu$  ayant la plus petite composante  $c$  possible pour que les points  $(x, y, z)$  de S vérifient  $\mu \leq ax + by + cz < \mu + |c|$ . Les points de S appartiennent au plan discret naïf  $P(a, b, c, \mu)$ .

Exemple :

Reprenons l'exemple précédent (figures 7.2 et 7.3), sur la figure 7.4, nous voyons la représentation par lignes de niveaux du morceau du plan discret  $P(5, 9, -29, -14)$ , les projections des points débuts ou fins de paliers à considérer dans l'algorithme sont indiqués.

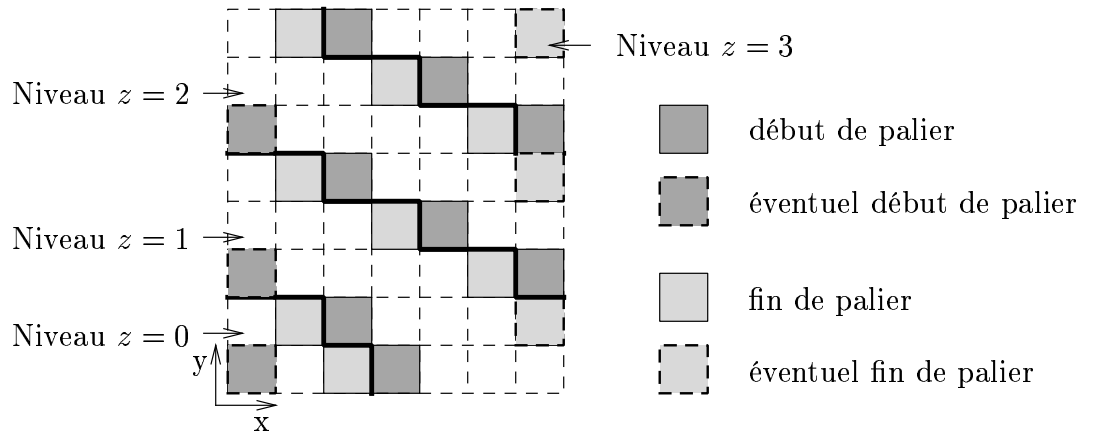


Figure 7.4

Appliquons l'algorithme  $Rec\_Morc\_Plan(S)$  à l'ensemble  $S$  des voxels du morceau de plan représenté sur la figure précédente pour retrouver les caractéristiques du plan support de  $S$  ayant la composante  $c$  minimale. Un seul couple de vecteurs engendre les caractéristiques d'un plan support de  $S$ . Les vecteurs de coordonnées  $(-1, 7, 2)$  et  $(4, 1, 1)$  présents sur la figure 7.5 fournissent les caractéristiques  $a = 5$ ,  $b = 9$  et  $c = -24$ .

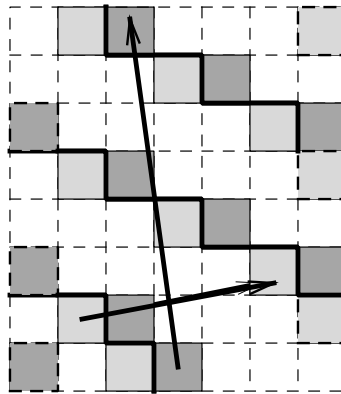


Figure 7.5

Voici la représentation par restes de  $S$  en fonction de  $a$ ,  $b$  et  $c$  :

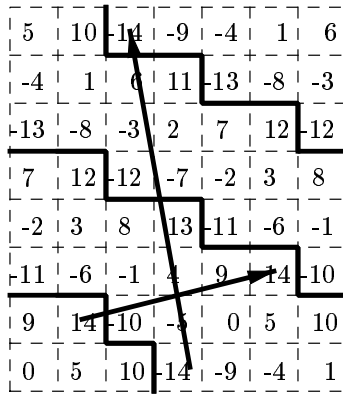


Figure 7.6

La valeur  $ax + by + cz$  est comprise dans  $S$  entre  $-14$  et  $14$ , par conséquent nous pouvons conclure que  $S$  est un morceau du plan discret  $P(5, 9, -24, -14)$ .

Nous pouvons aussi remarquer que les extrémités des vecteurs retenus appartiennent pour l'un des deux au pointillé borne supérieure de  $P$  et pour l'autre au pointillé borne inférieure de  $P$ ; nous obtenons *deux points d'appui supérieurs et deux points d'appui inférieurs*. Pour la majorité des exemples traités nous retrouvons cette propriété et quand les deux vecteurs ont une extrémité commune (ce qui correspond à une face de l'enveloppe convexe), cette propriété est aussi vérifiée, les extrémités des vecteurs sont *trois points d'appui inférieurs ou trois points d'appui supérieurs*.

Regardons un autre exemple, considérons sur la figure suivante un morceau du plan discret naïf  $P(4, 11, -21, -10)$

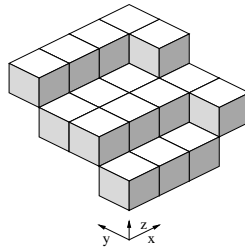


Figure 7.7

Soit  $S$  l'ensemble de voxels représentés sur la figure 7.7, l'algorithme  $Rec\_Morc\_Plan(S)$  fournit les caractéristiques  $a = 1$ ,  $b = 3$ ,  $c = -6$ ,  $\mu = -3$  calculées à partir des vecteurs  $u(3, -1, 0)$  et  $v(3, 1, 1)$ . Voici la représentation par restes de  $S$  en fonction de  $a, b, c$ :

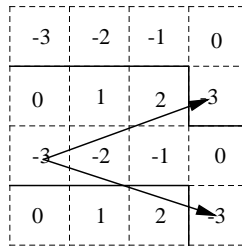


Figure 7.8

$S$  est un morceau du plan  $P'(1, 3, 6, -3)$ ,  $P'$  étant le plan avec la composante  $c$  minimale pour être un plan support de  $S$ . Nous remarquons ici aussi que les points extrémités des vecteurs de base sont *trois points d'appui inférieurs*.

Ces remarques sur le rôle des points d'appui dans la reconnaissance des morceaux de plans discrets sont primordiales pour l'élaboration de l'algorithme incrémental présenté dans le chapitre suivant.

## 7.5 Conclusion

Nous avons décrit dans ce chapitre quelques unes des méthodes utilisées et décrites dans les publications des dernières années, aucun algorithme incrémental n'a jamais été élaboré. La définition arithmétique des plans discrets nous a permis, en nous inspirant des résultats antérieurs, d'obtenir un premier algorithme de reconnaissance de morceaux convexes de plans discrets naïfs, systématique, capable de tester tous les vecteurs normaux possibles en s'appuyant sur les codages par niveaux et par restes. La complexité de ce premier algorithme arithmétique est importante mais, grâce à lui, nous avons pu explorer la reconnaissance de morceaux de plans discrets naïfs sur de nombreux exemples. Ceci nous a permis de dégager les premiers résultats théoriques et d'élaborer une version géométrique incrémentale très performante présentée dans le chapitre 8.

# Chapitre 8

## Reconnaissance incrémentale des plans discrets naïfs

## 8.1 Introduction

Dans ce chapitre nous présentons un algorithme incrémental, très performant, de reconnaissance de morceaux de plans discrets. Cet algorithme répond au problème suivant :

*Soit  $\mathcal{V}$  un sous-ensemble 18-connexe, borné et convexe de  $\mathbb{Z}^3$  se projetant bijectivement sur le plan  $Oxy$ ; soit  $\Pi$  cette projection. Existe-t-il quatre paramètres entiers  $0 \leq a \leq b \leq c$  et  $\mu$  tels que l'ensemble des solutions de la double inégalité diophantienne*

$$\begin{cases} \mu \leq ax + by + cz < \mu + c \\ (x, y) \in \Pi \end{cases}$$

*soit égal à  $\mathcal{V}$  ? En d'autres termes,  $\mathcal{V}$  est-il un morceau de plan discret ?*

Notre approche du problème utilise la définition arithmétique des plans discrets. A cet effet, la section 2 donne les notions utilisées dans la reconnaissance des plans discrets et nous proposons, au paragraphe 4, un algorithme incrémental analogue à celui que nous avons découvert pour les droites (cf. chapitre 3).

Bien que la définition d'un plan discret (cf. [2]) soit une généralisation évidente de celle des droites discrètes (cf. [50]), il est très difficile de comprendre les structures abstraites des sous-ensembles de  $\mathbb{Z}^3$ , même après une représentation graphique en voxels. Cette exploration, quasiment impossible à conduire *à la main*, a nécessité une étape expérimentale sur ordinateur. Les premiers résultats théoriques ont été obtenus à l'aide d'un premier algorithme présenté dans le chapitre précédent (section 7.4), systématique et capable de tester toutes les caractéristiques possibles en s'appuyant sur les codages par niveaux et par restes (cf. chapitre 6). Puis, grâce à lui, nous avons pu élaborer la version géométrique incrémentale de la section 4.

Précisons les traits essentiels de cet algorithme. Soit  $V$  un ensemble *convexe* de voxels (cf. [33, 36, 37]) à reconnaître, nous parcourons ses sections parallèles à un plan de coordonnées, en ajoutant les voxels un à un. Après plusieurs changements de caractéristiques, nous obtenons celles d'un plan ou une impossibilité. L'algorithme utilise des propriétés géométriques des plans discrets données dans la section 3 et énoncées sous forme de conjectures. Cette partie théorique est illustrée de nombreux exemples de morceaux de plans discrets représentés selon le besoin de visualisation par voxels, niveaux ou par restes (cf. chapitre 6).

Le coeur de l'algorithme utilise essentiellement la détermination de l'enveloppe convexe *plane* de deux ensembles de points liés à  $V$ . Au fil de la reconnaissance ces enveloppes convexes 2D évoluent en fonction du point ajouté au morceau de plan courant. Les constructions effectuées sont réalisées grâce à des notions de géométrie 2D, par conséquent, la complexité de cet algorithme est très faible, en tout cas bien inférieure à celle des techniques proposées dans [35, 59, 62] où la reconnaissance



des plans discrets repose sur la construction d’enveloppes convexes 3D d’ensembles de voxels, opération délicate et coûteuse.

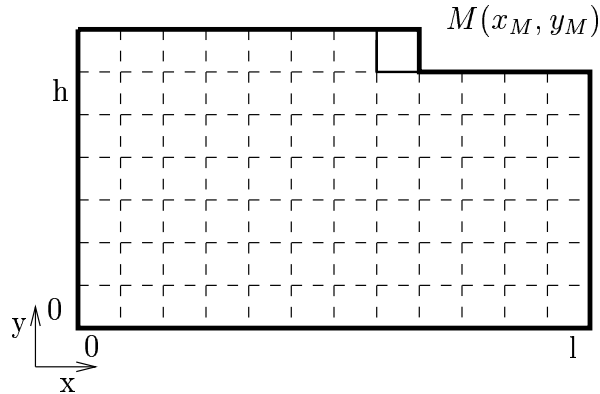
## 8.2 Notions fondamentales pour la reconnaissance de morceaux de plans discrets

Dans tout ce paragraphe, nous considérons un plan discret naïf  $P(a, b, c, \mu)$  dont les caractéristiques vérifient  $0 \leq a \leq b \leq c$ .

### 8.2.1 Morceau reconnu d’un plan discret

Nous avons besoin de préciser les sous-ensembles de  $\mathbb{Z}^3$  qui nous intéressent.

Un *morceau de plan* est un sous-ensemble convexe de voxels d’un plan discret naïf (i.e. ses projections sur les plans de coordonnées sont convexes [33, 36, 37]). En particulier, dans l’algorithme présenté en 8.4, à chaque étape de l’algorithme grâce au parcours par balayage des sections en  $y$ , la projection dans le plan  $Oxy$  des morceaux de plan a la forme rectangulaire “partiellement étendue” suivante,  $M(x_M, y_M)$  étant le dernier point ajouté dans le cadre de l’algorithme de reconnaissance :



**Figure 8.1 :** Représentation dans le plan  $Oxy$  d’un morceau de plan rectangulaire “partiellement étendu” tel que  $x \in [0, l]$  et  $y \in [0, h]$  ou  $x \in [0, x_M]$  et  $y = y_M$ .

Nous dirons qu’un morceau du plan  $P(a, b, c, \mu)$  est *reconnu* s’il possède au moins quatre points d’appui dans une des configurations données sur la page suivante.

- Trois points d'appui supérieurs (resp. inférieurs) et un point d'appui inférieur (resp. supérieur).

Exemple :

18	6	15	3	12	0	9	18	6
5	14	2	11	20	8	17	5	14
13	1	10	19	7	16	4	13	1
0	9	18	6	15	3	12	0	9

**Figure 8.2 :** Représentation par restes du morceau reconnu du plan  $P(9, 13, 21, 0)$  : trois points d'appui inférieurs en gris clair et un point d'appui supérieur en gris foncé.

- Deux points d'appuis supérieurs et deux points d'appuis inférieurs.

Exemple :

15	18	21	2	5	8	11	14	17
10	13	16	19	0	3	6	9	12
5	8	11	14	17	20	1	4	7
0	3	6	9	12	15	18	21	2

**Figure 8.3 :** Représentation par restes du morceau reconnu du plan  $P(3, 5, 22, 0)$ .

Ces configurations seront appelées par la suite respectivement **CAS3.1** et **CAS2.2**. Les quatre points d'appui engendrent 2 vecteurs, ce qui permet de définir de manière unique les plans d'appui ; ces deux vecteurs  $V = (v_x, v_y, v_z)$  et  $W = (w_x, w_y, w_z)$  forment une base de P et on a (cf. 6.4.4),

$$a = |v_y w_z - v_z w_y|$$

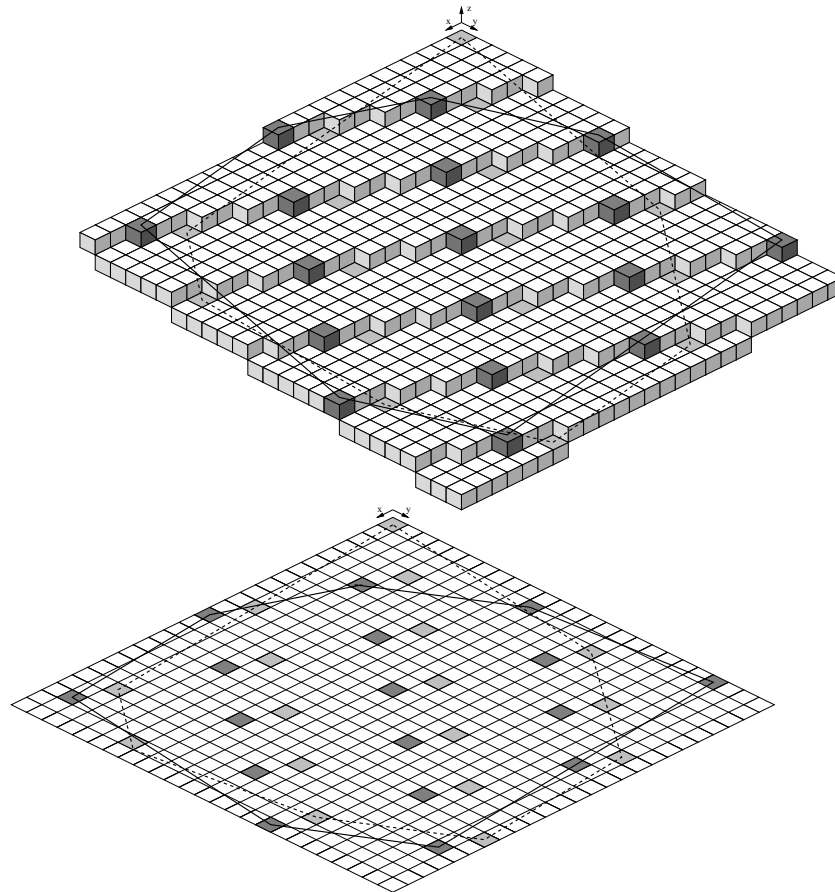
$$b = |v_x w_z - v_z w_x|$$

$$c = |v_y w_x - v_x w_y|$$

Dans “Three-dimensional digital planes” [35], Kim propose un algorithme de reconnaissance de plans discrets qui repose sur la construction de l’enveloppe convexe 3D de l’ensemble de voxels  $V$  à reconnaître. Une des faces de l’enveloppe convexe obtenue doit engendrer les caractéristiques du plan auquel appartiennent les voxels de  $V$ . Mais il néglige les cas où seule la configuration du CAS2.2 est présente et son algorithme perd en efficacité.

### 8.2.2 Polygones d’appui

Le *polygone d’appui supérieur* (resp. *inférieur*) d’un morceau reconnu du plan  $P$  est formé par les voxels dont la projection dans le plan  $0xy$  appartient à l’enveloppe convexe 2D des projections des points d’appui supérieurs (resp. inférieurs), il sera noté  $PS$  (resp.  $PI$ ).



**Figure 8.4 :** *Polygones d’appui d’un morceau reconnu du plan  $P(3, 7, 37, 0)$ .*

Remarque : Les polygones d'appui dépendent très fortement de la géométrie du morceau de plan considéré.

### 8.2.3 Point $k$ -extérieur à un morceau reconnu de plan discret

Un point  $M(x, y, z)$  de  $\mathbb{Z}^3$  est  **$k$ -extérieur** à un morceau reconnu du plan  $P(a, b, c, \mu)$  si  $r(M) = \mu - k$  ou  $r(M) = \mu + c + k - 1$  pour  $k \in \mathbb{Z}$  et  $k \geq 1$  avec  $r$  le reste en fonction de  $P$  défini au point  $M$  par  $r(M) = ax + by + cz$ .

Un point  $M$  de  $\mathbb{Z}^3$ ,  $k$ -extérieur avec  $k > 1$  à un morceau reconnu  $S$  du plan  $P(a, b, c, \mu)$ , est dit **fortement extérieur** à  $S$ .

Il est clair si  $r(M) = \mu - k$  que le point  $M$  se situe "en dessous" du plan  $P$  et, si  $r(M) = \mu + c + k - 1$ , le point se situe "au dessus" du plan  $P$ .

La reconnaissance d'un morceau de plan discret consiste à ajouter les voxels les uns après les autres en parcourant ses sections parallèles à un plan de coordonnées pour obtenir à chaque ajout un morceau reconnu de plan discret, pour cela, quand le point  $M$  ajouté est  $k$ -extérieur avec  $k \geq 1$ , plusieurs constructions géométriques doivent être réalisées et les voxels clés du morceau de plan sont extraits. Toutes les constructions géométriques reposent sur les polygones d'appui et selon leur position par rapport à  $M$ , ils échangent leurs fonctions :

- Si  $r(M) < \mu$ , ( $Ps$ ) est le plan d'appui le plus éloigné de  $M$ , on appelle alors le polygone d'appui supérieur PS le **convexe des pivots**, noté CVP, et ( $Pi$ ) étant le plan d'appui le plus proche de  $M$ , le polygone d'appui inférieur PI est appelé le **convexe des antipodes**, noté CVA, on pose CVP=PS et CVA=PI.
- De manière symétrique, si  $r(M) \geq \mu + c$ , le polygone d'appui supérieur PS est désigné par le nom de **convexe des antipodes**, noté CVP, et le polygone d'appui inférieur PI par l'appellation de **convexe des pivots**, noté CVA, on pose CVA=PS et CVP=PI.

Ces deux convexes ne contiennent qu'un petit nombre de voxels par rapport à ceux du morceau parcouru.

En fonction du point  $M$   $k$ -extérieur ajouté, nous allons extraire des voxels caractéristiques du convexe des pivots et du convexe des antipodes (en général 3).

## Ligne polygonale des vecteurs de pivot

La *ligne polygonale des vecteurs de pivot*, notée  $L$  se construit en fonction du point  $M$ ,  $k$ -extérieur, ajouté au morceau reconnu de plan et du convexe des pivots qui lui est associé ; toutes les constructions géométriques sont réalisées sur les projections dans le plan  $Oxy$  des voxels du convexe des pivots du morceau reconnu, quatre cas sont possibles :

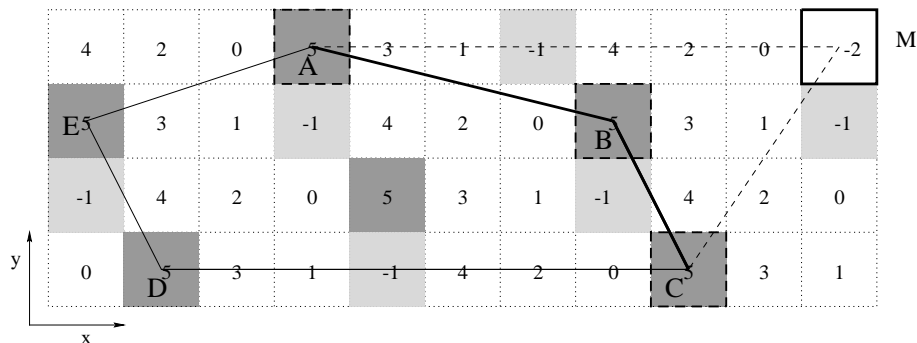
1. *Le CVP est formé par un point alors la ligne polygonale des vecteurs de pivot sera réduite à ce point.*
2. *Le CVP est formé par des points colinéaires et*
  - a. *Dans le plan  $Oxy$  la projection de  $M$  est colinéaire aux points projetés du CVP, alors la ligne polygonale des vecteurs de pivot sera formée par le point du CVP le plus proche de  $M$ ,*
  - b. *Dans le plan  $Oxy$  la projection de  $M$  n'est pas colinéaire aux points projetés du CVP, alors la ligne polygonale des vecteurs de pivot est formée par tous les points du CVP.*
3. *Le CVP est formé par des points non tous colinéaires. Dans ce cas, la ligne polygonale des vecteurs de pivot est composée par les points du CVP qui forment dans le plan  $Oxy$  des arêtes de l'enveloppe convexe 2D (composée des projetés des points du CVP) qui sont supprimées de cette enveloppe convexe 2D par l'ajout du projeté du point  $M$ .*

Les vecteurs présents sur la ligne polygonale des vecteurs de pivot sont appelés *vecteurs de pivot*.

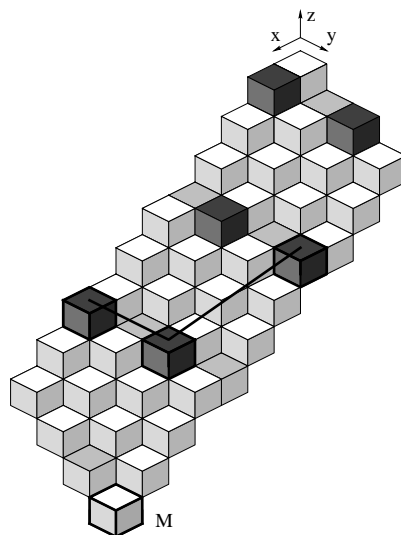
Remarque : Après de multiples expérimentations nous avons constaté qu'un polygone d'appui d'un morceau rectangulaire de plan discret construit de la manière décrite précédemment contenait au plus trois vecteurs linéairement indépendants.

Des exemples de construction de la ligne polygonale des vecteurs de pivot sont proposés dans les deux pages suivantes.

- Illustration du point 3 sur les figures 8.5 et 8.6, en trait fin la projection dans le plan  $Oxy$  du convexe des pivots associé au point  $M(10, 3, -10)$ , point 1-extérieur au morceau reconnu du plan  $P(5, 6, 7, -1)$  et, en plus épais, la projection de la ligne polygonale des vecteurs de pivot  $L$ .



**Figure 8.5 :** Ici  $r(M) = -2$  donc  $CVP=PS$  et sa projection dans le plan  $Oxy$  est composée des points  $A, B, C, D$  et  $E$ . Les points  $A, B, C$  sont les projetés des points de  $L$ , la ligne polygonale des vecteurs de pivot.



**Figure 8.6 :** Représentation voxel du morceau reconnu de plan précédent avec la ligne polygonale des vecteurs de pivot,  $L$ , formée des voxels noirs dont les contours sont épais.

- Illustration du point 2.a sur la figure ci-dessous, à un morceau reconnu de  $P(4, 5, 9, 0)$  est ajouté un point 1-extérieur de coordonnées  $(5, 3, -4)$ ,  $r(M) = -1$ . Le CVP (=PS) est formé de trois points colinéaires à  $M$ , la ligne des vecteurs de pivot L est réduite au point de coordonnées  $(5, 3, -4)$ .

6	1	5	0	4	-1	M				
1	5	0	4	8	3	7	2	6	1	5
5	0	4	8	3	7	2	6	1	5	0
0	4	8	3	7	2	6	1	5	0	4

Figure 8.7

- Illustration du point 2.b, nous trouvons sur la figure 8.8 un morceau reconnu du plan  $P(2, 8, 13, -8)$ , le point  $M$  de coordonnées  $(6, 4, -3)$  est 1-extérieur,  $r(M) = 5$ . Le CVP (=PI) est formé de trois points colinéaires mais  $M$  n'appartient pas à cette droite, par conséquent, ces trois points forment la ligne polygonale des vecteurs de pivot.

-7	-5	-3	-1	1	3	5	M			
-2	0	2	4	-7	-5	-3	-1	1	3	
3	-8	-6	-4	-2	0	2	4	-7	-5	
-5	-3	-1	1	3	-8	-6	-4	-2	0	
0	2	4	-7	-5	-3	-1	1	3	-8	

Projection de PI

Figure 8.8

## Antipodes d'un vecteur de pivot et ligne polygonale des antipodes

Une *antipode*  $A$  d'un vecteur de pivot  $W$  par rapport au point  $M$   $k$ -extérieur à un morceau reconnu du plan  $P$  est un sommet du convexe des antipodes associé à  $M$  tel que, dans le plan  $Oxy$ , le projeté de  $A$  ait la distance maximale à la droite passant par le projeté de  $M$  et de vecteur directeur le projeté de  $W$ . Nous noterons  $A = ant(W)$ .

De plus, si les deux sommets de  $W$  sont de part et d'autre de la droite  $AM$ ,  $A$  est dite *séparante*.

Remarque : Bien sûr, le maximum est atteint au même point en considérant la distance à la droite de vecteur directeur le projeté de  $W$  passant par un sommet de  $L$  extrémité du vecteur  $W$ . Pour des raisons algorithmiques, nous avons préféré la définition proposée ci-dessus.

Dans le morceau reconnu du plan  $P(5, 6, 7, -1)$  de la figure 8.9, le point  $M$  de coordonnées  $(10, 3, -10)$  est 1-extérieur,  $A$  est antipode des vecteurs de pivot  $W_1$  et  $W_2$  mais n'est séparante que pour le vecteur  $W_1$ .

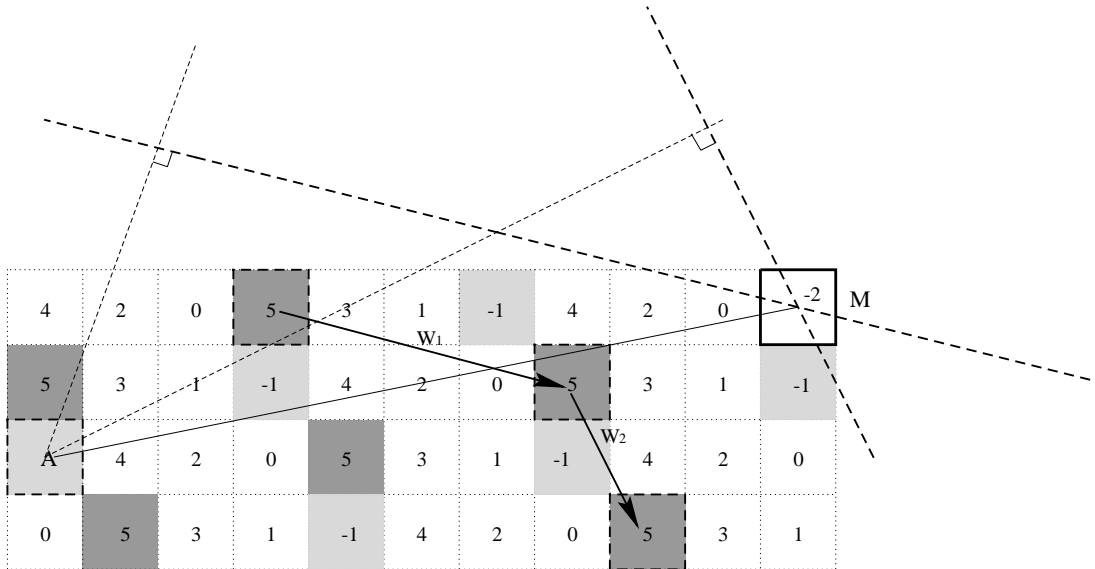
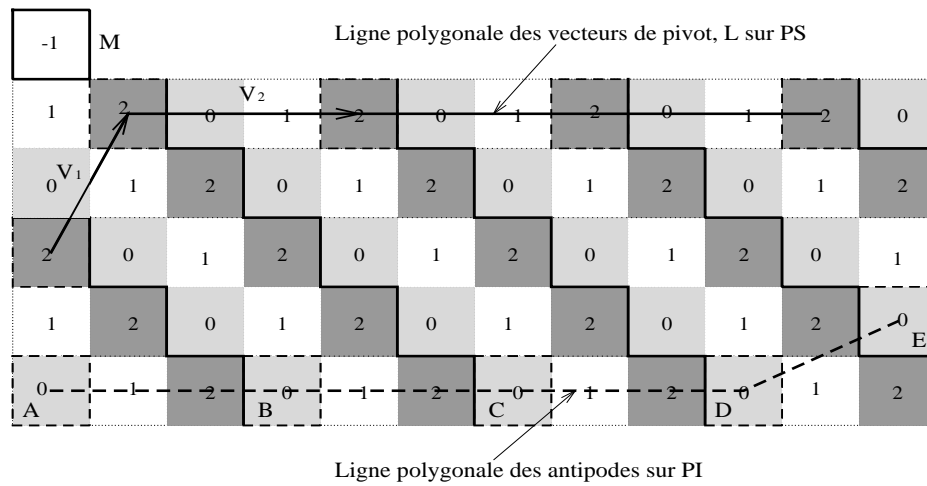


Figure 8.9

L'ensemble des antipodes des vecteurs de la ligne polygonale des vecteurs de pivot forme *la ligne polygonale des antipodes*.





**Figure 8.10**

Sur la figure 8.10, le point  $M(0, 5, -2)$  a été ajouté à un morceau reconnu du plan  $P(1, 1, 3, 0)$ ,  $r(M) = -1$ , le convexe des pivots est PS et le convexe des antipodes est PI. La ligne polygonale des vecteurs de pivot comporte 2 vecteurs distincts,  $V_1$  et  $V_2$ ,  $E$  est antipode de  $V_1$  et  $A, B, C, D$  sont antipodes de  $V_2$ , seules  $C$  et  $D$  sont séparantes. Sur cette représentation par restes, les projections des voxels de la ligne polygonale des vecteurs de pivot sont reliées par un trait plein et les projections des voxels de la ligne polygonale des antipodes par un trait pointillé.

## 8.3 Résultats théoriques

### 8.3.1 Le problème incrémental

Soient  $V$  un ensemble convexe de voxels à reconnaître,  $S$  un morceau reconnu du plan  $P(a, b, c, \mu)$ , PS et PI les polygones d'appui de  $S$ .

Considérons un point  $M(x, y, z)$  de  $V$  tel que  $S' = S \cup \{M\}$  soit convexe et  $r(M) = ax + by + cz$ , plusieurs cas sont alors possibles :

1. Si  $\mu \leq r(M) < \mu + c$ ,  $M$  appartient à  $P$  et  $S' = S \cup \{M\}$  est un morceau reconnu de  $P$ .
2. Si  $r(M) < \mu$  ou  $r(M) \geq \mu + c$ ,  $M$  n'appartient pas à  $P$ , mais si  $S' = S \cup \{M\}$  est un morceau reconnu du plan  $P'(a', b', c', \mu')$ , il faut déterminer comment obtenir les nouvelles caractéristiques.

A cet effet, nous avons établi plusieurs conjectures fondées sur des notions de géométrie 2D dans lesquels les polygones d'appui jouent un rôle important. Ils échangent

leurs fonctions selon la position du point  $M$  k-extérieur ajouté en étant convexe des pivots (CVP) ou convexe des antipodes (CVA) de  $M$ . Ces conjectures ont été largement testées sur de nombreux exemples.

### 8.3.2 Ajout d'un point 1-extérieur à un morceau reconnu $S$ du plan $P$

*Conjecture 8.1 :*

*Soient  $S$  un morceau reconnu du plan  $P(a, b, c, \mu)$  et un point  $M(x, y, z)$  de  $\mathbb{Z}^3$  tel que  $S' = S \cup \{M\}$  soit convexe. Si  $M$  est 1-extérieur à  $S$  alors  $S' = S \cup \{M\}$  est un morceau reconnu d'un plan  $P'(a', b', c', \mu')$ .*

Dans le cas où le point  $M$  ajouté à  $S$  est 1-extérieur, les conjectures énoncées dans la paragraphe a vont permettre de construire dans tous les cas une base  $\{U, W\}$  du plan support de  $S', P'(a', b', c', \mu')$ . Le paragraphe b est une étude de l'intersection de  $P$  et  $P'$  qui permet de mieux comprendre le phénomène géométrique sous-jacent au changement de base.

#### a. Base $\{U, W\}$ du plan support de $S' = S \cup \{M\}$

Soit  $M$  un point 1-extérieur à  $S$ ,

*Conjecture 8.2 :*

*Si la ligne polygonale  $L$  des vecteurs de pivot n'est pas réduite à un point, il existe au plus un vecteur  $W$  de  $L$  tel qu'une antipode  $A$  de  $W$  soit séparante.*

Nous appelons  $W$  le *vecteur principal de pivot* et  $MA$  l'*axe associé*.

*Conjecture 8.3 :*

*Si le vecteur principal de pivot  $W$  existe, soit  $MA$  l'axe associé, l'ensemble  $S' = S \cup \{M\}$  est un morceau de plan reconnu. Les deux vecteurs de l'ensemble  $\{W, U = MA\}$  forment une base de ce plan et les caractéristiques ainsi obtenues sont telles que  $c$  soit le plus petit possible.*

*Les extrémités de  $W$  se trouvent sur un des nouveaux polygones d'appui et les points  $A$  et  $M$  sur l'autre ; nous sommes dans la configuration du CAS2.2.*

Illustration de la conjecture 8.3 avec  $L$  non réduite à un point :

Voici la représentation par lignes de niveaux du morceau reconnu  $S$  du plan  $P(5, 6, 7, -1)$  :

-2	-3	-4	-4	-5	-6	-7	-7	-8	-9	
-1	-2	-3	-4	-4	-5	-6	-6	-7	-8	-9
-1	-1	-2	-3	-3	-4	-5	-6	-6	-7	-8
0	0	-1	-2	-3	-3	-4	-5	-5	-6	-7

**Figure 8.11**

Le point  $M(10, 3, -10)$  est 1-extérieur au morceau reconnu du plan  $P$ ,  $r(M) = -2$  :

4	2	0	5	3	1	-1	4	2	0	-2
5	3	1	-1	4	2	0	5	3	1	-1
A	4	2	0	5	3	1	-1	4	2	0
0	5	3	1	-1	4	2	0	5	3	1

M

**Figure 8.12**

La ligne polygonale des vecteurs de pivot est constituée de deux vecteurs  $W_1$  et  $W_2$ . Seul  $W_1$  possède une antipode  $A$  séparante qui a pour coordonnées  $(0, 1, -1)$ .  $W_1$  est le vecteur principal de pivot et l'ensemble  $S' = S \cup \{M\}$  est un morceau reconnu du plan dont une base est  $\{W_1, MA\}$ . Les caractéristiques de ce plan sont  $a' = 13, b' = 16, c' = 18, \mu' = -2$ .

Conjecture 8.4 :

Si la ligne polygonale des vecteurs de pivot  $L$  est réduite à un point  $K$ , soit  $W$  l'arête sur le CVA telle que le triangle  $[M, W]$  contienne le point  $K$  de  $L$  et telle que  $M$  et tous les points du CVA soient du même côté de la droite passant par  $W$ . Soit  $N$  un sommet de  $W$ , la base  $\{U = MN, W\}$  permet de reconnaître l'ensemble  $S' = S \cup \{M\}$  avec  $c$  le plus petit possible.

$M$  et les extrémités de  $W$  se trouvent sur un des nouveaux polygones d'appui et le point  $K$  sur l'autre ; nous sommes dans la configuration du CAS3.1.

Illustration de la conjecture 8.4 avec  $L$  réduite à un point :

Voici la représentation par lignes de niveaux du morceau reconnu  $S$  du plan  $P(4, 5, 9, 0)$  :

-1	-2	-2	-3	-3						
-1	-1	-2	-2	-2	-3	-3	-4	-4	-5	-5
0	-1	-1	-1	-2	-2	-3	-3	-4	-4	-5
0	0	0	-1	-1	-2	-2	-3	-3	-4	-4

Figure 8.13

Le point  $M(5, 3, -4)$  est 1-extérieur au morceau reconnu du plan  $P$ ,  $r(M) = -1$  :

6	1	5	0	4	-1					
1	5	0	4	K	3	7	2	6	1	5
5	0	4	8	3	7	2	6	1	5	0
0	4	8	3	7	2	6	1	5	N	4

W

Figure 8.14

La ligne polygonale des vecteurs de pivot est réduite au point  $K(4, 2, -2)$ .  $W$  sur PI a pour coordonnées  $(9, 0, -4)$  ;  $S' = S \cup \{M\}$  est un morceau reconnu de  $P'(12, 16, 27, 0)$ .

Conjecture 8.5 :

Si  $L$  n'est pas réduite à un point et s'il n'existe pas de vecteur principal de pivot :

- a. Si la ligne polygonale des antipodes est réduite à un point, il existe un seul point  $K$  de  $L$  et une seule arête  $W$  sur le CVA telle que le triangle  $[M, W]$  contienne le point  $K$ .
- b. Si la ligne polygonale des antipodes n'est pas réduite à un point, il existe un seul vecteur  $W$  sur la ligne polygonale des antipodes qui n'est pas un vecteur de la ligne polygonale des vecteurs de pivot, de plus le triangle  $[M, W]$  contient un unique point  $K$  de  $L$ .

Soit  $N$  un sommet de  $W$ , la base  $\{U = MN, W\}$  permet de reconnaître l'ensemble  $S' = S \cup \{M\}$  avec  $c$  le plus petit possible.

$M$  et les extrémités de  $W$  se trouvent sur un des nouveaux polygones d'appui et le point  $K$  sur l'autre ; nous sommes dans la configuration du CAS3.1.

Exemples :

1. Illustration du point a ;  $L$  n'est pas réduite à un point, aucun des vecteurs présents dans  $L$  ne possède d'antipode séparante et la ligne polygonale des antipodes est réduite à un point. Voici la représentation par lignes de niveaux du morceau reconnu  $S$  du plan  $P(4, 9, 11, -7)$  :

-3	-3	-3	-4	-4					
-2	-2	-3	-3	-3	-4	-4	-4	-5	-5
-1	-1	-2	-2	-2	-3	-3	-4	-4	-4
0	-1	-1	-1	-2	-2	-2	-3	-3	-3

Figure 8.15

Le point  $M(3, 3, -5)$  est 1-extérieur au morceau reconnu du plan  $P$ ,  $r(M) = -8$  :

-6	-2	2	-5	-1	-8				
-4	0	-7	-3	1	-6	-2	2	-5	-1
-2	2	-5	-1	3	-4	0		-3	1
0	-7	-3	1	-6	-2	2	-5	-1	-3

Figure 8.16

Un seul vecteur  $V$  est présent sur la ligne polygonale des vecteurs de pivot, l'antipode  $A(1, 0, -1)$  de  $V$  n'est pas séparante.  $W$  sur PI a pour coordonnées  $(6, 1, -3)$ ,

le point  $K(4, 1, -2)$  est contenu dans le triangle  $[M, W]$ , alors  $S' = S \cup \{M\}$  est un morceau reconnu de  $P'(5, 12, 14, -9)$ .

2. Illustration du point b; L n'est pas réduite à un point, aucun des vecteurs présents dans L ne possède d'antipode séparante et la ligne polygonale des antipodes n'est pas réduite à un point.

Sur la figure 8.17, le point  $M(0, 11, -4)$  est 1-extérieur au morceau reconnu du plan  $P(1, 1, 3, 0)$  représenté par restes,  $r(M) = -1$ . Les vecteurs  $V_1$  et  $V_2$  de la ligne polygonale des vecteurs de pivot L ne possèdent pas d'antipode séparante. Le vecteur  $W$  présent sur la ligne polygonale des antipodes n'apparaît pas sur L, par conséquent,  $S' = S \cup \{M\}$  est un morceau reconnu de  $P'(10, 11, 31, -3)$ . De plus, le point  $K(1, 10, -3)$  est contenu dans le triangle  $[M, W]$ , il sera un point du nouveau PS et  $D, N$  et  $M$  des points du nouveau PI.

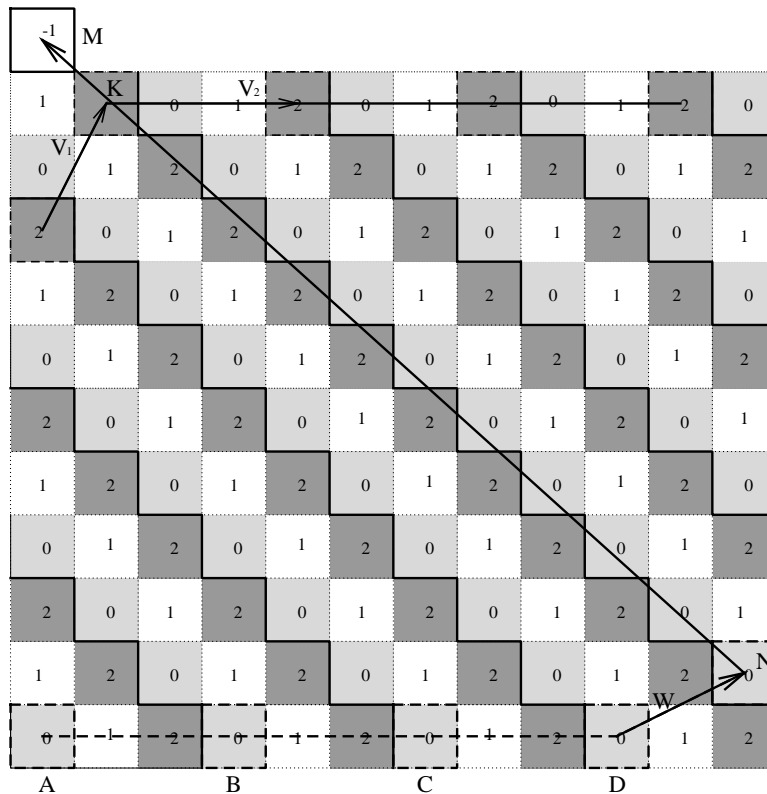


Figure 8.17

## b. Intersection des plans $P$ et $P'$ dans le cas 1-extérieur

Considérons  $\{W, U = NM\}$  la base de  $P'$  obtenue grâce aux théorèmes du paragraphe précédent,  $P'$  étant le plan support de  $S' = S \cup \{M\}$  avec  $S$  un morceau reconnu du plan  $P$  et  $M$  un point 1-extérieur à  $S$ . Dans tous les cas, trois points d'appui communs à  $P$  et  $P'$  ont été déterminés, nous les appellerons  $J, K$  et  $N$ , deux de ces points sont extrémités du vecteur  $W$ . Cette propriété implique que l'intersection des plans  $P$  et  $P'$  possède un zone commune 18-connexe sans trou 6-connexe appelée **couloir commun** délimitée par des voxels formant deux **pointillés frontières** dirigés par le vecteur  $W$  dont voici les équations dans le plan Oxy.

### *Pointillés frontières :*

*supérieurs* :  $(M + t * W)_{Oxy}, t \in \mathbb{Z}$

*inférieurs* :  $(K - U + t * W)_{Oxy}, t \in \mathbb{Z}$  et  $K$  un point sur le polygone d'appui de  $S'$  auquel  $M$  n'appartient pas

Les pointillés frontières sont composés de voxels "superposés" de  $P$  et  $P'$  et seul  $M$  apparaît sur un pointillé, tous les points de  $S$  sont dans la zone commune.

Exemple :

Sur la figure suivante nous voyons la représentation par restes d'un morceau reconnu du plan  $P(5, 6, -9, -4)$  et le point  $M(2, 5, -5)$  1-extérieur qui y est ajouté.  $r(M) = -5$ ,  $M$  est "en dessous" de  $P$ . Avec les théorèmes décrits précédemment nous obtenons un morceau reconnu du plan  $P'(12, 15, -22, -11)$  de base  $\{U = (-5, 4, 0), W = (2, 2, 2)\}$ .

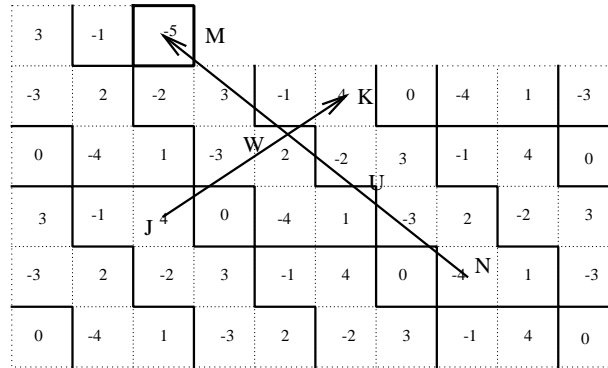
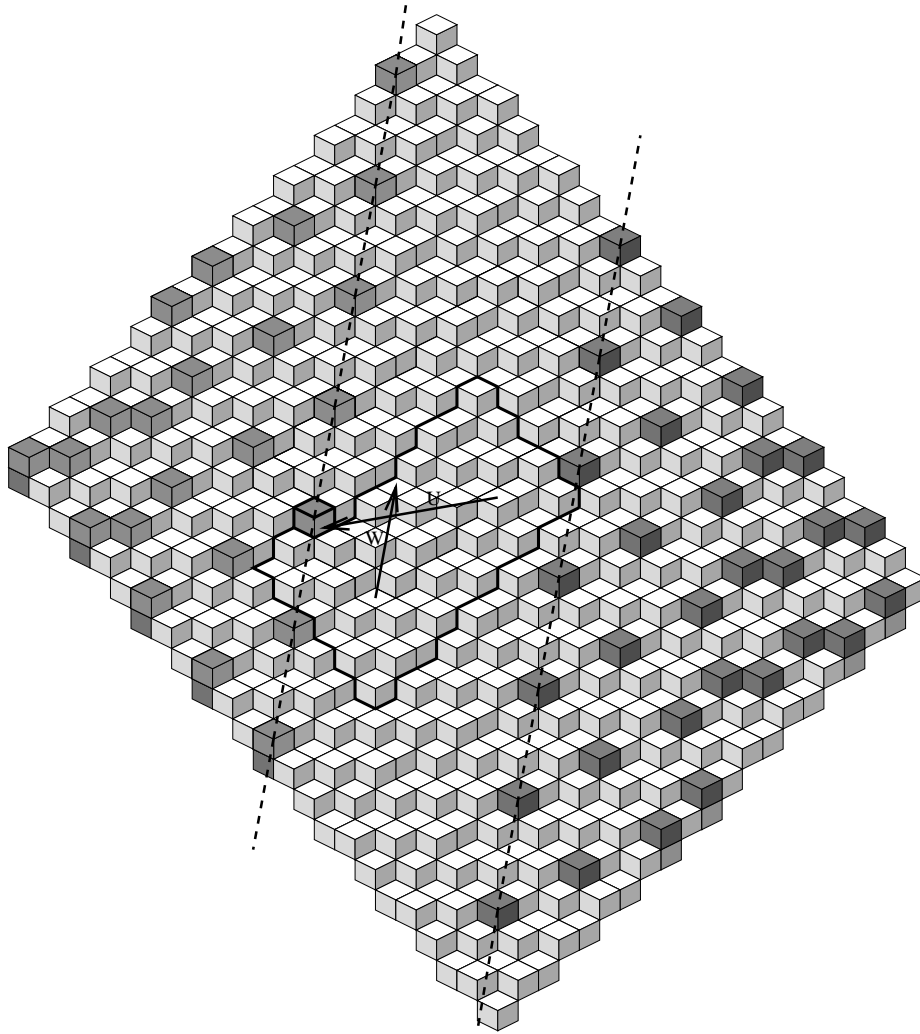


Figure 8.18

Les voxels communs à  $P$  et  $P'$  sont représentés en blanc sur la figure 8.19, les voxels de  $P$  en gris clair et ceux de  $P'$  en gris foncé, de plus, le domaine de reconnaissance est marqué par un trait épais et le point  $M$  est situé sous le voxel de  $P$  marqué en

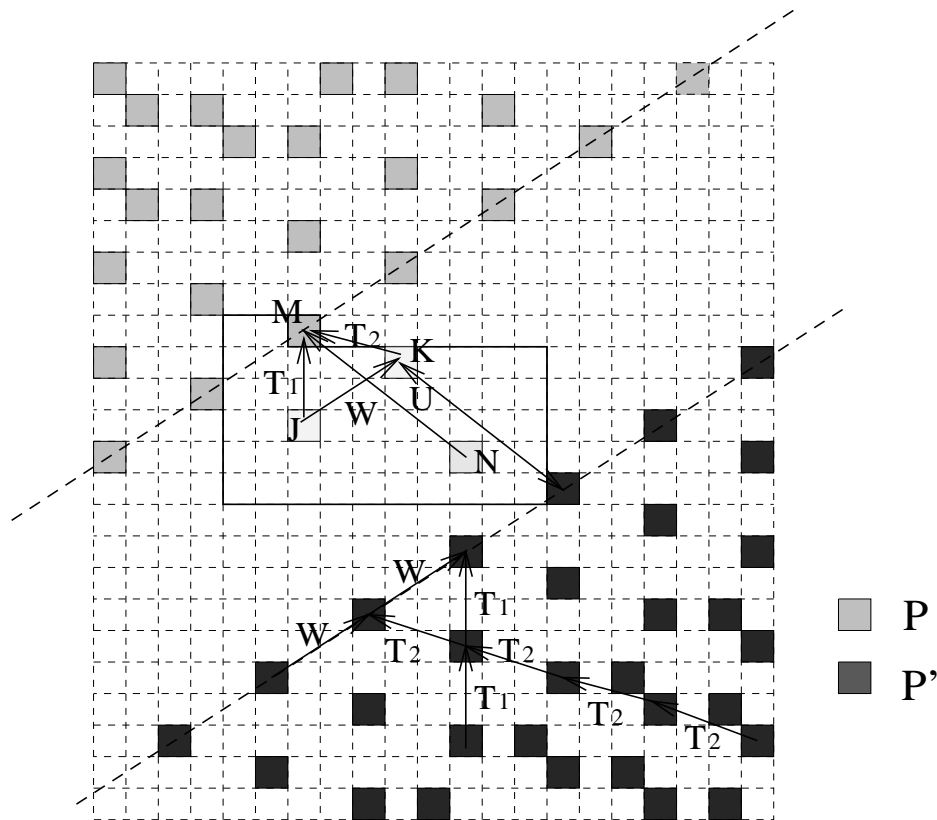
traits épais. Le couloir commun apparaît clairement sur ce dessin et les pointillés frontières supérieures et inférieures dirigés par  $W$  sont marqués.



**Figure 8.19**

L'ensemble constitué des trois points d'appui communs à  $P$  et  $P'$  et le point  $M$  engendre les vecteurs  $U$  et  $W$  mais d'autres vecteurs sont présents tels que  $T_1 = JM$  et  $T_2 = KM$  comme on peut le constater sur la projection dans le plan  $Oxy$  de cet ensemble de voxels sur la figure 8.20.





**Figure 8.20 :** Les pixels en gris clair représentent la projection de deux voxels ; un voxel de  $P$  situé “au dessus” d’un voxel de  $P'$  et les pixels en gris foncé un voxel de  $P'$  situé “au dessus” d’un voxel de  $P$ . Les pixels en blanc sont les projections des voxels communs à  $P$  et  $P'$ .

### 8.3.3 Ajout d’un point fortement extérieur à un morceau reconnu $S$ du plan $P$

Dans le cas où  $M$ , le point ajouté au morceau reconnu  $S$  du plan  $P$ , est  $k$ -extérieur avec  $k > 1$ , la recherche d’une nouvelle base  $\{U, W\}$  reposant sur les théorèmes précédents reste valable, mais il faut ensuite tester la validité des caractéristiques  $a'$ ,  $b'$ ,  $c'$  et  $\mu'$  obtenues par ces constructions. Si les caractéristiques sont reconnues valides, i.e. tous les points  $(x, y, z)$  de  $S$  et  $M$  vérifient

$$\mu' \leq a'x + b'y + c'z < \mu' + c',$$

alors  $S \cup \{M\}$  est un morceau reconnu du plan  $P'(a', b', c', \mu')$ . Mais par contre dans le cas où ces caractéristiques ne conviennent pas, nous ne pouvons pas conclure direc-

tement que  $S \cup \{M\}$  n'est pas un morceau de plan discret, une nouvelle base doit être construite en s'appuyant sur des polygones d'appui enrichis. En effet, dans certains cas très particuliers et assez rares (1% des cas testés), l'ajout d'un point fortement extérieur crée un déséquilibre dans la structure et les polygones d'appui se révèlent insuffisants pour élaborer une nouvelle base satisfaisante. Nous devons reconstruire une autre base en tenant compte de certains points qui doivent être ajoutés aux polygones d'appui et obtenir ainsi d'autres caractéristiques à tester ; cette opération est appelée **reconstruction avec des polygones d'appui enrichis**.

Le cas fortement extérieur est résumé par la conjecture suivante :

*Conjecture 8.6 :*

*Soit  $M$  un point fortement extérieur ajouté à un morceau reconnu  $S$  de plan discret,  $S' = S \cup \{M\}$  est un morceau reconnu de plan discret si et seulement si :*

- les caractéristiques obtenues grâce aux constructions géométriques des conjectures de la section 8.3.2 sont valides,*

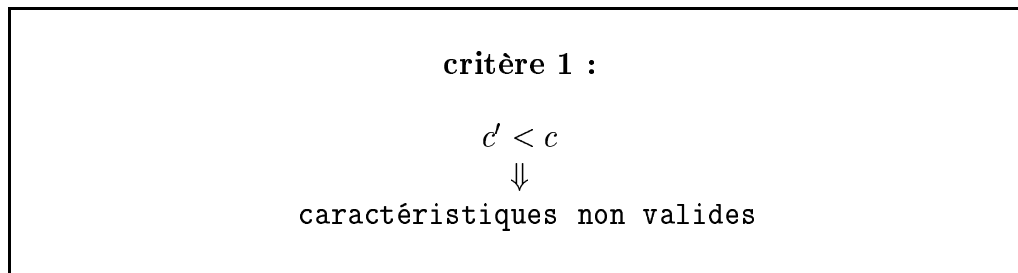
*ou sinon,*

- après reconstruction avec les polygones d'appui enrichis, les caractéristiques déduites de la base élaborée sont valides.*

Trois critères ont été élaborés pour tester la validité des caractéristiques obtenues, ils sont présentés dans le paragraphe *a*. L'opération de reconstruction avec des polygones d'appui enrichis est exposée dans le paragraphe *b*.

#### **a. Critères de validité des caractéristiques**

Deux tests très simples peuvent être effectués pour détecter des caractéristiques non valides, ils sont énoncés ci-dessous sous forme de critères.



La condition énoncée sur la page précédente est claire car à chaque étape, nous

construisons une base qui engendre des caractéristiques telles que  $c$  soit le plus petit possible.

**critère 2 :**

Certains des quatre points obtenus par les constructions basées sur les théorèmes de la section 8.3.2 ne sont pas des points d'appui de l'éventuel morceau reconnu  $S' = S \cup \{M\}$  du plan  $P'(a', b', c', \mu')$  ou bien  $\mu' > 0$

↓

caractéristiques non valides

Les deux conditions du critère 2 impliquent obligatoirement la présence de points dans le domaine déjà reconnu qui n'appartiennent pas au nouveau plan déterminé.

Dans le cas où les caractéristiques obtenues ne sont pas détectées comme étant erronées par les deux critères précédents, un autre test plus complet est nécessaire. En effet, pour comprendre le phénomène géométrique mis en œuvre, il faut étudier l'intersection entre les deux plans  $P$  et  $P'$ , où  $P'$  est candidat pour être le plan support de  $S' = S \cup \{M\}$ .

**Intersection de  $P$  et  $P'$  dans le cas fortement extérieur**

Comme dans la section précédente, l'intersection des plans  $P$  et  $P'$  possède une zone 18-connexe sans trou 6-connexe, les pointillés supérieurs et inférieurs sont aussi présents mais ne délimitent pas cette zone. En effet, d'autres voxels interviennent se situant sur  $\mathbf{k} - 1$  *pointillés intermédiaires* définis ci-dessous dans le plan  $Oxy$ .

Nous considérons le vecteur  $V = JN$ , où  $J$  et  $N$  sont des points de la nouvelle base calculée situés sur des polygones d'appui différents.  $V$  est commun à  $P$  et  $P'$  par construction et est utilisé dans la définition des pointillés intermédiaires.

**Pointillés intermédiaires :**

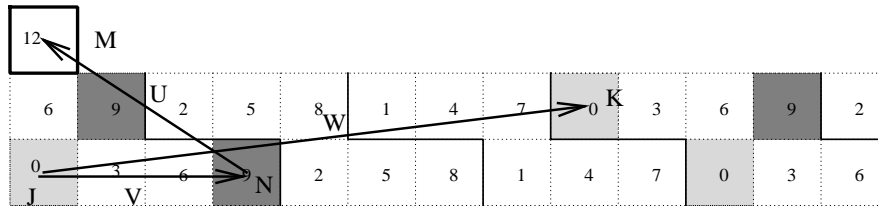
**supérieurs** :  $(M + V * i + t * W)_{Oxy}$   $t \in \mathbb{Z}$  et  $i \in [1, \dots, k - 1]$

**inférieurs** :  $(K - U - V * i + t * W)_{Oxy}$   $t \in \mathbb{Z}$ ,  $i \in [1, \dots, k - 1]$  et  $K$  un point sur le polygone d'appui de  $S'$  auquel  $M$  n'appartient pas

Ces  $2(k - 1)$  pointillés intermédiaires composés de voxels superposés de  $P$  et  $P'$  peuvent apparaître dans le domaine déjà reconnu  $S$  et leur présence doit donc être testée pour vérifier la validité des caractéristiques calculées pour le domaine  $S' = S \cup \{M\}$ .

Exemple où les caractéristiques sont valides :

Nous voyons ci-dessous la représentation par restes d'un morceau reconnu du plan  $P(3, 6, 10, 0)$  et le point  $M(0, 2, 0)$  qui y est ajouté.  $r(M) = 12$ ,  $M$  est 3-extérieur et "au dessus" de  $P$ . Avec les théorèmes décrits précédemment nous obtenons le candidat  $P'(6, 9, 19, 0)$  de base  $\{U = (-3, 2, 0), W = (8, 1, -3)\}$  comme support possible du morceau  $S' = S \cup \{M\}$ .



**Figure 8.21**

La figure 8.22 représente la projection dans le plan  $Oxy$  de  $P$  et de  $P'$ . Le domaine  $S$  est tracé en gras sur le dessin, les pixels blancs sont la projection des voxels de l'intersection de  $P$  et de  $P'$ , les pixels gris clairs sont la projection de deux voxels de  $P$  et  $P'$  situés l'un au dessus de l'autre, le voxel de  $P$  étant situé au dessus de celui de  $P'$  et, symétriquement, les pixels gris foncés représentent la projection d'un voxel de  $P'$  situé au dessus d'un voxel de  $P$ .

Sur cette figure les pointillés frontières sont représentés en traits pleins. Le point  $M$  est 3-extérieur, il y a donc deux pointillés intermédiaires supérieurs et inférieurs dirigés par le vecteur  $W$ . Ils sont en traits pointillés sur le dessin et les projections des voxels appartenant aux pointillés intermédiaires et frontières sont marquées par des points noirs. Le passage d'un pointillé intermédiaire à un autre se fait via le vecteur  $V$ . Dans cet exemple, aucun voxel des pointillés intermédiaires n'est situé dans le domaine déjà reconnu  $S$ , par conséquent, les caractéristiques calculées sont valides et  $S' = S \cup \{M\}$  est un morceau reconnu du plan  $P'(6, 9, 19, 0)$ .

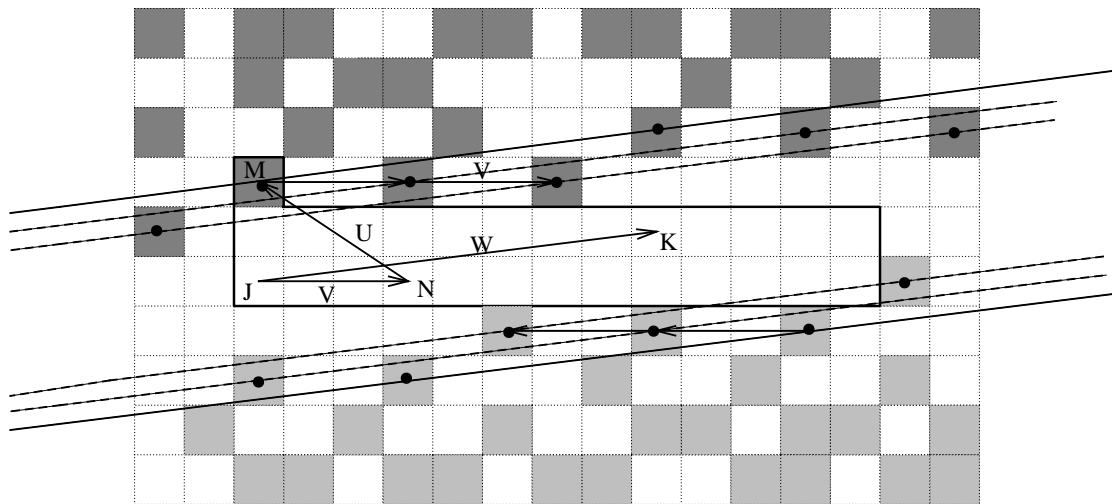
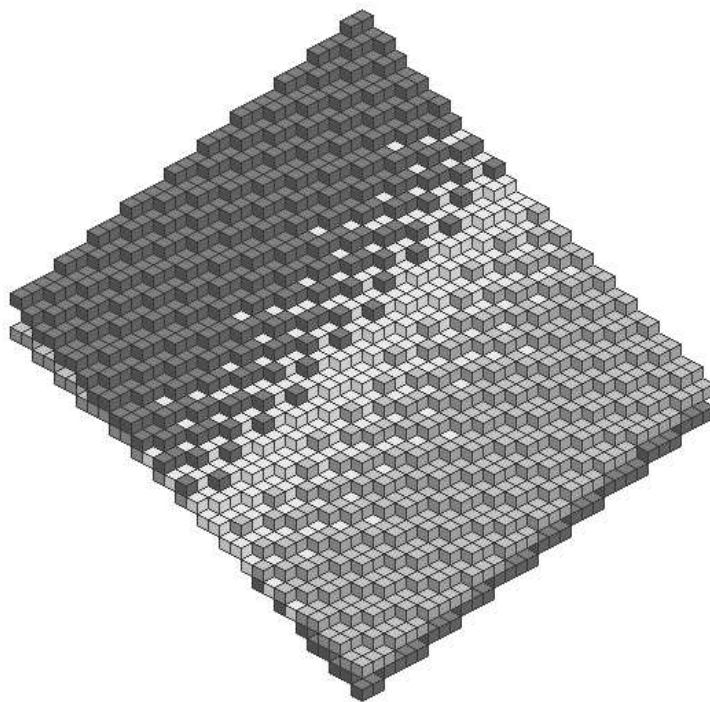


Figure 8.22



**Figure 8.23** : Représentation voxel de  $P(3, 6, 10, 0)$  en gris clair et  $P'(6, 9, 19, 0)$  en gris foncé, les voxels de  $P \cap P'$  sont marqués en gris très clair. Les voxels des pointillés frontières et intermédiaires apparaissent clairement.

Exemple où les caractéristiques ne sont pas valides :

Soit  $S$  un morceau reconnu du plan  $P(4, 8, 13, 0)$ , le point  $M(1, 4, -3)$  a pour reste  $-3$ , il est 3-extérieur à  $S$  et en dessous de  $P$ . Le plan  $P'$  engendré par la nouvelle base  $\{U(1, 4, -3), W(7, 3, -4)\}$  a pour caractéristiques  $a' = 7, b' = 17, c' = 25$  et  $\mu' = 0$ .

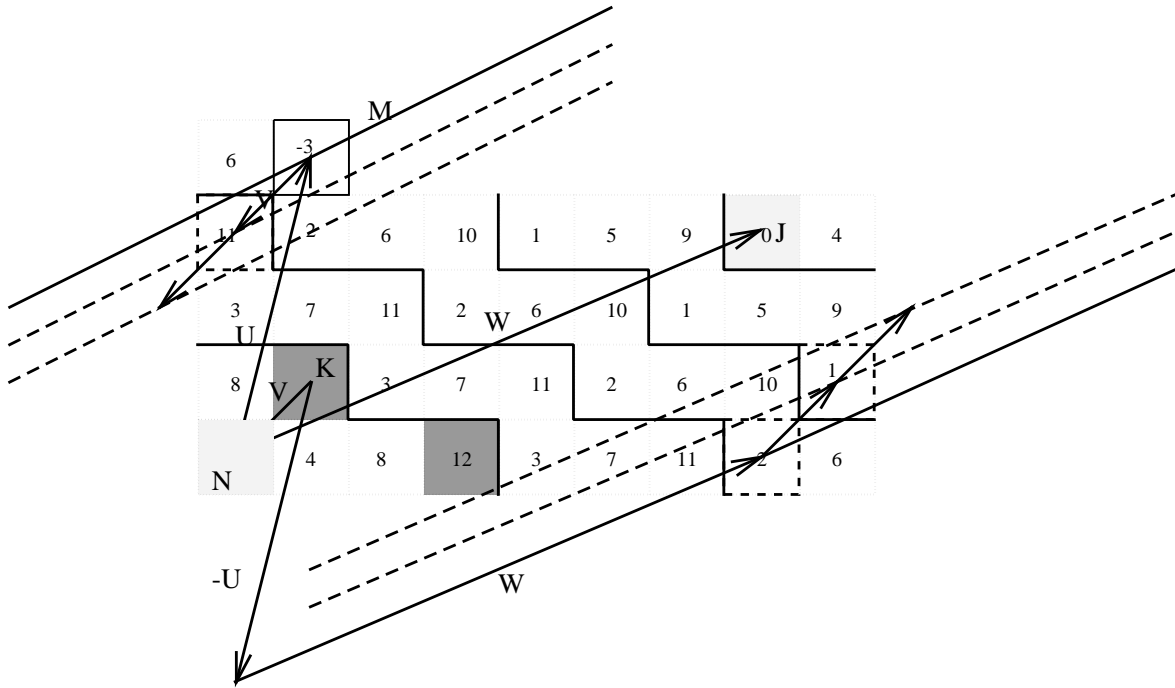
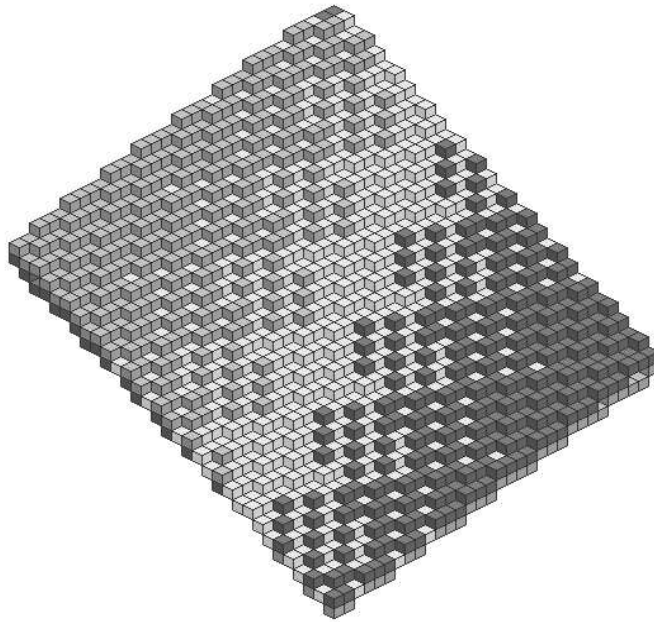


Figure 8.24

Comme on peut le constater sur la figure 8.24, le premier pointillé intermédiaire supérieur intersecte le domaine déjà reconnu au point de coordonnées  $(0, 3)$  dans le plan  $Oxy$ . Par conséquent le point de coordonnées  $(0, 3, -1)$  du domaine  $S$  n'appartient pas à  $P'$ . De plus, deux autres points apparaissent dans le domaine déjà reconnu, l'un d'eux se situe sur un pointillé intermédiaire inférieur et l'autre sur le pointillé frontière inférieur. Les caractéristiques obtenues ne sont pas valides et dans ce cas,  $S \cup \{M\}$  n'est pas un morceau de plan.



**Figure 8.25 :** *Représentation voxel des deux plans  $P$  et  $P'$ .*

Nous pouvons déduire de l'étude précédente un critère de validité des caractéristiques, validité dans le sens où le plan possédant ces caractéristiques sera plan support de  $S'$  :

**critère 3 :**

Intersection vide entre le domaine reconnu  
et les pointillés intermédiaires et frontière inférieure

$\Leftrightarrow$

Caractéristiques valides

Nous ne pouvons malheureusement pas affirmer que si les caractéristiques calculées ne sont pas valides  $S'$  n'est pas un morceau de plan discret. Nous devons reconstruire une autre base en tenant compte de certains points qui doivent être ajoutés aux polygones d'appui et obtenir ainsi d'autres caractéristiques à tester, nous appelons cette méthode *reconstruction avec des polygones d'appui enrichis*.

### b. Reconstruction avec des polygones d'appui enrichis

Les trois critères de validité précédemment décrits doivent être testés dans le cas où le point ajouté est fortement extérieur. Si les caractéristiques ne sont pas valides, nous devons reconstruire une nouvelle base avec des polygones d'appui enrichis. Les points ajoutés aux polygones d'appui sont ceux qui appartiennent au domaine déjà reconnu mais ne sont pas des points du nouveau plan obtenu. Nous représentons ci-dessous deux exemples de reconstruction.

Exemple 1 :

Considérons sur la figure 8.26 la représentation par restes du morceau reconnu  $S$  du plan  $P(18, 21, 23, 0)$ , le point  $M(3, 5, -7)$  2-extérieur est ajouté à  $S$ . Les constructions préconisées par les conjectures de la section 8.3.2 engendrent la nouvelle base  $\{U, W\}$  dont les vecteurs sont visibles sur la figure 8.26.

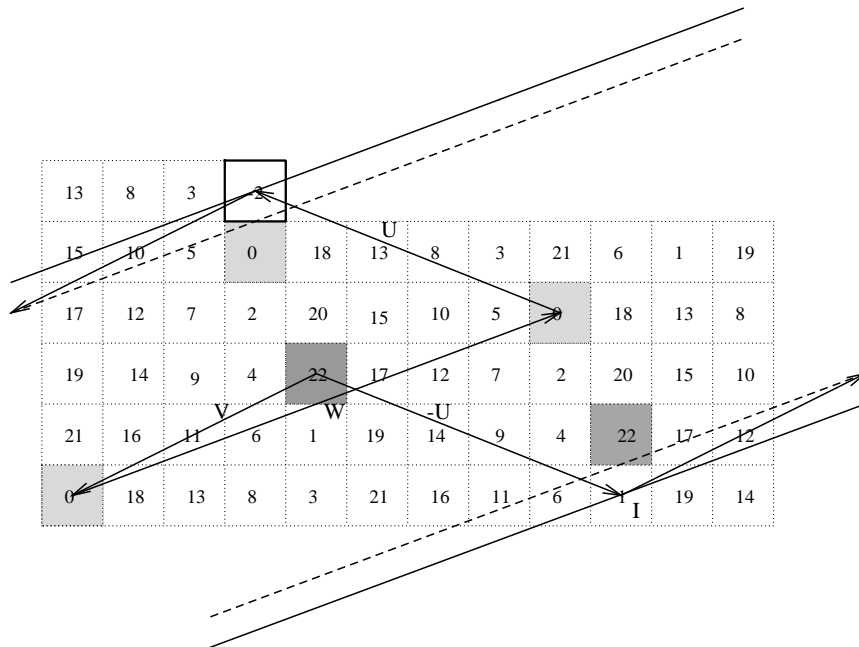


Figure 8.26



Les caractéristiques  $a' = 24$ ,  $b' = 29$ ,  $c' = 31$ ,  $\mu' = 0$  déduites de la base formée par les vecteurs  $U$  et  $W$  ne sont pas valides selon le critère 3. En effet, comme nous pouvons l'observer sur la figure 8.26, le pointillé frontière inférieur possède un point d'intersection nommé  $I$  avec le domaine déjà reconnu  $S$ ,  $I$  a pour coordonnées  $(9, 0, -7)$ . Le point  $I$ , ayant pour reste 1, doit enrichir le polygone d'appui inférieur. Par suite, la base obtenue à partir de ces polygones d'appui est représentée sur la figure suivante.

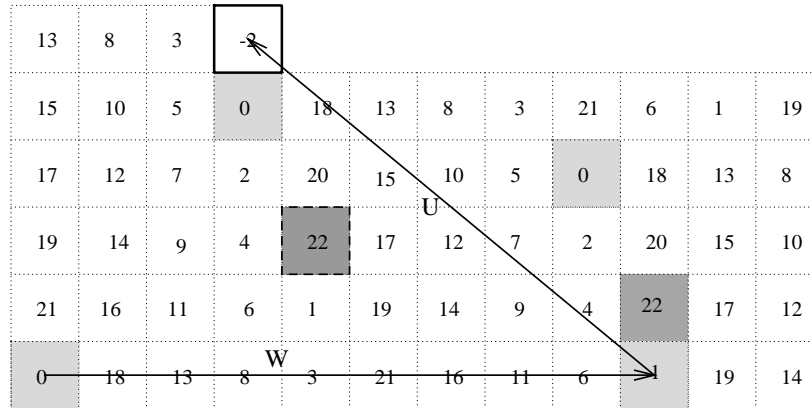


Figure 8.27

Les caractéristiques engendrées par cette nouvelle base sont  $a' = 35$ ,  $b' = 42$ ,  $c' = 45$  et  $\mu' = 0$ .  $S \cup \{M\}$  est un morceau reconnu du plan  $P'(35, 42, 45, 0)$  dont nous voyons la représentation par restes ci-dessous.

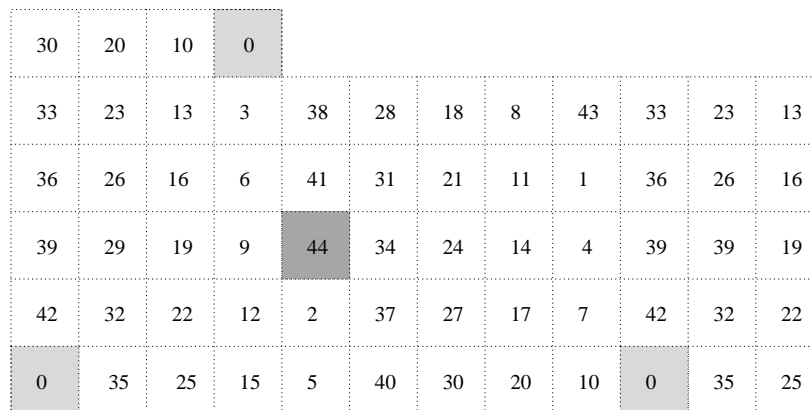


Figure 8.28

Exemple 2 :

Considérons sur la figure 8.29 la représentation par restes du morceau reconnu S du plan  $P(10, 17, 24, 0)$ , le point  $M(2, 6, -4)$  3-extérieur est ajouté à S.

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

Figure 8.29

La base  $\{U, W\}$ , dont les vecteurs sont présents sur la figure 8.30, est construite grâce aux conjectures de la section 8.3.2. Les deux vecteurs U et W engendrent les caractéristiques  $a' = 23$ ,  $b' = 37$ ,  $c' = 54$  mais les extrémités des deux vecteurs ont respectivement pour reste 52 et 0 en fonction de  $a'$ ,  $b'$  et  $c'$ . Par conséquent, ces points ne sont pas des points d'appui d'un plan ayant pour caractéristiques  $a'$ ,  $b'$ ,  $c'$  et le critère 2 détecte la non validité du vecteur normal. Voici la représentation par restes de S en fonction de  $a'$ ,  $b'$  et  $c'$  :

6	29	52										
23	46	15	38	7	30	-1	22	45	14	37	6	
40	9	32	1	24	47	16	39	8	31	0	23	
3	26	49	18	41	10	33	2	25	48	17	40	
20	43	12	35	4	27	50	19	42	11	34	3	
37	6	29	52	21	44	13	36	5	28	51	20	
0	23	46	15	38	7	30	53	22	45	14	37	

Figure 8.30

Les points ayant pour restes  $-1$  et  $53$  sont en dehors des bornes de la structure délimitée par les réseaux d'indices  $0$  et  $52$  auxquels les extrémités de  $U$  et  $W$  appartiennent. Ces deux points, de coordonnées  $(6, 5, -6)$  et  $(7, 0, -2)$ , sont ajoutés selon leur reste aux polygones d'appui inférieur ou supérieur. Par suite, la base obtenue est représentée sur la figure 8.31.

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

Figure 8.31

Les caractéristiques engendrées par cette nouvelle base sont  $a' = 26$ ,  $b' = 42$ ,  $c' = 61$  et  $\mu' = 0$ .  $S \cup \{M\}$  est un morceau reconnu du plan  $P'(26, 42, 61, 0)$  dont nous voyons la représentation par restes sur la figure 8.32.

8	34	60										
27	53	18	44	9	35	0	26	52	17	43	8	
46	11	37	2	28	54	19	45	10	36	1	27	
4	30	56	21	47	12	38	3	29	55	20	46	
23	49	14	40	5	31	57	22	48	13	39	4	
42	7	33	59	24	50	15	41	6	32	58	23	
0	26	52	17	43	8	34	60	25	51	16	42	

Figure 8.32

Pour terminer cette partie théorique, reprenons l'exemple précédemment décrit dans cette section (figure 8.24, page 168),  $S$  est un morceau reconnu du plan  $P(4, 8, 13, 0)$ , le point  $M(1, 4, -3)$  3-extérieur lui est ajouté. Comme nous l'avons constaté, les caractéristiques obtenues grâce aux conjectures de la section 8.3.2 ne sont pas valides ; trois points n'appartenant pas au nouveau plan calculé sont dans  $S$ . La reconstruction d'une nouvelle base avec des polygones d'appui enrichis par ces points échoue, les caractéristiques ne sont pas valides et nous pouvons alors conclure que  $S \cup \{M\}$  n'est pas un morceau de plan discret.

## 8.4 L'algorithme incrémental de reconnaissance de morceaux de plans discrets naïfs

La théorie décrite dans la section précédente permet d'élaborer dans ce paragraphe un algorithme incrémental, très efficace, de reconnaissance de morceaux de plans discrets naïfs.

### 8.4.1 Initialisation des caractéristiques

Soit  $\mathcal{V}$  un ensemble convexe de voxels situés dans le 48-ième d'espace tel que, pour tout voxel de  $\mathcal{V}$  de coordonnées  $(x, y, z)$  de  $\mathbb{Z}^3$ , on ait  $x \geq |z|$ ,  $y \geq |z|$  et  $z \leq 0$ .

---

#### Initialisation des caractéristiques

Utiliser l'algorithme de reconnaissance des droites 2D (cf. chapitre 3) pour la première ligne en  $y$  de  $\mathcal{V}$  ;  
l'algorithme fournit les caractéristiques  $a_0, b_0, r_0$ .

Initialiser les caractéristiques :

$$a = a_0, b = a_0, c = b_0, \mu = r_0$$

Initialiser les polygones d'appui, PI et PS.

---

Bien sûr, si l'algorithme de reconnaissance des droites 2D appliqué sur la première ligne en  $y$  de  $\mathcal{V}$  échoue, la reconnaissance de  $\mathcal{V}$  s'arrête,  $\mathcal{V}$  n'est pas un morceau de plan discret.

### 8.4.2 L'algorithme simplifié

Un algorithme simplifié, de complexité minimale, est proposé dans ce paragraphe, il ne considère pas les cas rares décrits dans la section 8.3.3 et ne reconstruit pas de base avec des polygones enrichis quand cela est nécessaire. Si un point fortement extérieur est ajouté à un morceau reconnu de plan discret et si les caractéristiques obtenues grâce aux théorèmes de la section 8.3.2 ne sont pas valides, la reconnaissance s'arrête.

Dans cet algorithme, deux fonctions sont utilisées :

**CAS3\_1(CVA,L,M)** où,

CVA est le convexe des antipodes du morceau reconnu de plan discret courant,  
 $L$  est la ligne polygonale des vecteurs de pivot,  
 $M$  est le point extérieur ajouté.

Si  $L$  est réduite à un point (marque (i) de l'algorithme), CAS3\_1 calcule la nouvelle base formée des vecteurs  $U$  et  $W$  en tenant compte des constructions géométriques données dans la conjecture 8.4. Sinon  $L$  n'est pas réduite à un point (marque (iii) de l'algorithme) , dans ce cas, la conjecture 8.5 est utilisé pour obtenir les vecteurs  $U$  et  $W$ .

**CAS2\_2(CVA,L,M)** où,

CVA est le convexe des antipodes du morceau reconnu de plan discret courant,  
 $L$  est la ligne polygonale des vecteurs de pivot,  
 $M$  est le point extérieur ajouté.

Les vecteurs  $U$  et  $W$  sont calculés dans CAS2\_2 en utilisant les constructions géométriques décrites dans les conjectures 8.2 et 8.3 (marque (ii) de l'algorithme).

---

## Algorithme incrémental simplifié de reconnaissance de morceaux de plans discrets

*Initialiser les caractéristiques;*

**Tant que**  $\mathcal{V}$  n'a pas été entièrement parcouru et que les caractéristiques sont valides **répéter**

reste =  $r(M)$  ;

**Si** reste  $\geq \mu$  et reste  $< \mu + c$  **alors**

**Si** reste =  $\mu$  **alors** ajouter  $M$  à  $PI$ ; **Fsi**

**Si** reste =  $\mu + c - 1$  **alors** ajouter  $M$  à  $PS$ ; **Fsi**

**Sinon**

**Si** reste  $< \mu$  **alors**

    CVP =  $PS$ ; CVA =  $PI$  ;

**Sinon** // reste  $\geq \mu + c$  //

    CVP =  $PI$ ; CVA =  $PS$  ;

**Fsi**

*Trouver la ligne polygonale des vecteurs de pivot  $L$  sur CVP;*

**Si**  $L$  est réduite à un point **alors**

$(U,W) = CAS3\_1(CVA,L,M)$  ; (i)

**Sinon**

**Si** il existe une antipode séparante sur CVA pour un vecteur de  $L$  **alors**

$(U,W) = CAS2\_2(CVA,L,M)$  ; (ii)

**Sinon**

$(U,W) = CAS3\_1(CVA,L,M)$  ; (iii)

**Fsi**

**Fsi**

*Calculer les nouvelles caractéristiques en fonction des vecteurs obtenus  $(U,W)$ ;*

**Si**  $M$  est fortement extérieur **alors**

*tester la validité des caractéristiques;* (\*\*)

**Fsi**

**Si** les caractéristiques sont valides ou si  $M$  est 1-extérieur **alors**

*mettre à jour les polygones d'appui  $PI$  et  $PS$ ;*

**Fsi**

**Fsi**

$M$  = point suivant de  $\mathcal{V}$  ;

**Fin répéter**

---

Les *nouvelles caractéristiques*  $a$ ,  $b$  et  $c$  sont calculées à partir des vecteurs  $U$  et  $W$  comme indiqué dans le théorème 6.8. La nouvelle caractéristique  $\mu$  dépend de la position de  $M$  par rapport au morceau de plan reconnu courant :

- si  $M$  est situé en dessous du plan courant,  $M$  devient un point d'appui inférieur du nouveau plan support et on a  $\mu = ax_M + by_M + cz_M$ ,
- si  $M$  est situé au dessus du plan courant,  $M$  devient un point d'appui supérieur du nouveau plan support et on a  $\mu = ax_M + by_M + cz_M - c + 1$ .

La *validité des caractéristiques* est testée lorsque  $M$  est fortement extérieur en utilisant les 3 critères donnés dans la section 8.3.3.

La *mise à jour des polygones d'appui* en fonction des nouvelles caractéristiques se déroule en deux temps :

- il faut d'abord trouver les points d'appui du nouveau plan support qui appartiennent à  $S'$ . Les extrémités des vecteurs  $U$  et  $W$  sont des points d'appui, les autres s'obtiennent par combinaison linéaire des vecteurs  $U$  et  $W$  à partir de ces quatre points en se restreignant au domaine  $S'$ .
- ensuite, il suffit de construire l'enveloppe convexe des projections des points d'appui supérieurs et inférieurs dans le plan  $Oxy$  pour obtenir les nouveaux polygones d'appui.

Remarque : Les conditions nécessaires (cf. 6.3) que doit vérifier un plan discret peuvent être testées pour une meilleure efficacité avant de mettre en œuvre l'algorithme de reconnaissance. Ce pré-traitement est de complexité  $O(n)$  avec  $n$  le nombre de voxels de  $\mathcal{V}$ .

### Complexité de l'algorithme simplifié

La complexité de cet algorithme n'est pas facile à évaluer, en effet, nous ne connaissons pas le nombre d'étapes où les caractéristiques vont être modifiées au cours de la reconnaissance d'un morceau de plan discret. Un changement peut être imaginé dans le pire des cas sur chaque fin ou début de palier, soit  $df$  le nombre de voxel de fin ou de début de palier de l'ensemble de voxels  $\mathcal{V}$  à reconnaître.

Regardons en détail la complexité d'une étape qui consiste à ajouter un point  $M$  à un morceau reconnu  $S$  du plan discret  $P(a, b, c, \mu)$  dont nous connaissons les polygones d'appui  $PI$  et  $PS$  possédant respectivement  $k_I$  et  $k_S$  points.

Après le calcul élémentaire de  $r(M)$ , plusieurs directions sont possibles :

1. Si  $\mu + 1 \leq r(M) \leq \mu + c - 2$ ,  $M$  est un point du plan courant et aucun traitement n'est effectué.
2. Si  $r(M) = \mu$  ou  $r(M) = \mu + c - 1$ ,  $M$  est un point d'appui du plan courant  $P$  et doit être ajouté au polygone d'appui concerné. Cette opération consiste à parcourir le polygone en faisant des tests élémentaires et une adjonction, elle est donc en  $O(k)$  avec  $k = \max\{k_I, k_S\}$ .
3. Si  $r(M) < \mu$  ou  $r(M) > \mu + c - 1$  alors,
  - le calcul des nouvelles caractéristiques du plan support  $P'(a', b', c', \mu')$  de  $S'$  demande un certain nombre de parcours des polygones d'appui accompagnés d'opérations élémentaires et a par conséquent un coût en  $O(k)$ .
  - la validité des caractéristiques est testée, si cela est nécessaire, tout d'abord avec les critères 1 et 2 qui utilisent des tests élémentaires puis avec le critère 3 qui demande quelques calculs peu coûteux pour déterminer les points d'intersection entiers entre le domaine et les pointillés frontières et intermédiaires.
  - la construction des polygones d'appui en fonction des nouvelles caractéristiques obtenues se déroule en deux parties; trouver les nouveaux points d'appui se fait par des calculs simples puis il faut calculer les enveloppes convexes de la projection des points dans le plan  $Oxy$ . Le coût de la mise à jour des polygones d'appui est donc en  $O(t' \log t')$  avec  $t' = \max(t'_I, t'_S)$  où  $t'_I$  et  $t'_S$  sont respectivement les points d'appui inférieurs et supérieurs de  $P'$  contenus dans  $S'$ .

Par conséquent, le coût de l'ajout d'un voxel à un morceau reconnu de plan discret par l'algorithme de reconnaissance est inférieur ou égal à  $O(\max(k, t' \log t'))$  avec  $k$  et  $t'$  définis comme précédemment. Par conséquent, la complexité de l'algorithme simplifié de reconnaissance peut être estimée en majorant le coût d'une étape et en considérant  $df$  comme nombre maximal d'étape mais cela est peu explicite et ne traduit pas la simplicité et l'utilité de la propriété incrémentale de l'algorithme.

### 8.4.3 L'algorithme complet

Pour que l'algorithme détermine dans tous les cas si un ensemble de voxels convexe est un morceau de plan discret, il faut modifier la partie marquée (\*\*) de l'algorithme simplifié :



**Si M est fortement extérieur alors**  
    *tester la validité des caractéristiques ;*                   (\*\*)  
**Fsi**

---

devient,

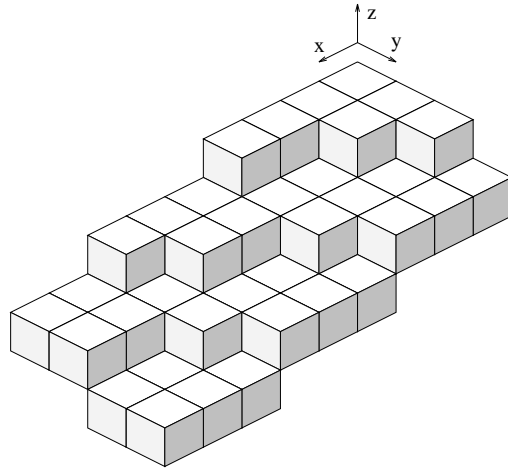
---

**Si M est fortement extérieur alors**  
    *tester la validité des caractéristiques ;*  
    **Si les caractéristiques ne sont pas valides alors**  
        *chercher une nouvelle base en utilisant des polygones d'appui enrichis ;*  
        *tester la validité des caractéristiques ;*  
    **Fsi**  
**Fsi**

---

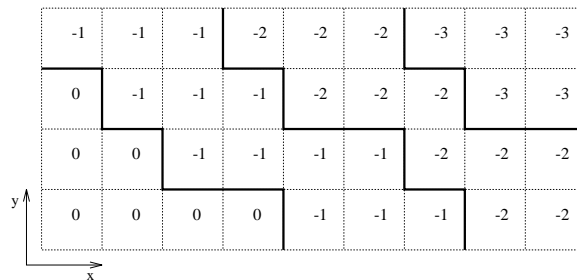
Ce nouveau calcul de caractéristiques augmente la complexité de l'algorithme, en effet, les points ajoutés aux polygones d'appui s'obtiennent pour l'instant (nous espérons trouver rapidement une autre méthode de détection) par balayage du domaine parcouru quand la non validité des caractéristiques a été détectée par les critères 1 ou 2. Puis, l'insertion de ces points pour former les polygones d'appui enrichis a le coût de l'insertion de points dans une enveloppe convexe 2D.

### 8.4.4 Un exemple de déroulement de l'algorithme de reconnaissance de morceaux de plans discrets



**Figure 8.33 :** Ensemble  $\mathcal{V}$  de voxels à reconnaître.

Considérons l'ensemble  $\mathcal{V}$  de voxels codé par ses lignes de niveaux sur la figure 8.33. Rappelons que le pixel  $(x, y)$  est affecté de la valeur  $z$  du point  $M = (x, y, z)$  qu'il représente.



**Figure 8.34**

L'initialisation des caractéristiques se fait en reconnaissant la droite discrète sur la ligne  $y = 0$  avec l'algorithme  $2D$  (cf. Chapitre 3). Nous obtenons la droite de pente  $2/7$  et de borne inférieure 0.

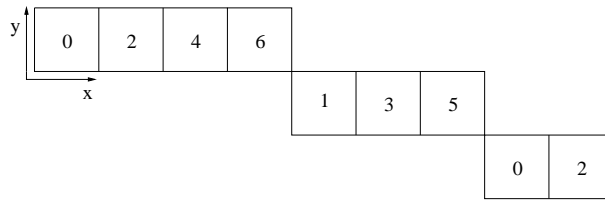


Figure 8.35

Les caractéristiques initiales sont alors

$$a = 2, b = 2, c = 7, \mu = 0,$$

car dans ce 48-ième d'espace, on doit avoir  $0 \leq a \leq b \leq c$ . Les voxels de la ligne  $y = 0$  de  $\mathcal{V}$  appartiennent au plan  $P_0(2, 2, 7, 0)$ .

Les polygones d'appui sont alors initialisés :

PS est réduit au point  $(3, 0, 0)$

PI a pour sommets  $(0, 0, 0)$  et  $(7, 0, -2)$ .

Puis nous testons chaque voxel  $M(x, y, z)$  de la ligne  $y = 1$  et déterminons la valeur  $r(M) = ax + by + cz$ .

Le voxel de coordonnées  $(0, 1, 0)$  a comme reste 2 pour  $P_0$ , il appartient donc à ce plan (car  $\mu = 0 \leq 2 < 7 = \mu + c$ ) de même que le point voisin  $(1, 1, 0)$  qui a comme reste 4. Mais au point  $M_1$  de coordonnées  $(2, 1, -1)$ , on a  $r(M_1) = -1 (= \mu - 1)$  donc  $M_1 \notin P_0$ ,  $M_1$  est 1-extérieur au morceau reconnu de plan support  $P_0$  composé des voxels déjà parcourus.  $M_1$  est situé en dessous de  $P_0$ , par suite, CVP=PS et CVA=PI. En projetant le morceau déjà parcouru sur le plan  $0xy$ , nous obtenons le codage des restes suivant :

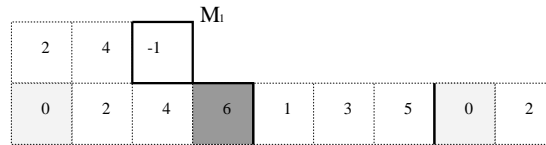
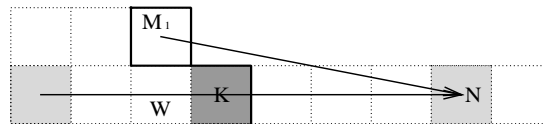


Figure 8.36

PS étant réduit à un seul point, le vecteur  $W(7, 0, -2)$  présent sur PI et le vecteur  $M_1N$  avec  $N(7, 0, -2)$  forment une base d'un morceau reconnu de plan comprenant tous les voxels parcourus y compris  $M_1$  (cf. conjecture 8.4).

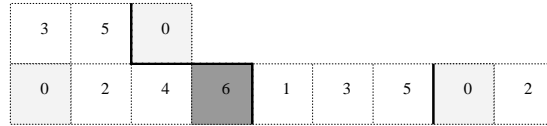


**Figure 8.37**

Nous obtenons les caractéristiques

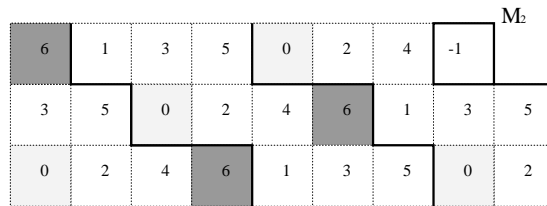
$$a = 2, b = 3, c = 7.$$

De plus,  $r(M_1) = 0$  et comme  $M_1$  est un nouveau point d'appui inférieur on a  $\mu = 0$ . L'ensemble des voxels parcourus est donc un morceau reconnu du plan  $P_1(2, 3, 7, 0)$  et nous sommes dans le CAS3.1 avec trois points d'appui inférieurs et un point d'appui supérieur :



**Figure 8.38**

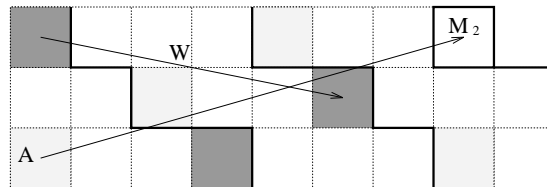
Les points suivants de  $\mathcal{V}$  appartiennent à  $P_1$  jusqu'au voxel  $M_2$  de coordonnées  $(7, 2, -3)$ , en effet  $r(M_2) = -1$ ,  $M_2$  est 1-extérieur à  $P_1$ .



**Figure 8.39**

PI a pour sommets  $(0, 0, 0)$ ,  $(7, 0, -2)$ ,  $(4, 2, -2)$ ,  $(2, 1, -1)$   
 PS a pour sommets  $(3, 0, 0)$ ,  $(5, 1, -1)$ ,  $(0, 2, 0)$ .

$M_2$  est situé en dessous de  $P_1$ , nous avons  $CVP=PS$  et  $CVA=PI$ . La ligne polygonale des vecteurs de pivot est composée du vecteur  $W$  ayant pour sommets les points  $(0, 2, 0)$  et  $(5, 1, -1)$  qui appartiennent à PS. Le point  $A$  sur PI de coordonnées  $(0, 0, 0)$  est une antipode du vecteur  $W$  par rapport à  $M_2$ . De plus, les deux sommets de  $W$  sont de part et d'autre de la droite  $AM_2$ ,  $A$  est séparante. La nouvelle base est alors  $\{W, AM_2\}$  (cf. conjecture 8.2 et 8.3) que nous pouvons observer sur la figure 8.40.



**Figure 8.40**

Les caractéristiques sont :

$$a = 5, b = 8, c = 17.$$

Comme  $r(M_2) = 0$ , on a  $\mu = 0$ . L'ensemble des voxels déjà parcourus est un morceau reconnu du plan  $P_2(5, 8, 17, 0)$ , nous sommes dans le CAS2.2 avec deux points d'appui supérieurs et deux points d'appui inférieurs :

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

**Figure 8.41**

La reconnaissance se poursuit par l'ajout des trois voxels suivants qui appartiennent à  $P_2$  jusqu'au voxel  $M_3$  de coordonnées  $(2, 3, -1)$ ,  $r(M_3) = 17$ ,  $M_3$  est 1-extérieur à  $P_2$  et se situe au dessus de ce plan, par conséquent CVP=PI et CVA=PS.

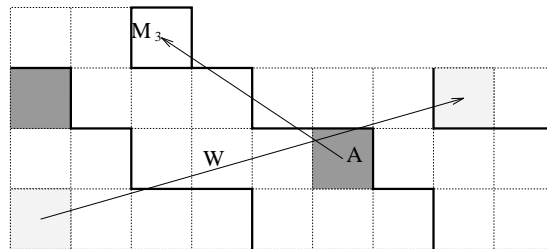
7	12	17						
16	4	9	14	2	7	12	0	5
8	13	1	6	11	16	4	9	14
0	5	10	15	3	8	13	1	6

**Figure 8.42**

PI a pour sommets les points  $(0, 0, 0), (7, 2, -3)$

PS a pour sommets les points  $(5, 1, -1), (0, 2, 0)$ .

Le vecteur de pivot  $W$  est sur PI et a pour coordonnées  $(7, 2, -3)$ , son antipode par rapport à  $M_3$  sur PS est  $A(5, 1, -1)$ , elle est séparante et la nouvelle base sera alors  $\{W, AM_3\}$  que nous observons sur la figure 8.43.



**Figure 8.43**

Les caractéristiques sont

$$a = 6, b = 9, c = 20.$$

De plus,  $r(M_3) = 19$ , or  $M_3$  est un nouveau point d'appui supérieur donc  $r(M_3) = \mu + c - 1$  et par suite  $\mu = 19 + 1 - 20 = 0$ . L'ensemble des voxels parcourus est un morceau reconnu du plan  $P_3(6, 9, 20, 0)$ , nous sommes dans le CAS2.2 :

7	13	19						
18	4	10	16	2	8	14	0	6
9	15	1	7	13	19	5	11	17
0	6	12	18	4	10	16	2	8

**Figure 8.44**

Les derniers points de  $\mathcal{V}$  appartiennent à  $P_3$ , la reconnaissance s'arrête.  $\mathcal{V}$  est un morceau reconnu du plan de caractéristiques  $a = 6, b = 9, c = 20, \mu = 0$ .

7	13	19	5	11	17	3	9	15
18	4	10	16	2	8	14	0	6
9	15	1	7	15	19	5	11	17
0	6	12	18	4	10	16	2	8

**Figure 8.45** : Représentation par restes du morceau reconnu du plan  $P_3(6, 9, 20, 0)$ .

## 8.5 Conclusion

L'algorithme incrémental de reconnaissance présenté dans ce chapitre permet de retrouver les caractéristiques d'un morceau convexe de plan discret en utilisant uniquement les coordonnées entières des voxels et les enveloppes convexes 2D des projections des points d'appui. Les conjectures, fondements de l'algorithme établi, devraient être démontrées grâce aux résultats sur les enveloppes convexes 3D de G.

Yaacoub et J.P. Reveillès dont les premières propriétés sont données dans [53].

Il est clair que la discrétisation d'une surface au voisinage d'un point *régulier* donne en général un morceau de plan discret. On peut donc espérer *facettiser* des surfaces discrètes en appliquant cet algorithme localement. Le point délicat de ce travail réside dans la stratégie de parcours puisque la facettisation n'est évidemment pas canonique. Un premier essai est proposé dans le chapitre suivant.

Comme il est assez facile de déterminer toutes les intersections des plans naïfs passant par l'origine avec le cube de ses  $(2k + 1)^3 - 1$  voisins  $k \geq 1$ , (i.e. 26, 124 ...) nous pouvons également espérer de ce travail une notion de *k-régularité* des surfaces discrètes qui devrait redonner, par passage à la limite, la notion de variété différentiable du continu. Les prémices de cette étude sont données dans le chapitre suivant concernant les applications de l'algorithme de reconnaissance de morceaux de plans discrets.





## Chapitre 9

# Applications de l'algorithme de reconnaissance de morceaux de plans discrets

## 9.1 Introduction

Une étude précise des voisinages discrets minimaux autour d'un point appartenant à un plan discret est la première étape pour définir une notion de différentiabilité discrète. L'algorithme de reconnaissance de morceaux de plans discrets nous a permis de caractériser les différents voisinages possibles qui sont appelés *tricubes*. Nous proposons, dans la première section de ce chapitre, un dénombrement précis des différentes configurations de 8 voxels voisins d'un point appartenant à un plan discret, puis une propriété sur le nombre de tricubes différents dans un plan discret naïf donné et enfin une application sur la détection des points réguliers d'une image.

Ph. Borianne souligne dans [8] qu'en imagerie médicale 3D, la polyédrisation est d'une grande utilité, cette opération consiste à représenter tout objet discret constitué de voxels par un ou plusieurs polyèdres de  $\mathbb{R}^3$ . La polyédrisation est employée pour améliorer le réalisme des images discrètes en "lissant" les volumes obtenus [45, 60] ou encore pour simplifier la représentation des volumes discrets en diminuant les données nécessaires à la description des volumes et ainsi permettre des manipulations en temps réel [7]. Dans [8], Ph. Borianne et J. Françon proposent une première technique de polyédrisation réversible qui utilise la méthode d'approximation des moindres carrés et, par conséquent, qui n'est pas entièrement discrète. La deuxième partie de ce chapitre est consacrée à un premier essai de facettisation d'objets voxels symétriques en appliquant l'algorithme incrémental de reconnaissance de morceaux de plans discrets présenté dans le chapitre précédent. Une stratégie de parcours et des exemples sont donnés dans le cas d'objets voxels dont les symétries sont connues.

## 9.2 Les tricubes des plans discrets naïfs

### 9.2.1 Définition et dénombrement des tricubes

Dans le cadre de l'étude des plans discrets, nous nous sommes intéressés au dénombrement et à la caractérisation des différentes configurations des plans discrets naïfs autour de l'origine. Dans un plan discret naïf  $P$ , un point  $M$  possède 8 voisins appartenant à ce même plan, l'ensemble de ces 9 voxels est appelé *tricube de  $M$  dans  $P$* .

Nous pouvons observer sur les pages suivantes tous les tricubes possibles pour les plans discrets appartenant au 48ième d'espace tel que le vecteur normal  $(a, b, c)$  de ces plans vérifie  $0 \leq a \leq b \leq c$ .

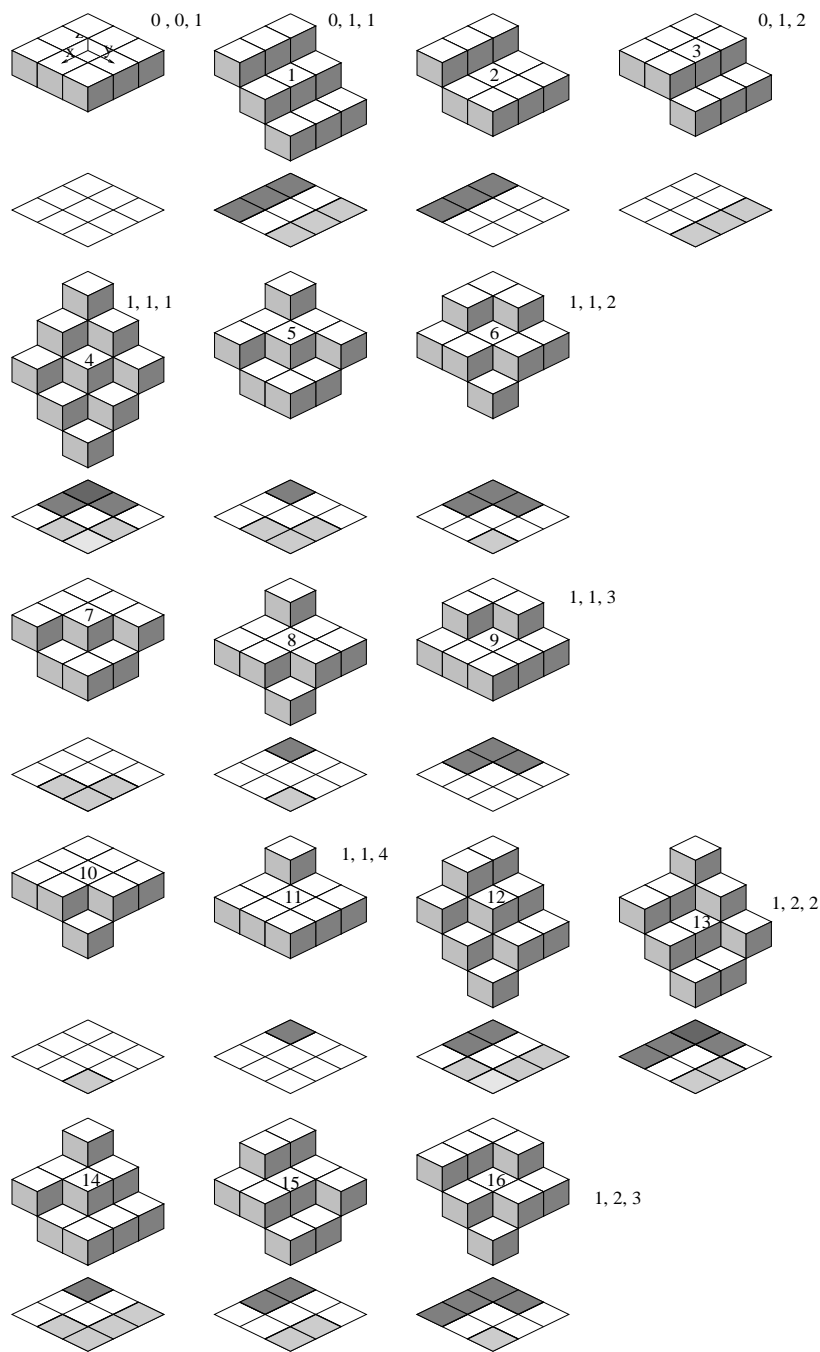
40 configurations différentes sont possibles dans ce 48ième d'espace (leurs représentations sur les pages suivantes sont numérotées sur le point origine) et les caractéristiques minimales associées à ces structures ont été calculées grâce à l'algorithme de reconnaissance de morceaux de plans discrets (cf. chapitre 8) et figurent à côté de ces configurations. 23 vecteurs normaux de base ont été déterminés.

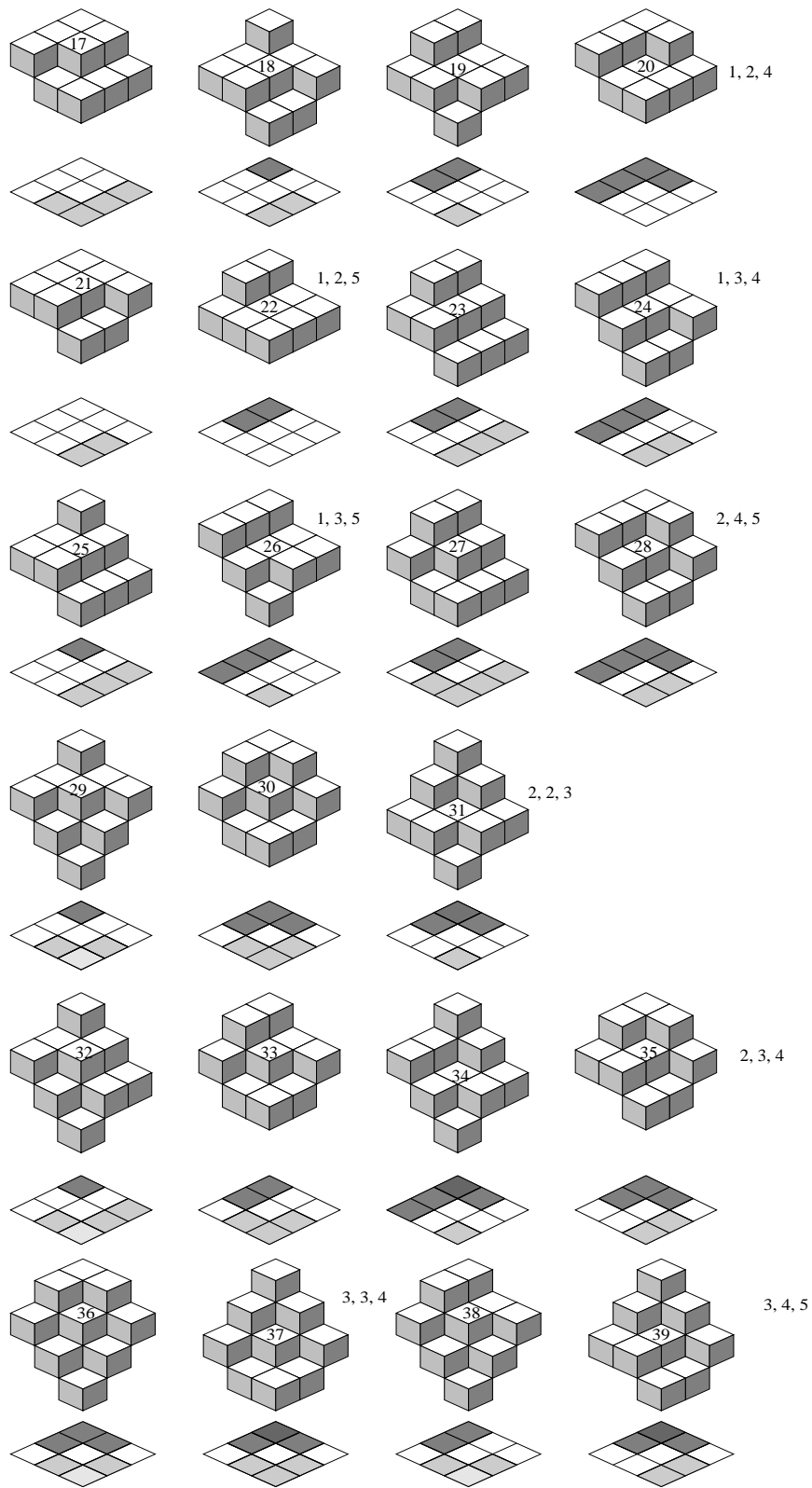
Les tricubes de ce 48ième d'espace se projettent bijectivement dans le plan  $Oxy$  et certains d'entre eux sont communs à d'autres 48ièmes d'espaces ayant le même plan de projection bijective, d'autres encore se projettent bijectivement sur un autre plan de coordonnées. Ces remarques sont importantes car certains tricubes sont communs à plusieurs 48ièmes d'espace voisins.

A partir des 23 vecteurs normaux de base le décompte du nombre de configurations différentes se projetant dans le plan  $Oxy$  de manière bijective figure dans le tableau de la page suivante. Le total est calculé sur la dernière ligne du tableau. 249 tricubes se projettent bijectivement dans le plan  $Oxy$ . Mais parmi eux, les 4 tricubes engendrés à partir du vecteur de caractéristiques 1,1,1 se projettent bijectivement dans les trois plans de coordonnées, de plus, les tricubes obtenus à partir des vecteurs de caractéristiques 0,1,1 et 1,2,2 se projettent bijectivement dans deux plans de coordonnées. Par suite, dans tout l'espace  $\mathbb{Z}^3$  nous trouvons  $3 * 249 - 8(\text{configurations redondantes issues du vecteur } 1,1,1) - 6(\text{configurations redondantes issues du vecteur } 0,1,1) - 24(\text{configurations redondantes issues du vecteur } 1,2,2) = 709$  tricubes différents possibles.

Vecteur normal $ a ,  b ,  c $	Nombre de configurations dans un 48ième d'espace	Nombre de configurations ayant une projection bijective dans le plan $Oxy$
0, 0, 1	1	1
0, 1, 1	1	4
0, 1, 2	2	8
1, 1, 1	1	4
1, 1, 2	2	8
1, 1, 3	3	12
1, 1, 4	2	8
1, 2, 2	2	16
1, 2, 3	3	24
1, 2, 4	4	32
1, 2, 5	2	16
1, 3, 4	2	16
1, 3, 5	2	16
2, 2, 3	3	12
2, 3, 4	4	32
2, 4, 5	2	16
3, 3, 4	2	8
3, 4, 5	2	16
23	40	249

Tricubes des plan discrets  $P(a, b, c, \mu)$  appartenant au 48ième d'espace tel que  $0 \leq a \leq b \leq c$  :





Une propriété intéressante a été dégagée en observant les tricubes appartenant à un plan discret donné :

*Un plan discret possède au plus 9 tricubes distincts.*

Par exemple, le plan  $P(7, 11, 37, 0)$  est composé des 9 tricubes numérotés 0, 7, 9, 10, 11, 17, 20, 21, 22 sur les pages précédentes. Par contre, le plan  $P(3, 4, 5, 0)$  ne possède que 6 tricubes numérotés 30, 32, 34, 38, 39.

Un théorème démontrant et généralisant la propriété précédente est présenté dans [52].

## 9.2.2 Application à la localisation des points réguliers d'une image

### Principe utilisé

Dans une image en 256 niveaux de gris, un pixel est pondéré par un nombre de 0 à 255 qui représente sa couleur. Nous considérons la position  $(x, y)$  d'un pixel dans l'image et la couleur comme niveau  $z$  associé. Par suite, le pixel et ses 8 voisins peuvent être considérés comme 9 voxels de l'espace. Un pixel sera dit *point régulier* de l'image si avec ses 8 voisins ils forment un tricube d'un plan discret.

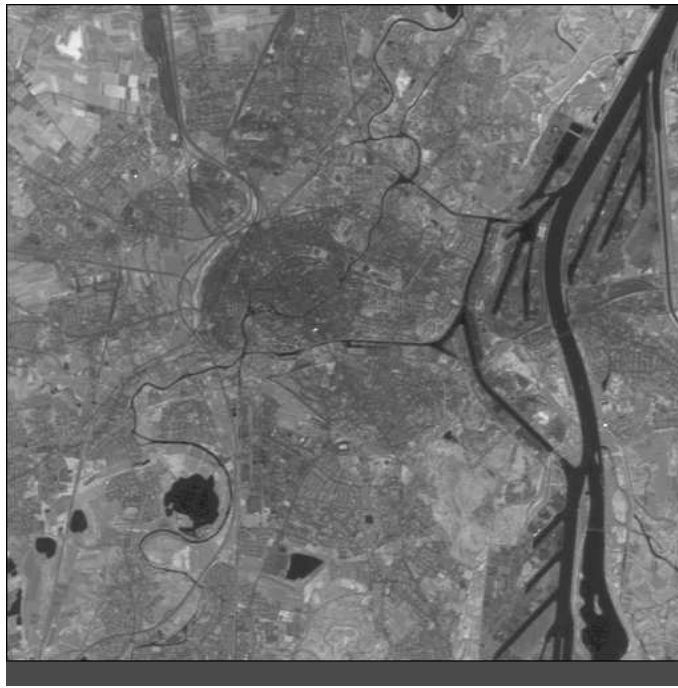
L'algorithme utilisé dans les exemples proposés ci-dessous détecte les points réguliers par un balayage horizontal de l'image et les colorie en gris clair. Les points non réguliers apparaissent en noir.

Dans les images bruitées la progression des couleurs n'est pas très régulière et pour pallier cela, nous avons introduit une notion de seuil définie par un entier  $i$ ; soit  $coul$  l'entier caractérisant la couleur associée à un pixel de coordonnées  $(x, y)$ , alors la troisième coordonnée  $z$  est définie par la valeur  $[\frac{coul}{i}]$ .

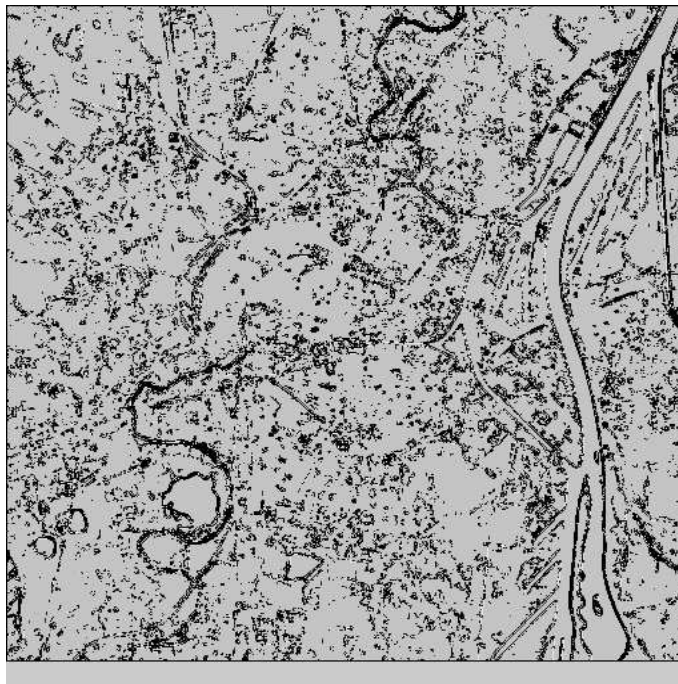
### Exemples :

#### 1. Image de Strasbourg

Cette image provient du satellite SPOT, elle a été captée en 1986 et est formée de  $512 * 512$  pixels. Cette image contient une grande quantité d'objets dans un espace radiométrique très restreint.



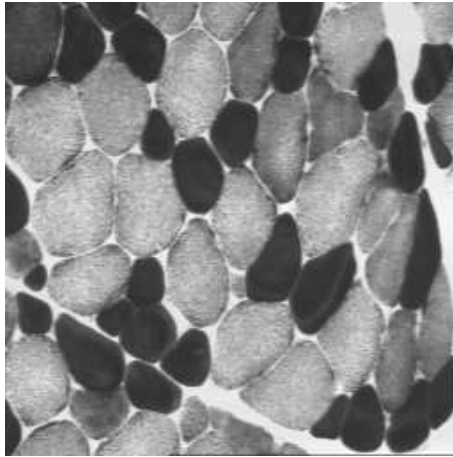
**Figure 9.1 :** *Image de télédétection de Strasbourg ville.*



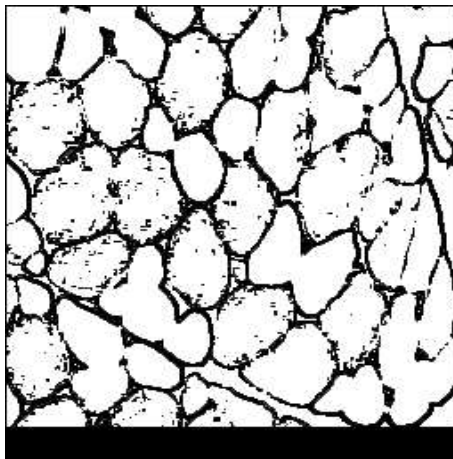
**Figure 9.2 :** *Détection des points réguliers avec un seuil égal à 25.*



## 2. Image d'un muscle



**Figure 9.3 :** *Image d'un muscle.*



**Figure 9.4 :** *Détection des points réguliers avec un seuil égal à 50.*

Les résultats obtenus avec cette méthode ne sont pas très satisfaisants, beaucoup d'objets ne sont pas détectés. En conservant le principe général décrit précédemment et en utilisant d'autres propriétés de géométrie discrète nous espérons pouvoir affiner la détection des points réguliers et obtenir de meilleurs résultats.

## 9.3 Polyédrisation d'objets voxels symétriques

### 9.3.1 Principe utilisé

Pour cette première tentative, nous nous sommes intéressés aux objets voxels symétriques par rapport à un 48ième d'espace. L'algorithme incrémental de reconnaissance de morceaux de plans discrets, présenté dans le chapitre précédent, est utilisé pour "découper" le 48ième d'espace de travail en morceaux de plans discrets partiellement rectangulaires. Ensuite, en fonction des symétries de l'objet, la polyédrisation du morceau situé dans le 48ième d'espace de travail est appliquée aux autres parties de l'objet. Le principe employé est le suivant :

Les voxels de l'objet présents dans le 48ième d'espace de travail sont placés dans un tableau  $T$  tel que, pour tout voxel  $(x, y, z)$  de l'objet présent dans le 48ième d'espace de travail on ait  $T[x][y] = z$ . Le domaine à reconnaître est géré grâce à des entiers qui "marquent" les zones n'ayant pas été parcourues. Trois autres tableaux sont utilisés :

- $TP$ , tableau temporaire où les niveaux ont été recopiés à partir de l'indice courant,
- $TI$ , tableau des indices indiquant la position des domaines non encore reconnus dans  $T$ ,
- $TC$ , tableau des couleurs où les points de chaque facette reconnue sont pondérés avec une couleur différente.

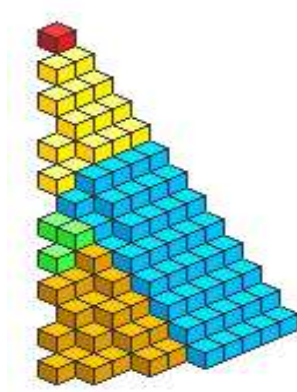
Le parcours du 48ième d'espace se fait ligne par ligne pour  $y$  fixé. La stratégie adoptée dans les exemples présentés par la suite consiste d'abord à regrouper dans une même facette les voxels situés au début du domaine ayant un même  $z$ . Puis, la première ligne tel que  $z$  ne soit pas constant est parcourue en appliquant l'algorithme de reconnaissance de segments de droites 2D (cf. chapitre 3), si l'algorithme échoue avant la fin de la ligne, le point échec est indicé dans  $TI$ , les caractéristiques obtenues avant l'échec (s'il y en a eu un) sont mémorisées. Les points de  $T$  situés au dessus du segment reconnu sont recopiés dans  $TP$  et l'algorithme de reconnaissance de morceaux de plans discrets est appliqué sur ces points avec des caractéristiques initiales égales à celles obtenues précédemment. Dès qu'un point ne peut pas appartenir à un plan contenant ce point et l'ensemble déjà parcouru, la reconnaissance s'arrête et le premier point de la ligne en cours de parcours est indicé dans  $TI$ . La zone correspondant à la facette reconnue est alors marquée dans  $TC$ . L'indice correspondant à la prochaine zone non encore parcourue est recherché dans  $TI$ , et, avec une procédure identique à celle décrite ci-dessus la reconnaissance se poursuit tant que le domaine entier n'a pas été reconnu.

Remarque : La stratégie adoptée est très simpliste et une polyédrisation optimale n'est pas forcément réalisée, par contre, la mémorisation des facettes est simple à

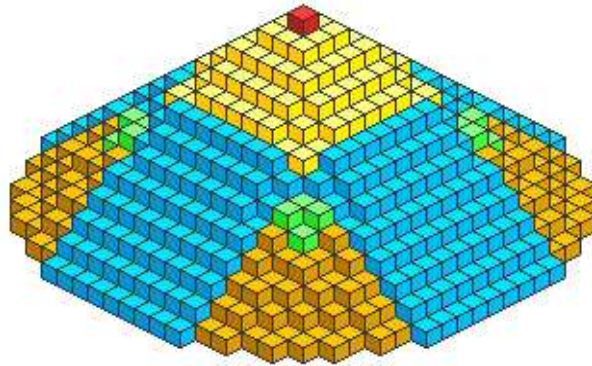
réaliser ; les caractéristiques doivent être retenues ainsi que deux voxels caractérisant la forme rectangulaire obtenue.

### 9.3.2 Exemples de polyédrisations d'objets voxels symétriques

#### 1. Polyédrisation d'un cône discret

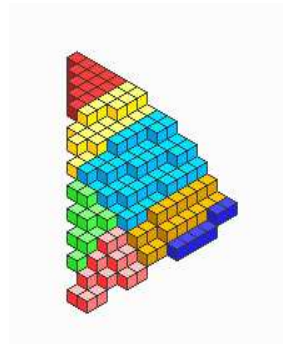


**Figure 9.5 :** *Morceau du cône dans le 48ième d'espace où l'algorithme de facettisation est appliqué.*

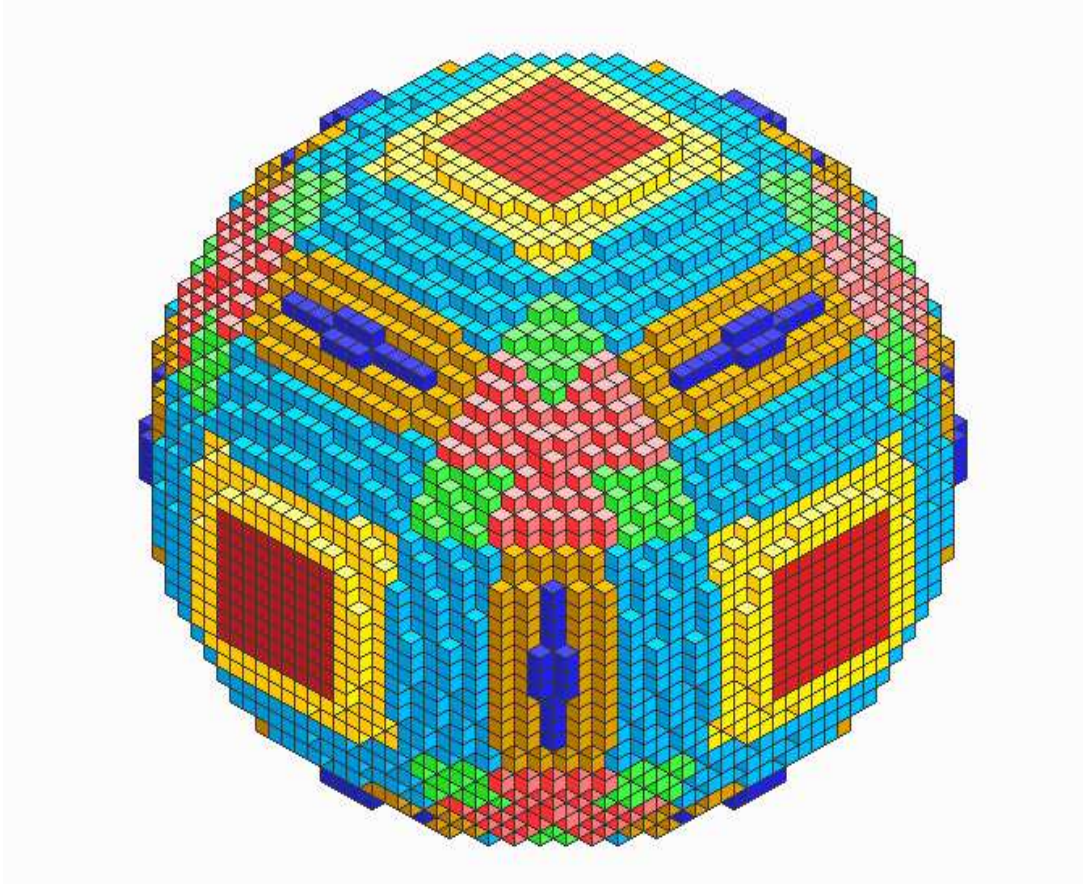


**Figure 9.6 :** *Polyédrisation résultant de l'application des symétries du cône ; les facettes de couleurs identiques possèdent le même vecteur normal au signe près.*

## 2. Polyédrisation d'une sphère discrète



**Figure 9.7 :** *Morceau de la sphère pour lequel l'algorithme de facettisation est appliqué.*



**Figure 9.8 :** *Polyédrisation résultant de l'application des symétries de la sphère ; les facettes de couleurs identiques possèdent le même vecteur normal au signe près.*

### 9.3.3 Nombreux problèmes restant à résoudre

La méthode présentée dans la sous-section précédente ne s'applique qu'aux objets dont les symétries sont connues. Dans la réalité, ce n'est pas le cas et plusieurs traitements doivent être réalisés avant d'utiliser l'algorithme de reconnaissance de morceaux de plans discrets. Les parties fonctionnelles doivent être localisées et cela demande déjà une technique appropriée qui n'est pas encore connue, de plus, plusieurs choix sont possibles, les critères de décision restent à déterminer. Une fois les parties fonctionnelles obtenues, la polyédrisation finale dépend de la manière de les parcourir et, en fonction du parcours adopté, le résultat peut aller du satisfaisant au désastreux, un pré-traitement orientant ce parcours est certainement nécessaire.

## 9.4 Conclusion

Dans ce chapitre, nous avons d'abord établi la notion de tricube d'un point dans un plan discret et nous avons fait un décompte précis du nombre de tricubes existants. Cette notion de voisinage d'un point est une première étape vers le calcul différentiel discret. De plus, l'application des tricubes à la localisation des points réguliers peut, en affinant les résultats utilisés, être un apport discret simple pour le traitement d'images comme les techniques proposées dans [11]. La notion de tricube a certainement un rôle à jouer dans la triangulation d'objets voxels, ce travail reste à réaliser.

La seconde partie de ce chapitre est consacrée à une première méthode de polyédrisation s'appliquant aux objets voxels dont les symétries sont connues. Cette technique utilise l'algorithme de reconnaissance de morceaux de plans discrets donné dans le chapitre précédent. Un gros travail reste à réaliser pour la polyédrisation d'objets voxels quelconques, un certain nombre de problèmes à résoudre ont été recensés.



# Chapitre 10

## Conclusion

Nous avons présenté dans ce mémoire des algorithmes originaux et performants de reconnaissance d'objets discrets ainsi que des propriétés inédites des droites et plans discrets. Ces algorithmes ont pu être réalisés grâce aux définitions mathématiques des objets concernés.

Après avoir fait un bref historique des très nombreux travaux effectués depuis 1970 au sujet de la droite discrète, nous avons montré l'intérêt et la richesse de sa définition arithmétique en élaborant un algorithme incrémental de reconnaissance de segments de droites discrètes naïves simple et naturel. Nous avons déduit de cet algorithme incrémental et des symétries des droites naïves un algorithme de polygonalisation de courbes discrètes qui est déjà appliqué en imagerie médicale ; compression de données, visualisation rapide de données scanner par couches (collaborations avec le CIRAD et l'EPFL).

Dans la pratique, les courbes à segmenter peuvent être 4-connexes et nous avons donc étendu l'algorithme incrémental de reconnaissance de segments de droites naïves aux droites 4-connexes. Un algorithme de reconnaissance de droites connexes d'épaisseurs quelconques, non incrémental, a aussi été élaboré.

Ces algorithmes de segmentation de courbes 2D ou 3D appliqués sur des exemples précis, comme les cercles discrets ou les ellipses pour le 2D, ont fait apparaître des dynamiques particulières qui méritent une étude plus approfondie.

En 3D, le besoin de définitions précises des objets discrets manipulés est criant ; dérouler un algorithme de tracé de droite discrète 3D pour déterminer si un point appartient à une droite n'est pas une solution efficace. C'est pourquoi nous avons établi une définition arithmétique de droite discrète 3D regroupant les structures engendrées par les algorithmes de construction existants. De plus, comme cette définition repose sur celle des droites discrètes 2D, nous en avons déduit un premier algorithme linéaire de segmentation de courbes discrètes 3D.

Néanmoins, beaucoup de questions restent en suspens, comme l'intersection d'un plan avec une droite et, une définition plus générale de la droite discrète 3D semble nécessaire pour apporter des réponses satisfaisantes.

Grâce à l'étude du plan discret arithmétique, des propriétés intéressantes et des algorithmes de parcours ont été obtenus. Quelques auteurs avaient déjà étudié la reconnaissance des morceaux de plans discrets mais les algorithmes proposés étaient complexes et non incrémentaux. A partir des propriétés des plans discrets et en utilisant uniquement des techniques de géométrie 2D, nous avons pu élaborer un algorithme incrémental de reconnaissance de morceaux de plans discrets de faible complexité.

Une perspective immédiate est d'étendre cet algorithme aux plans épais et à des domaines de reconnaissance plus généraux.



En établissant des stratégies efficaces de parcours et de détection des parties fonctionnelles, nous pensons obtenir un algorithme performant de *polyédrisation réversible d'objets voxels quelconques* grâce à l'algorithme incrémental de reconnaissance de morceaux de plans discrets.

La notion de voisinage discret autour d'un point, esquissée dans ce mémoire, est une première étape vers le *calcul différentiel discret*. Nous pouvons espérer de ce travail une notion de *k-régularité* des surfaces discrètes qui devrait redonner, par passage à la limite, la notion de variété différentiable du continu.

Deux directions utilisant la notion de tricube demandent à être développées :

- La technique utilisée dans la localisation des points réguliers d'une image peut, en affinant les résultats utilisés, se révéler être un apport discret simple au *traitement d'image* et une ouverture dans cette direction.
- La notion de tricube a certainement un rôle à jouer dans la *triangulation d'objets voxels*; ce travail reste à réaliser.

A l'issue de ce mémoire, des réflexions plus générales sont apparues :

- La notion de *polyèdre discret* demande une étude et une définition précise qui sont à élaborer.
- Les algorithmes de reconnaissance présentés dans ce mémoire ne conviennent pas aux *courbes ou surfaces bruitées* existant dans la réalité, une perspective intéressante serait de les adapter en tenant compte, selon des règles à établir, des bruits possibles dans les informations traitées.
- Comme nous l'avons montré à la fin du chapitre 3, l'algorithme de reconnaissance de segments de droites discrètes 2D est en relation étroite avec les fractions continues. Une question se pose alors immédiatement sur la relation entre les *fractions continues* et l'algorithme de reconnaissance de morceaux de plans discrets. Ce problème reste à étudier.

D'autres domaines d'étude peuvent aussi être envisagés :

- Un *lancer de rayon discret* performant pourra voir le jour en établissant des théorèmes discrets efficaces, notamment sur l'intersection entre une facette et une droite discrète 3D.
- Certains des algorithmes discrets déjà développés semblent se prêter naturellement à une parallélisation partielle ou complète. La *parallélisation des algorithmes discrets* est un domaine encore inexploré.
- Les structures rencontrées en *quasicristallographie* sont très proches des objets discrets que nous étudions et, la géométrie discrète pourrait constituer un apport dans ce domaine.

# Bibliographie

- [1] J. AMANATIDES and A. WOO. A fast voxel traversal algorithm for ray tracing. In *Eurographic's 87*, pages 3–12, 1987.
- [2] E. ANDRES. Le plan discret. In *Colloque de géométrie discrète en imagerie : fondements et applications*, Strasbourg, September 1993.
- [3] C. ARCELLI and A. MASSAROTTI. Regular arcs in digital contours. In *Comp. Grap. Image Process*, volume 4, pages 339–360, 1975.
- [4] C. ARCELLI and A. MASSAROTTI. On the parallel generation of straight digital lines. In *Comp. Grap. Image Process*, volume 7, pages 67–83, 1978.
- [5] J. BERSTEL. Tracé de droites, fractions continues et morphismes itérés. In *Mots, mélanges offerts à M.P. Schutzenberger*. Editions Hermes, 1987.
- [6] J. BERSTEL and M. POCCHIOLA. Random generation of finite sturmian worlds. In *Preprint, LIENS*, Ecole Normale Supérieure, Paris, 1993.
- [7] J.D. BOISSONNAT and B. GEIGER. Three-dimensional reconstruction of complex shapes based on Delaunay triangulation. In *Biomedical Image Processing and Biomedical Visualization*, volume 1905, pages 964–975, San Jose CA, Février 1993. SPIE.
- [8] Ph. BORIANNE and J. FRANÇON. Reversible polyhedrization of discrete volumes. In *Proc. DCGI'4*, pages 157–168, Grenoble, 1994.
- [9] R. BRONS. Linguistic methods for the description of a straight line on a grid. In *Comp. Grap. Image Process.*, volume 3, pages 48–62, 1974.
- [10] R. BRONS. Theoretical and linguistical methods for describing straight lines. In *NATO ASI Serie, Fundamental Algorithms for Computer Graphics*, volume F17. R.A. Earnshaw, Springer Verlag Berlin Heidelberg, 1985.
- [11] J.M. CHASSERY and A. MONTANVERT. *Géométrie discrète en analyse d'images*. Hermès, Paris, 1991.
- [12] I. DEBLED and J.P. REVEILLÈS. An incremental algorithm for digital plane recognition. In *Proc. DCGI'4*, pages 207–222, Grenoble, September 1994.
- [13] I. DEBLED and J.P. REVEILLÈS. A linear algorithm for segmentation of digital curves. In *Third International Workshop on parallel Image Analysis*, College Park, MD, Juin 1994.

- [14] I. DEBLED and J.P. REVEILLÈS. A new approach to digital planes. In *Vision Geometry III*, volume 2356, pages 12–21, Boston, November 1994. SPIE.
- [15] D.P. DOBKIN and J.I. MUNRO. Efficient uses of the past. In *Proc. 21st Annual Sympos. on Found. of Comput. Sci.*, pages 200–206, 1980.
- [16] L. DORST and A.W.M. SMEULDERS. Discrete representations of straight lines. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume PAMI-6, pages 450–463, 1984.
- [17] L. DORST and A.W.M. SMEULDERS. Decomposition of discrete curves into piecewise straight segments in linear time. In *Contemporary Mathematics*, volume 119, 1991.
- [18] M.E. DYER. Linear time algorithms for two- and three-variable linear programs. In *SIAM J. comput.*, pages 31–45, 1984.
- [19] O. FIGUEIREDO and J.P. REVEILLÈS. A contribution to 3D digital lines. In *Proc. DCGI'5*, pages 187–198, Clermont-Ferrand, 1995.
- [20] S. FORCHHAMMER. Digital plane and grid points segments. In *Computer Vision, Graphics, and Image Processing*, volume 47, pages 373–394, 1989.
- [21] J. FRANÇON. Sur la topologie d'un plan arithmétique. à paraître dans *Theor. Comput.Sc.*
- [22] J. FRANÇON. Arithmetic planes and combinatorial manifolds. In *Proc. DCGI'5*, pages 209–217, Clermont-Ferrand, 1995.
- [23] J. FRANÇON. Discrete combinatorial surfaces. In *CVGIP*, volume 57, pages 20–26, 1995.
- [24] H. FREEMAN. Boundary encoding and processing. In *Picture Processing and Psychopictorics*, pages 241–266. New York Academic, 1970.
- [25] G.H. HARDY and E.M. WRIGHT. *An introduction to the theory of numbers*. Oxford society, 1989.
- [26] G. HÉGRON. *Synthèse d'image : algorithmique élémentaire*. Bordas Paris, Dunod informatique, 1985.
- [27] S. H. Y. HUNG. On the straightness of digital arcs. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume PAMI-7, pages 203–215, 1985.
- [28] S. H. Y. HUNG and T. KASVAND. On the chord property and its equivalences. In *Proceedings 7th Int. Conf. on Pattern Recognition, Montreal*, pages 116–119, 1984.
- [29] M.A. JACOB. Les applications quasi-affines. Thèse. Université Louis Pasteur, Strasbourg, 1993.
- [30] A. KAUFMAN. An algorithm for 3-D scan conversion of polygons. In *Proc. Eurographic's 87*, Amsterdam, 1987.

- [31] A. KAUFMAN. Efficient algorithms for 3-D scan conversion of parametric curves, surfaces, volumes. In *Computer Graphic 21*, 4, pages 171–179, 1987.
- [32] A. KAUFMAN and E. SHIMONI. 3-d scan conversion algorithms for voxel-based graphics. In *Chapel Hill, NC*, pages 45–76, 1986.
- [33] C.E. KIM. Digital convexity, straightness and convex polygons. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume PAMI-4, pages 618–626, 1982.
- [34] C.E. KIM. Three-dimensional digital line segments. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume PAMI-5, pages 231–234, 1983.
- [35] C.E. KIM. Three-dimensional digital planes. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume PAMI-6, pages 639–645, 1984.
- [36] C.E. KIM and A. ROSENFELD. Convex digital solids. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume PAMI-4, pages 612–618, 1982.
- [37] C.E. KIM and A. ROSENFELD. Digital straight lines and convexity of digital regions. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume PAMI-4, pages 149–153, 1982.
- [38] C.E. KIM and I. STOJMENOVIĆ. On the recognition of digital planes in three-dimensional space. In *Pattern Recognition Letters 12*, pages 665–669, North-Holland, 1991.
- [39] R. KLETTE, I. STOJMENOVIĆ, and J. ZUNIĆ. A representation of digital planes by least square fits. In *ICALP'95*, 1995.
- [40] V. A. KOVALESKY. New definition and fast recognition of digital straight segments and arcs. In *Proceedings of the tenth international conference on Pattern Recognition*, Atlantic City, New Jersey, June 16-21 1990.
- [41] W. G. KROPATSCH and H. TOCKNER. Detecting digital curves. 1989.
- [42] LI-DE-WU. On the chain code of a line. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume PAMI-4, pages 347–353, 1982.
- [43] M. LINDENBAUM and A. BRUCKSTEIN. A recursive  $O(n)$  algorithm for segmenting chain into straight line portions. In *IEE T.PAMI*, 15/9, pages 919–953, 1993.
- [44] M. LINDENBAUM and J. KOPLOWITZ. A new parametrization of digital straight lines. In *IEE T.PAMI*, 13/4, pages 847–852, 1991.
- [45] W.E. LORENSEN and H.E. CLINE. Marching cubes : a high resolution 3D surface construction algorithm. In *Computer Graphics*, volume 21, pages 163–169, july 1987.

- [46] R. A. MELTER and A. ROSENFELD. New views of linearity and connectedness in digital geometry. In *Pattern Recognition Letters 10*, pages 9–16, North-Holland, 1989.
- [47] R. A. MELTER, I. STOJMENOVIĆ, and J. ZUNIĆ. A new characterization of digital lines by least square fits. In *Pattern Recognition Letters 14*, pages 83–88, North-Holland, 1993.
- [48] T. PAVLIDIS. Structural pattern recognition. New York : Springer-Verlag, 1987.
- [49] F.P. PREPARATA and S.J. HONG. Convex hulls of finite sets of points in two and three dimensions. In *Commun. ACM*, volume 20, pages 87–93, 1977.
- [50] J.P. REVEILLÈS. Géométrie discrète, calculs en nombre entiers et algorithmique. Thèse d'état. Université Louis Pasteur, Strasbourg, 1991.
- [51] J.P. REVEILLÈS. Structure des droites discrètes. In *Journées mathématique et informatique*, Marseille-Luminy, Octobre 1989.
- [52] J.P. REVEILLÈS. Combinatorial pieces in digital lines and planes. In *Vision Geometry IV*, San Diego, July 1995. SPIE.
- [53] J.P. REVEILLÈS and G. YAACOUB. A sublinear 3D convexhull algorithm for lattices. In *Proc. DCGI'5*, pages 219–232, Clermont-Ferrand, 1995.
- [54] C. RONSE. A simple proof of Rosenfeld's characterization of digital straight line segments. In *Pattern Recognition Letters*, pages 323–326, North-Holland, 1985.
- [55] A. ROSENFELD. Digital straight line segments. In *IEEE trans. on Computers*, volume 23, pages 1264–1369, 1974.
- [56] A. ROSENFELD, T.Y. KONG, and A.Y. WU. Digital surfaces. In *CVGIP*, volume 53, pages 305–312, 1991.
- [57] Y.M. SHARAIHA and P. GARAT. A compact chord property for digital arcs. In *Pattern Recognition*, 26(5), pages 799–803, 1993.
- [58] J. STOER and C. WITZGALL. Convexity and optimization in finite dimensions. In *Berlin : Springer*, 1970.
- [59] I. STOJMENOVIĆ and R. TOŠIĆ. Digitization schemes and the recognition of digital straight lines, hyperplanes and flats in arbitrary dimensions. In *Vision Geometry, contemporary Mathematics Series*, volume 119, pages 197–212. American Mathematical Society, Providence, RI, 1991.
- [60] J.P. THIRION and H.E. CLINE. The marching lines algorithm : new results and proofs. INRIA, rapport 1881, avril 1993.
- [61] A. TROESCH. Interprétation géométrique de l'algorithme d'Euclide. In *T.C.S.*, volume 115, pages 291–319, 1993.
- [62] P. VEERLAERT. On the flatness of digital hyperplanes. In *Journal of Mathematical Imaging and Vision*, volume 3, pages 205–221, 1993.

- [63] P. VEERLAERT. Digital planarity of rectangular surface segments. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 16, pages 647–652, 1994.
- [64] B. VIDAL. Vers un lancer de rayon discret. Thèse de doctorat, Lille, 1992.