

THÈSE

pour l'obtention du grade de

Docteur de l'Université de Poitiers

Faculté des Sciences Fondamentales et Appliquées

(Diplôme national – arrêté du 25 avril 2002)

École Doctorale : **Sciences Pour l'Ingénieur**
Secteur de Recherche : **Informatique et applications**

Présentée par

Martine DEXET

Architecture d'un modèleur géométrique à base topologique d'objets discrets et méthodes de reconstruction en dimensions 2 et 3

Soutenue le 5 décembre 2006
devant la commission d'examen

JURY

<i>Rapporteurs :</i>	Jean-Pierre BRAQUELAIRE, Professeur des Universités Jean-Marc CHASSERY, Directeur de Recherche CNRS	LaBRI, Bordeaux LIS, Grenoble
<i>Examineurs :</i>	Yves BERTRAND, Professeur des Universités David CŒURJOLLY, Chargé de Recherche CNRS	SIC, Poitiers LIRIS, Lyon
<i>Directeur de thèse :</i>	Éric ANDRES, Professeur des Universités	SIC, Poitiers

Remerciements

Durant mes quatre années de thèse, j'ai eu l'occasion de côtoyer maintes et maintes personnes croisées au détour des couloirs du laboratoire SIC et un peu ailleurs ... La plupart de ces personnes a joué un rôle important durant toute la période qu'a duré cette thèse. Je leur rends hommage dans ces quelques lignes.

Je commencerai par remercier tous les membres de mon jury de thèse :

- Mes deux rapporteurs, ACHILLE BRAQUELAIRE et JEAN-MARC CHASSERY ; je les remercie en particulier d'avoir eu la patience d'attendre la première version de ce mémoire.
- Mes examinateurs, YVES BERTRAND (merci aussi pour avoir toujours eu une oreille attentive) et DAVID CŒURJOLLY (un grand merci pour son implication dans ce travail)
- Mon directeur de thèse, ERIC ANDRES (merci pour la compréhension dont il a fait preuve lors de mon absence prolongée pour cause de *bébé*).

Je tiens de plus à remercier GUILLAUME DAMIAND pour avoir activement participé à ce travail durant ma première année de thèse. Je le remercie en particulier pour sa très grande disponibilité.

Je remercie aussi toutes ces personnes avec lesquelles j'ai eu l'occasion d'exercer le métier d'enseignant, et en particulier PATRICE NAUDIN et PIERRE CALLADINE pour leur gentillesse et la confiance qu'il m'ont accordée durant ces quatre années de vacances. Ce fut un honneur pour moi de travailler à leurs côtés.

Et puis, je remercie tous les thésards et non thésards du laboratoire SIC que cette thèse m'a amenée à rencontrer, pour nos discussions et les bons moments passés ensemble (et pour toutes les parties de coinche et de tarot que j'ai brillamment ... perdues). En vrac : MATTHIAS, PAS-CAL, ALEXANDRU, ADRIAN, RODOLPHE, BRUNO, les DAVID(S), SAM, FRED, YANN, RUDDY, les PHILIPPE(S), ALLAN, XAVIER, FRANÇOISE, DENIS, BENJAMIN, YANNICK, les FRANÇOIS, LUDO, GUILLAUME L., CARLOS, MONTAINE, BENOÎT, DAMIEN ...

Je remercie en particulier vivement NICOLAS pour toute l'aide qu'il m'a apportée durant ces quatre années de thèse. Tout d'abord, pour son inégalable maîtrise de la programmation (ou plutôt de l'informatique en général). Sans lui, et ses très performants algorithmes de calcul d'intersections entre surfaces [Gui06], mes magnifiques (?) objets froissés n'auraient certainement pas vu le jour aussi rapidement. Et puis, et surtout, je le remercie pour ses encouragements et le soutien qu'il m'a prodigué durant ces six dernières années, ainsi que pour m'avoir relayée en tant que parent au foyer pendant les périodes où je ne pouvais pas assurer ce rôle.

Enfin, j'adresse un immense **merci** à tous mes collègues¹ de l'équipe ARITH (LIRMM, Montpellier), ainsi qu'à quelques membres *adoptés*², pour m'avoir formidablement accueillie, aidée, encouragée, soutenue et pour m'avoir fait profiter de leur incomparable humour³ depuis que je suis arrivée parmi eux en août dernier. Merci à tous ...

Martine à la plage

¹Les chefs : ARNAUD (le nouveau chef), CHRISTOPHE, FABRICE, JEAN-CLAUDE, LAURENT, PATRICE, THIERRY, VALÉRIE (notre éternelle chefFE) ; et les pas encore chefs : JEAN-LUC, NADIA, NICO, THOMAS.

²GUILLAUME et MATHIAS.

³Humour toujours plus *subtil* (hein NONO?) d'un jour à l'autre ...

À Matthieu.

Table des matières

Introduction	1
1 Rappels de géométrie discrète	5
1.1 Introduction	5
1.2 Notions élémentaires	6
1.2.1 Notations et définitions	6
1.2.2 Espaces discrets	6
1.2.2.1 Pavages réguliers	7
1.2.2.2 Pavages irréguliers	7
1.2.3 Topologie discrète	8
1.2.3.1 Voisinage	8
1.2.3.2 Connexité	10
1.3 Modèles de discrétisation	11
1.3.1 Description analytique des hyperplans discrets	11
1.3.2 Le modèle Naïf	13
1.3.3 Le modèle Supercouverture	14
1.3.3.1 Supercouverture dans les espaces discrets classiques	14
1.3.3.2 Supercouverture dans d'autres espaces discrets	15
1.3.4 Le modèle Standard	15
1.3.4.1 Points et hyperplans Standard	16
1.3.4.2 Droite Standard en dimension 3	16
1.3.4.3 Description analytique Standard d'un segment	17
1.4 Conclusion	18
2 Une <i>préimage généralisée</i> pour la reconnaissance de primitives discrètes	19
2.1 Introduction	19
2.2 Représentation par dualité : un espace de paramètres	21
2.2.1 Représentation duale d'un point	21
2.2.1.1 Définitions	21

2.2.1.2	Propriétés	22
2.2.2	Objets duaux	23
2.2.2.1	Définitions et propriétés	23
2.2.2.2	Dual d'un polytope	25
2.2.3	Application à l'espace discret classique	29
2.2.3.1	Description du dual d'un hypervoxel	29
2.2.3.2	Exemples en dimensions 2 et 3	30
2.2.4	Cas d'une droite en dimension 3	31
2.3	La notion de préimage généralisée	33
2.3.1	Définition	33
2.3.2	Préimage généralisée d'hypervoxels	37
2.3.2.1	Étude de la préimage généralisée d'un ensemble de pixels 1-connexe	37
2.3.2.2	Cas d'un ensemble de voxels 2-connexe	38
2.4	Reconnaissance d'hyperplans discrets	39
2.4.1	Algorithme de reconnaissance	39
2.4.2	Exemples d'applications	39
2.4.2.1	Hyperplans Supercouverture	39
2.4.2.2	Hyperplans Standard	41
2.4.2.3	Modèles Naïfs	41
2.5	Conclusion	41
3	Reconstruction analytique inversible de courbes et volumes discrets	43
3.1	Introduction	44
3.2	Reconstruction inversible en dimension 2	46
3.2.1	Méthodes de reconstruction proposées par R. BRETON [Bre03]	47
3.2.1.1	Méthode avec joints [BSDA03]	47
3.2.1.2	Méthode sans joints [SBDA05]	48
3.2.1.3	Bilan	49
3.2.2	Méthode étendue	50
3.2.2.1	Principe	50
3.2.2.2	Reconnaissance des segments discrets	52
3.2.2.3	Polygonalisation de la courbe	53
3.2.3	Résultats	54
3.2.3.1	Méthode étendue	54
3.2.3.2	Comparaison avec les méthodes avec et sans joints	56
3.2.4	Reconstruction de courbes non simples	56
3.2.5	Conclusion et perspectives	59

3.3	Reconstruction de courbes et volumes discrets en dimension 3	63
3.3.1	Données traitées	64
3.3.2	Principe	64
3.3.2.1	Surfel de départ et méthodes de propagation	65
3.3.2.2	Ordre de reconnaissance	67
3.3.3	Reconnaissance de plan contrainte	67
3.3.3.1	Contrainte de sommet	67
3.3.3.2	Contrainte d'arête	67
3.3.3.3	Choix du plan	68
3.3.4	Polygonalisation de face	68
3.3.4.1	Reconstruction d'une 2-courbe discrète planaire	69
3.3.4.2	Ensemble de solution de surfels vide	72
3.3.5	Résultats	72
3.3.6	Conclusion et perspectives	78
3.4	Conclusion du chapitre	85
4	Un modelleur géométrique à base topologique d'objets discrets	87
4.1	Introduction	87
4.2	Le modèle topologique des cartes généralisées	89
4.2.1	Définition d'une carte généralisée	90
4.2.2	Cellules topologiques	90
4.2.3	Opérations de base	91
4.3	Une structure hiérarchique	92
4.3.1	Les niveaux	94
4.3.1.1	Niveau 3 : la représentation euclidienne	94
4.3.1.2	Niveau 2 : la représentation discrète analytique	95
4.3.1.3	Niveau 1 : les contours discrets	95
4.3.1.4	Niveau 0 : représentation discrète classique duale	96
4.3.1.5	Mise en œuvre	97
4.3.2	Contraintes de cohérence	98
4.4	Opérations dans la structure	99
4.4.1	Construction des niveaux et mise en place des liaisons	100
4.4.1.1	Discrétisation d'un objet euclidien	100
4.4.1.2	Reconstruction d'un objet discret	107
4.4.2	Modification et mise à jour des niveaux	108
4.4.2.1	Manipulation des représentations	109
4.4.2.2	Mise à jour de la structure	109

4.5 Conclusion et perspectives	115
Bibliographie	121
Table des figures	129

Introduction



FIG. 1 – *Happy Buddha* : image composée de 372×935 pixels.

Les *images numériques* sont constituées d'un ensemble de données discrètes appelées *pixels* en dimension 2 (voir Figure 1⁴) et *voxels* en dimension 3. Ces images sont le plus souvent le résultat d'un processus d'acquisition (imagerie médicale, photographie numérique, ...). Elle peuvent par ailleurs être créées (assemblages de pixels ou de voxels) par le biais de logiciels dédiés à la création et à la manipulation de telles données.

Dans le domaine du traitement d'images, un problème important est celui de la reconnaissance des formes perçues dans une image, et en particulier la reconnaissance de formes linéaires, telles que les droites. Essayer de faire le rapprochement entre les informations discrètes présentes dans une image et les informations "continues" dont elles sont éventuellement issues est ici primordial.

Plusieurs questions se posent alors naturellement à nous, telles que : qu'est-ce qu'une droite discrète ? Comment détecter une droite discrète dans une image ? Comment obtenir une droite discrète à partir d'une droite euclidienne ? Ou le problème inverse : comment obtenir une droite euclidienne à partir d'une droite discrète ?

Afin de répondre à ces différentes questions, le domaine de la *géométrie discrète* fournit un ensemble d'outils adaptés à la nature discrète des images.

Dans ce domaine, diverses définitions de primitives discrètes telles que des droites, plans, ou plus généralement hyperplans ont été proposées [Bre65, Rev91, Rev95, And00, COK95, And03]. Les premières droites discrètes ont tout d'abord été décrites de manière algorithmique par J. BRESENHAM [Bre65], puis une description analytique des hyperplans discrets a été proposée par J.P. REVEILLÈS [Rev91], fournissant ainsi une description en compréhension des objets discrets plutôt qu'en extension (énumération des données discrètes).

À partir de ces définitions, un ensemble de méthodes de *reconnaissance* de primitives discrètes a été proposé [FST96, Kim84, Buz03, KS91, GDRZ05, DR95, Siv04, VC99, BD05, ST91, Vit99]. L'objectif de ces méthodes est d'indiquer s'il existe des primitives discrètes contenant un ensemble de données discrètes données, et le cas échéant déterminer les différents paramètres de ces primitives.

⁴Image extraite de la page web <http://graphics.stanford.edu/data/3Dscanrep/>.

Enfin, de nombreux travaux visant à établir des relations entre le domaine discret et le domaine continu (ou *euclidien*), concernant notamment l’obtention d’une description discrète d’un objet euclidien et inversement, ont été effectués. Ces deux procédés appelés *discrétisation* et *reconstruction* sont un sujet important de recherche en géométrie discrète.

De nombreux modèles et méthodes de discrétisation associant à une primitive euclidienne une unique primitive discrète ont été proposés, principalement pour l’obtention d’hyperplans discrets [Rev95, And00, And03], et de droites discrètes en dimension 3 [And00, Kim83, DR95], mais aussi plus récemment, pour l’obtention de cercles discrets [And92, Cœu02, FJT06], sphères et hyperphères discrètes [FT06], ou encore de courbes polynomiales discrètes [FT07].

Le problème de la reconstruction de données discrètes a lui aussi fait l’objet de nombreux travaux, exclusivement en dimensions 2 et 3 [BF94, SBDA05, FP99, SDC05, Bre03, Siv04], et notamment dans le but d’obtenir une reconstruction inversible pour un modèle donné, c’est à dire telle que la discrétisation de l’objet obtenu suivant ce modèle est identique aux données discrètes d’origine. L’obtention d’une reconstruction inversible est un problème crucial en géométrie discrète. En effet, non seulement, l’objet reconstruit est “proche” de l’objet discret d’origine, mais nous sommes de plus assurés qu’il n’y a ni ajout, ni perte d’information entre l’objet discret d’origine et la discrétisation de l’objet euclidien obtenu.

Dans cette thèse, nous nous sommes intéressés à la fois au problème de la reconnaissance et au problème de la reconstruction de données discrètes.

Nous avons tout d’abord proposé une nouvelle méthode de reconnaissance d’hyperplans discrets. Cette méthode est basée sur le calcul d’un objet géométrique particulier, que nous avons baptisé *préimage généralisée*. Cet objet est défini dans un espace euclidien adapté appelé *espace de paramètres*, et fournit l’ensemble des paramètres des hyperplans reconnus. De plus, cette méthode présente l’avantage de permettre la reconnaissance d’hyperplans discrets obtenus avec diverses méthodes et modèles de discrétisation.

Nous avons ensuite développé deux algorithmes de reconstruction inversible d’objets discrets en dimensions 2 et 3. Ces deux algorithmes sont basés sur l’algorithme de reconnaissance d’hyperplans discrets présenté dans ce mémoire, et fournissent une reconstruction inversible pour le modèle de discrétisation Standard [And03, And00].

Le premier algorithme permet la reconstruction de courbes discrètes en dimension 2, ouvertes ou fermées, simples ou non simples, et améliore notamment les résultats obtenus par R. BRETON dans sa thèse [Bre03].

Le deuxième algorithme permet la reconstruction de la surface de volumes discrets en une surface polygonale. Le principal intérêt de cette méthode est qu’elle fournit un résultat topologiquement cohérent avec l’objet discret d’origine et constitue par la même une réponse aux problèmes de non cohérence topologique ou de non réversibilité de la reconstruction rencontrés par I. SIVIGNON dans sa thèse [Siv04], et corrigé ensuite dans [CGS04] grâce à l’utilisation d’une méthode basée sur l’algorithme des Marching Cubes [LC87].

Parallèlement à ces travaux sur la reconnaissance et la reconstruction de données discrètes, nous nous sommes intéressés aux problèmes de la modélisation, de la manipulation et du traitement des objets discrets. Notre but était de mettre en œuvre nos différents algorithmes au sein d’un logiciel de modélisation d’objets géométriques, ces objets devant être représentés aussi bien sous forme discrète qu’euclidienne.

La motivation principale de ce travail provient du constat suivant : les lois applicables en géométrie euclidienne ne le sont généralement pas en géométrie discrète. Par exemple, deux droites discrètes n’ont pas obligatoirement de point d’intersection discret (voir Figure 2a). De

même, deux droites discrètes parallèles peuvent avoir une intersection non vide (voir Figure 2b). Ou encore, deux points discrets ne définissent a priori pas une seule et unique droite discrète (voir Figure 2c).

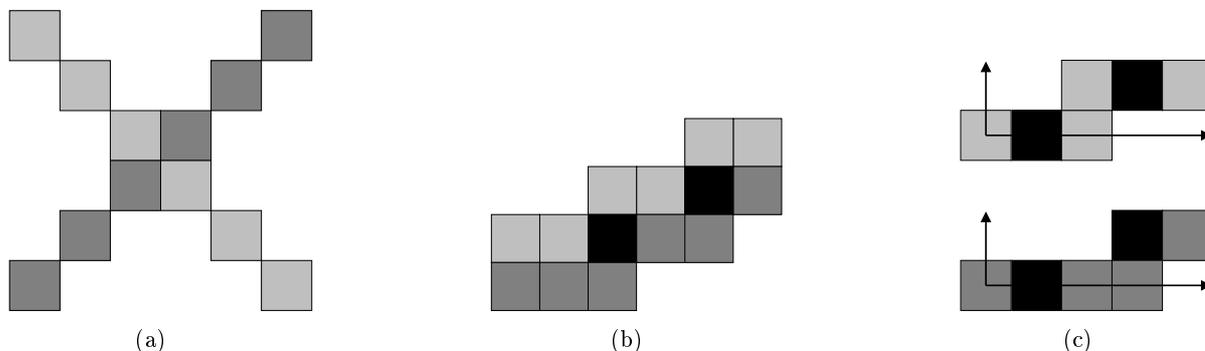


FIG. 2 – Quelques propriétés du monde discret : (a) Deux droites discrètes séquentes sans intersection, (b) Deux droites discrètes parallèles avec intersection, (c) Deux droites discrètes différentes contenant les deux mêmes pixels.

De même, la mise en œuvre de certaines opérations, telles que la rotation, s'avère être relativement simple dans l'espace euclidien, alors qu'elle présente de réelles difficultés dans l'espace discret [And96]. À contrario, l'intersection de deux objets est aisément mise en œuvre dans le monde discret, alors qu'elle l'est nettement moins dans le monde euclidien.

Ainsi, il nous a semblé qu'afin de pouvoir manipuler plus aisément un objet géométrique, disposer d'une représentation discrète et d'une représentation continue de cet objet présentait un intérêt certain.

Dans cette optique, nous avons développé un logiciel de modélisation permettant la création, l'importation et le traitement d'objets géométriques représentés à la fois sous forme discrète et polygonale (dimension 2) ou polyédrique (dimension 3). Une des particularités de ce logiciel est que les représentations discrète et continue d'un même objet coexistent à l'intérieur d'une même structure. Il est ainsi possible, en fonction de l'opération que l'on souhaite effectuer, de choisir la représentation la mieux adaptée. De plus, le noyau de ce logiciel est basé sur un modèle topologique : *les cartes généralisées* [Lie89, Lie94]. Ainsi, nous pouvons représenter topologiquement aussi bien les objets euclidiens que les objets discrets. Cette propriété est particulièrement intéressante au niveau discret car habituellement, nous disposons uniquement des relations d'adjacence entre pixels (ou voxels), ce qui dans de nombreux cas de figure n'est pas suffisant. En effet, nous pouvons rapidement savoir si un pixel se trouve "à côté" d'un autre pixel, mais nous ne pouvons pas aisément déterminer quels sont les pixels qui sont contenus dans un même objet discret. La représentation topologique des pixels (ou voxels) permet alors de pallier ce type problème.

Organisation du mémoire

Ce mémoire de thèse est constitué de quatre chapitres.

Le premier chapitre est consacré à des rappels de géométrie discrète. Nous donnons tout d'abord quelques définitions relatives aux espaces discrets, ainsi qu'aux primitives discrètes. Nous donnons ensuite les définitions de plusieurs modèles de discrétisation existants.

Dans les deuxième et troisième chapitres, nous abordons les problèmes de la reconnaissance et de la reconstruction de données discrètes. Dans le chapitre 2, nous présentons tout d'abord l'espace de paramètres sur lequel sont basées nos méthodes de reconnaissance et de reconstruction d'objets discrets. Nous présentons ensuite notre algorithme de reconnaissance. Enfin, dans le chapitre 3, nous décrivons nos méthodes de reconstruction d'objets discrets en dimensions 2 et 3.

Le quatrième et dernier chapitre de ce mémoire consiste en une rapide description de notre logiciel de modélisation d'objets géométriques. Nous faisons tout d'abord quelques rappels concernant le modèle topologique des cartes généralisées. Nous présentons ensuite la structure de données que nous avons conçue, et détaillons les opérations internes à cette structure, notamment les opérations permettant sa construction.

Enfin, nous concluons et donnons plusieurs perspectives à ces travaux.

Chapitre 1

Rappels de géométrie discrète

Sommaire

1.1	Introduction	5
1.2	Notions élémentaires	6
1.2.1	Notations et définitions	6
1.2.2	Espaces discrets	6
1.2.2.1	Pavages réguliers	7
1.2.2.2	Pavages irréguliers	7
1.2.3	Topologie discrète	8
1.2.3.1	Voisinage	8
1.2.3.2	Connexité	10
1.3	Modèles de discrétisation	11
1.3.1	Description analytique des hyperplans discrets	11
1.3.2	Le modèle Naïf	13
1.3.3	Le modèle Supercouverture	14
1.3.3.1	Supercouverture dans les espaces discrets classiques	14
1.3.3.2	Supercouverture dans d'autres espaces discrets	15
1.3.4	Le modèle Standard	15
1.3.4.1	Points et hyperplans Standard	16
1.3.4.2	Droite Standard en dimension 3	16
1.3.4.3	Description analytique Standard d'un segment	17
1.4	Conclusion	18

1.1 Introduction

Dans ce chapitre, nous rappelons différentes notions de géométrie discrète qui seront utilisées dans la suite de ce mémoire.

Nous donnons tout d'abord un ensemble de définitions élémentaires relatives aux objets discrets tels que espaces discrets, les points discrets, ou encore les courbes discrètes. Nous présentons ensuite plusieurs modèles de discrétisation existants auxquels nous ferons référence par la suite.

1.2 Notions élémentaires

Cette section est consacrée à l'introduction de différentes notions que nous utiliserons tout au long de ce mémoire. Nous définissons tout d'abord les différents espaces discrets dans lesquels nous serons amenés à travailler, puis les relations topologiques entre les différents éléments composant ces espaces.

1.2.1 Notations et définitions

Dans toute la suite de ce mémoire, $n \in \mathbb{N}^*$ désignera la *dimension* de l'espace dans lequel nous nous plaçons. De plus, pour tout $k \in \mathbb{N}^*$, $\llbracket 1, k \rrbracket$ désignera le sous-ensemble des entiers naturels compris entre 1 et k . Enfin, nous définissons un α -hypercube, $\alpha \in \mathbb{R}_+$, de la manière suivante :

Définition 1 (α -Hypercube) *L'hypercube (ou cube de dimension n) de centre $(c_1, \dots, c_n) \in \mathbb{R}^n$, et de taille $\alpha \in \mathbb{R}_+$, est l'ensemble des points $p = (p_1, \dots, p_n) \in \mathbb{R}^n$ vérifiant*

$$\forall i \in \llbracket 1, n \rrbracket, c_i - \frac{\alpha}{2} \leq p_i \leq c_i + \frac{\alpha}{2}$$

Nous pouvons voir sur la figure 1.1 quelques illustrations d' α -hypercubes en dimensions 2, 3, 4 et 5. Nous appelons *hypervoxel* tout hypercube de taille unitaire ($\alpha = 1$). En particulier, les hypervoxels de dimensions 2 et 3 sont respectivement appelés *pixels* et *voxels*.

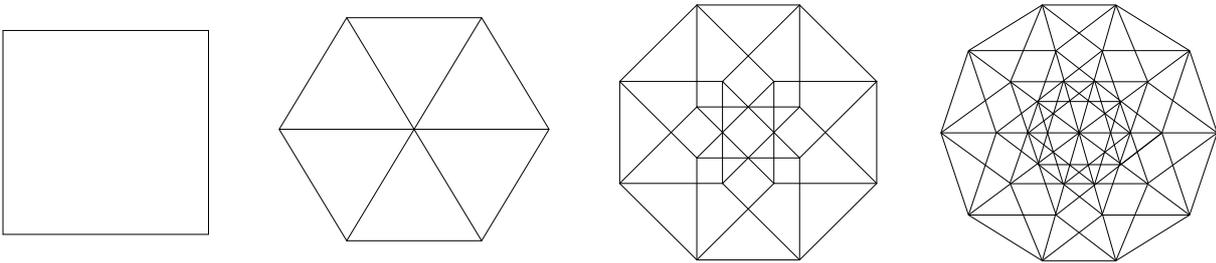


FIG. 1.1 – De gauche à droite : projections symétriques en dimension 2 des hypercubes de dimensions respectives 2 (un carré), 3 (un cube), 4 (un *tesseract*) et 5. Notons que sur certains exemples, telle que la deuxième figure, plusieurs sommets de l'objet coïncident.

Définition 2 (Boule fermée) *Soit d une distance sur \mathbb{R}^n . Alors, la boule fermée $B_d(c, r)$ de centre $c \in \mathbb{R}^n$ et de rayon $r \in \mathbb{R}$ est définie par*

$$B_d(c, r) = \{p \in \mathbb{R}^n \mid d(c, p) \leq r\}$$

1.2.2 Espaces discrets

Ces espaces sont constitués de points isolés⁵ de l'espace euclidien \mathbb{R}^n , appelés *points discrets* de dimension n , auxquels sont associés des pavages de \mathbb{R}^n , tels que par exemple les zones de Voronoï. nous appellerons *pavé* tout élément d'un tel espace. De multiples espaces discrets peuvent être

⁵Soit d une distance sur \mathbb{R}^n . Les points d'un ensemble $E \subseteq \mathbb{R}^n$ sont dits *isolés* si pour tout point $p \in E$ il existe $\epsilon \in \mathbb{R}_+^*$ tel que la boule fermée $B_d(p, \epsilon)$ ne contient que p .

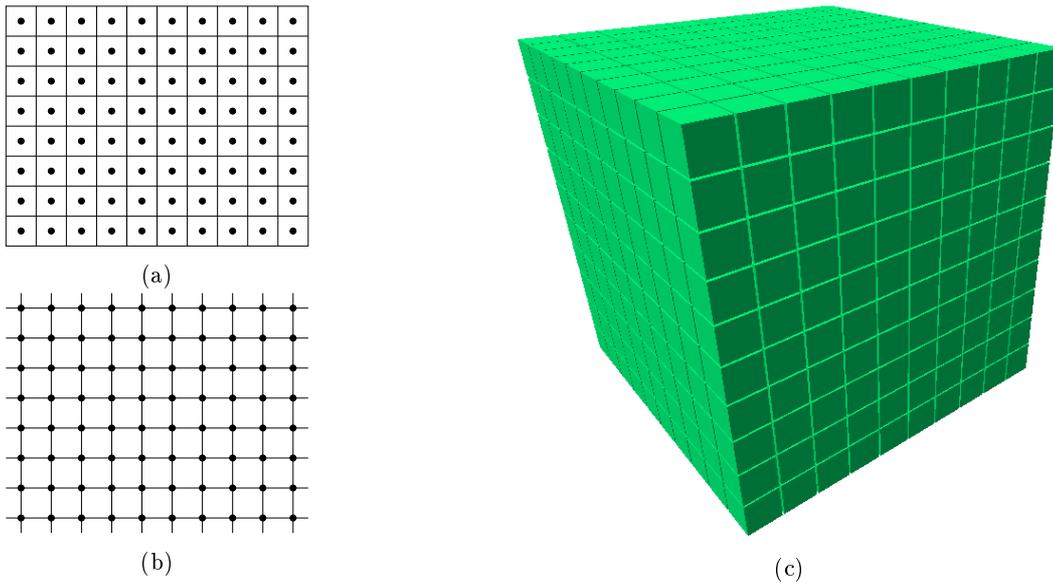


FIG. 1.2 – Espaces discrets classiques en dimensions 2 et 3 : (a) et (b) Deux pavages possibles de l'espace par des pixels ; chaque point discret est matérialisé par un point noir. (c) Pavage de l'espace par des voxels.

considérés et utilisés. Dans cette thèse, nous nous intéressons principalement aux pavés formés d'unions de *polytopes*, de tailles et de formes variées.

Nous distinguons en particulier deux types d'espaces : les espaces correspondant à un pavage régulier de \mathbb{R}^n , et les espaces correspondant à un pavage irrégulier.

1.2.2.1 Pavages réguliers

Une première catégorie d'espaces discrets correspond aux pavages réguliers de \mathbb{R}^n , c'est à dire composés de pavés convexes ou concaves, mais tous identiques. Nous pouvons voir quelques exemples de tels espaces sur les figures 1.2, 1.3a et 1.3b. Nous parlons alors d'*espace discret régulier*.

Parmi, ces espaces, nous trouvons les espaces discrets qui sont étroitement liés à la représentation discrète des images. Ces espaces, que nous qualifierons de *classiques*, sont à l'heure actuelle les espaces les plus étudiés et utilisés en infographie. Dans tout ce mémoire, nous assimilerons l'*espace discret classique* de dimension n à l'ensemble \mathbb{Z}^n .

Il existe plusieurs représentations de cet espace discret telles que celles décrites sur la figure 1.2. La représentation la plus utilisée correspond à un pavage de l'espace réel \mathbb{R}^n à l'aide d'hypervoxels dont le centre est un point discret (voir Figures 1.2a et 1.2c). Une autre représentation, duale de la première, consiste à paver l'espace à l'aide d'hypervoxels dont le centre est un point à coordonnées demi-entières. Nous appelons cet espace l'*espace discret classique dual* (voir Figure 1.2b).

1.2.2.2 Pavages irréguliers

Dans ce mémoire, nous nous intéresserons aussi à d'autres types d'espaces discrets peu utilisés jusqu'à présent, mais qui connaissent actuellement un intérêt grandissant. Cette deuxième catégorie d'espaces correspond à des pavages irréguliers de \mathbb{R}^n , c'est à dire des pavages de l'espace

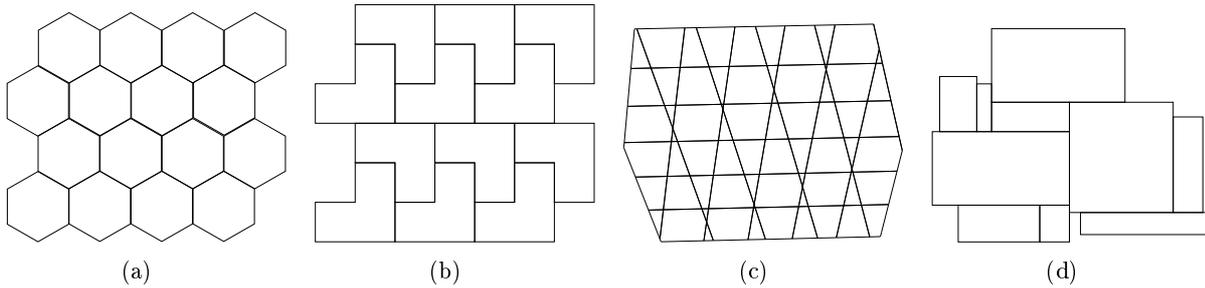


FIG. 1.3 – Exemples de pavages réguliers et irréguliers en dimension 2 : (a) Pavage régulier par des hexagones. (b) Pavage régulier par des pavés concaves. (c) Pavage irrégulier. (d) Grille irrégulière isothétique.

à l'aide de pavés de formes différentes. Nous appelons ces espaces des *espaces discrets irréguliers* (voir Figures 1.3c et 1.3d). Parmi ces derniers, nous distinguons les pavages en dimension 2 formés de rectangles dont les arêtes sont parallèles aux axes du repère de l'espace euclidien [SB01, SB02], aussi appelés *grilles irrégulières isothétiques* [Cœu05, CZ06] (voir Figure 1.3d). Notons que ces pavages incluent notamment les espaces discrets classiques et peuvent pas exemple être utilisés dans le cadre de la segmentation *multi-échelle* d'images [GS06], ainsi que pour représenter des structures telles que les *quad-trees* [FB74].

Dans la suite, nous noterons \mathbb{D}^n un espace discret quelconque de dimension n .

1.2.3 Topologie discrète

Dans cette section, nous introduisons les notions de voisinage et de connexité entre points discrets afin d'établir des relations d'adjacence et d'incidence entre les pavés d'un espace discret.

1.2.3.1 Voisinage

Dans le cadre des espaces discrets classiques (voir Section 1.2.2.1), nous utilisons la définition de *voisinage* entre points discrets suivante :

Définition 3 (k -voisinage [And00]) Soit $k \in \llbracket 0, n - 1 \rrbracket$. Deux points discrets $p = (p_1, \dots, p_n) \in \mathbb{Z}^n$ et $q = (q_1, \dots, q_n) \in \mathbb{Z}^n$ sont dits k -voisins si

$$\forall i \in \llbracket 1, n \rrbracket, |p_i - q_i| \leq 1$$

et

$$k \leq n - \sum_{i=1}^n |p_i - q_i|$$

Autrement dit, deux points discrets p et q en dimension 2 sont 0-voisins si les pixels P et Q , de centres respectifs p et q ont au moins un sommet commun. De même, p et q sont 1-voisins si P et Q ont au moins une arête commune. La figure 1.4 illustre la notion de k -voisinage en dimension 2.

Si nous appliquons la définition de k -voisinage à la dimension 3, nous obtenons la caractérisation suivante : deux points discrets p et q sont 0-voisins si leurs voxels respectifs P et Q ont

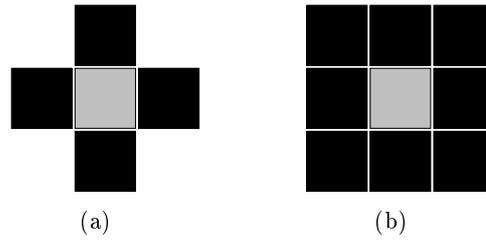


FIG. 1.4 – Exemples de voisinages en dimension 2 : (a) Un pixel gris et ses quatre 1-voisins (en noir). (b) Le même pixel avec ses huit 0-voisins.

au moins un sommet commun. De même, p et q sont 1-voisins si P et Q ont au moins une arête commune. Enfin, p et q sont 2-voisins si P et Q ont au moins une face commune. La figure 1.5 illustre la notion de k -voisinage en dimension 3.

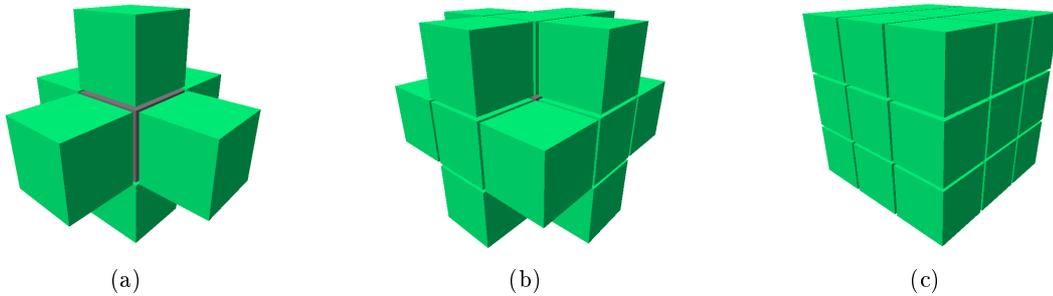


FIG. 1.5 – Exemples de voisinages en dimension 3 : (a) Un voxel (au centre) et ses six 2-voisins. (b) Le même voxel et ses dix-huit 1-voisins. (c) Le même voxel et ses vingt-six 0-voisins.

Remarque 1 Nous pouvons remarquer que cette définition de voisinage est directement liée aux relations d'adjacence entre les hypervoxels. Elle présente de plus l'intérêt d'être homogène en toutes dimensions, contrairement à d'autres définitions du voisinage classiquement utilisées, telles que celles présentées dans le tableau 1.1.

	Notations utilisées	Notations classiques
Dimension 2	1-voisinage	4-voisinage
	0-voisinage	8-voisinage
Dimension 3	2-voisinage	6-voisinage
	1-voisinage	18-voisinage
	0-voisinage	26-voisinage

TAB. 1.1 – Correspondance entre les notations de k -voisinage en dimensions 2 et 3 que nous utilisons et les notations classiques.

Dans le cadre d'espaces discrets non plus classiques, mais quelconques (voir Section 1.2.2), nous pouvons suite à la remarque 1 donner une autre définition du k -voisinage :

Définition 4 (Généralisation du k -voisinage) Soit $k \in \llbracket 0, n - 1 \rrbracket$. Deux points discrets p et q d'un espace discret \mathbb{D}^n sont dits k -voisins si leurs pavés associés P et Q sont adjacents par au moins une cellule topologique de dimension k .

1.2.3.2 Connexité

A partir de la définition de voisinage précédemment donnée, nous déduisons les notions de *courbe* et de *connexité* discrètes.

Définition 5 (k -chemin) Soient $k \in \llbracket 0, n - 1 \rrbracket$, et $\mathcal{C} = \{p_1, \dots, p_m\} \subset \mathbb{D}^n$ une suite de m points discrets, $m \in \mathbb{N}_+^*$. Alors, \mathcal{C} est un k -chemin si

$$\forall i \in \llbracket 1, m - 1 \rrbracket, p_i \text{ et } p_{i+1} \text{ sont } k\text{-voisins}$$

Définition 6 (k -connexité) Soit \mathcal{O} un ensemble de points discrets. \mathcal{O} est dit k -connexe si et seulement si il existe un k -chemin entre deux éléments quelconques de \mathcal{O} .

Définition 7 (k -courbe) Soient $k \in \llbracket 0, n - 1 \rrbracket$, et $\mathcal{C} = \{p_1, \dots, p_m\} \subset \mathbb{D}^n$ une suite de m points discrets, $m \in \mathbb{N}_+^*$. Alors, \mathcal{C} est une k -courbe discrète si \mathcal{C} est un k -chemin tel que

$$\forall i \in \llbracket 2, m - 2 \rrbracket, p_i \text{ a exactement deux } k\text{-voisins}$$

De plus, \mathcal{C} est une k -courbe **fermée** si p_1 et p_m sont k -voisins. Dans le cas contraire, \mathcal{C} est dite **ouverte**.

Enfin, si \mathbb{D}^n est un espace discret régulier, nous parlerons de k -courbe **régulière**. En particulier, une k -courbe dans l'espace discret classique sera appelée une k -courbe **classique**. De manière similaire, si \mathbb{D}^n est un espace discret irrégulier, nous parlerons de k -courbe **irrégulière**.

La figure 1.6 montre quelques exemples de k -courbes dans les espaces discrets classiques de dimensions 2 et 3. La figure 1.7, quant à elle, illustre deux 1-courbes irrégulières.

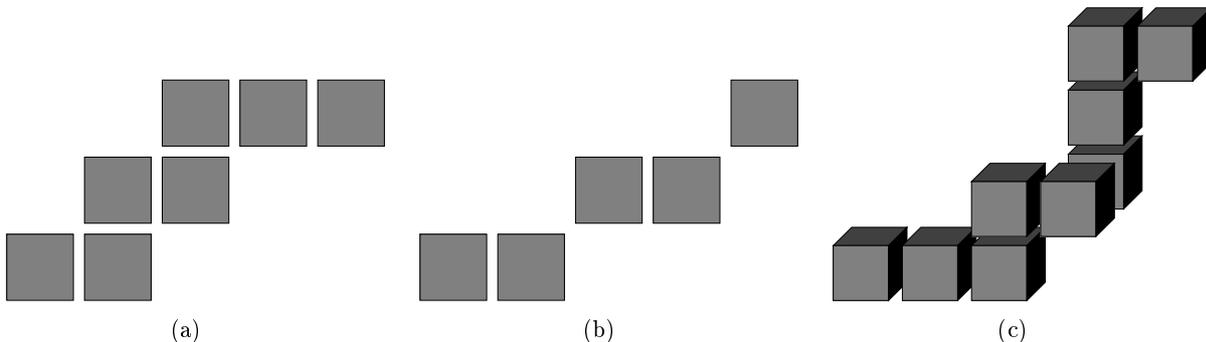


FIG. 1.6 – Exemples de k -courbes : (a) Une 1-courbe en dimension 2. (b) Une 0-courbe en dimension 2. (c) Une 2-courbe en dimension 3.

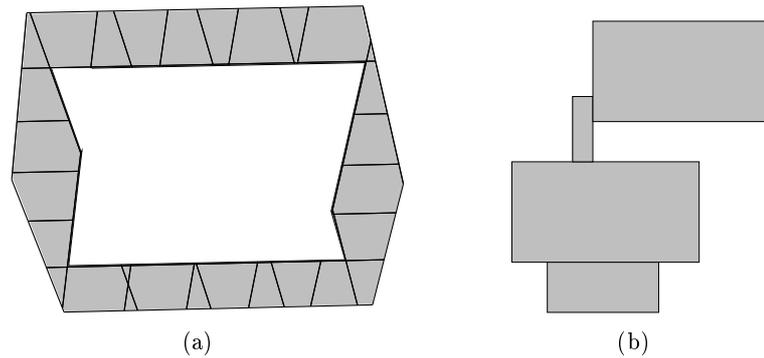


FIG. 1.7 – Exemples de 1-courbes irrégulières : (a) 1-courbe fermée dans un espace irrégulier. (b) 1-courbe ouverte dans une grille irrégulière isothétique.

1.3 Modèles de discrétisation

Un objet discret peut être obtenu de diverses manières, telles que la construction (assemblage de pavés tels que les pixels), l'acquisition (photographie numérique, Imagerie par Résonance Magnétique) ou la *discrétisation* d'un objet continu. Ce dernier processus s'effectue grâce à l'utilisation de méthodes ou modèles de discrétisation.

Parmi les modèles existants, nous en distinguons trois, largement utilisés, appelés modèles Naïfs [Rev95, And00], Supercouverture [COK95] et Standard [And03], que nous présentons dans ce chapitre. Chacun de ces modèles autorise la discrétisation de certaines classes de primitives euclidiennes, et par la même, d'une manière plus générale, la discrétisation d'objets euclidiens. Ils s'appliquent en particulier aux espaces discrets classiques, et sont définis en toute dimension. Le modèle Supercouverture a de plus été étendu aux espaces discrets irréguliers [Cœu05], comme nous le verrons par la suite.

Dans cette section, nous donnons tout d'abord les définitions analytiques des primitives discrètes élémentaires que sont les hyperplans discrets. Nous abordons ensuite le problème de la discrétisation de primitives euclidiennes, en présentant les modèles de discrétisation Naïf, Supercouverture et Standard. Nous illustrons nos propos en appliquant chacun de ces modèles à des droites ou plans euclidiens.

Pour chacun de ces modèles, nous nous limitons à la présentation de quelques notions auxquelles nous ferons référence dans la suite de ce mémoire. Nous ajoutons par ailleurs plusieurs références bibliographiques afin que le lecteur puisse s'y référer pour obtenir des détails supplémentaires.

1.3.1 Description analytique des hyperplans discrets

Dans l'espace discret, tout comme dans l'espace euclidien, de nombreux objets géométriques discrets élémentaires. Nous pouvons par exemple définir une droite ou un plan discret (objet infinis), ou encore un segment discret (objet fini). De plus, la description de ces objets peut être faite soit en utilisant une représentation dite *en extension*, par énumération des pavés les constituant, soit une représentation dite *en compréhension* ou *analytique*. Cette dernière consiste à décrire un objet discret à l'aide d'un ensemble d'inéquations qui doivent être vérifiées par les points discrets le constituant. Cette représentation présente l'avantage d'être indépendante du

nombre de points discrets de l'objet décrit, contrairement à la représentation en extension.

La définition analytique d'un *hyperplan analytique discret* a été introduite par J.-P. REVEILLÈS en 1991 [Rev91] dans le cadre des espaces discrets classiques. Elle permet de définir l'ensemble des points discrets d'un hyperplan à l'aide d'une double inéquation. Nous rappelons ici cette définition, ainsi que celle d'un *hyperplan discret fermé*.

Définition 8 (Hyperplan analytique discret [Rev91]) *L'hyperplan analytique discret de paramètres $A \in \mathbb{R}$, $B \in \mathbb{R}$ et $C \in \mathbb{R}^n$, est l'ensemble des points discrets $p = (p_1, \dots, p_n) \in \mathbb{Z}^n$ vérifiant*

$$A \leq \sum_{i=1}^n C_i p_i < B$$

Définition 9 (Hyperplan analytique discret fermé [And00]) *L'hyperplan analytique discret fermé de paramètres $A \in \mathbb{R}$, $B \in \mathbb{R}$ et $C \in \mathbb{R}^n$, est l'ensemble des points discrets $p = (p_1, \dots, p_n) \in \mathbb{Z}^n$ vérifiant*

$$A \leq \sum_{i=1}^n C_i p_i \leq B$$

Soit H un hyperplan analytique discret. L'épaisseur arithmétique ω de H est définie par $\omega = B - A$. En fonction de son épaisseur arithmétique, un hyperplan analytique peut être qualifié de diverses manières.

Définition 10 ([And00]) *Soit H un hyperplan analytique discret de paramètres $A \in \mathbb{R}$, $B \in \mathbb{R}$ et $C \in \mathbb{R}^n$. Alors*

- H est un **hyperplan Naïf** si $\omega = \max_{1 \leq i \leq n} |C_i|$;
- H est un **hyperplan Standard** si $\omega = \sum_{i=1}^n |C_i|$;
- H est dit **mince** si $\omega < \max_{1 \leq i \leq n} |C_i|$;
- H est dit **épais** si $\omega > \sum_{i=1}^n |C_i|$.

La figure 1.8 montre quatre exemples en dimension 2 de droites discrètes Naïve, Standard, mince et épaisse.

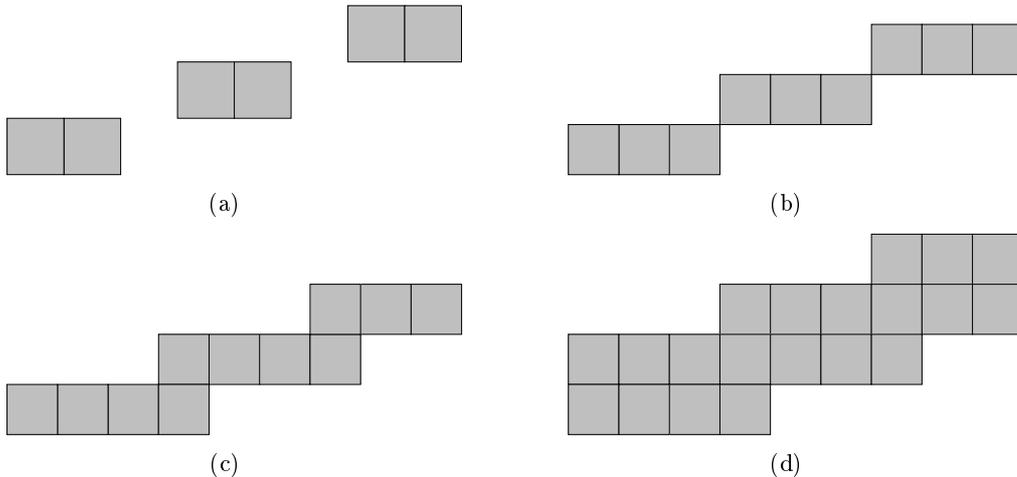


FIG. 1.8 – Segments de droites discrètes de paramètres $C = (1, -3)$ et d'épaisseur variable : (a) Une droite mince. (b) Une droite Naïve. (c) Une droite Standard. (d) Une droite épaisse.

1.3.2 Le modèle Naïf

Dans cette section, nous définissons les hyperplans Naïf et Naïf fermé. Nous donnons ensuite une caractérisation géométrique de la discrétisation Naïve d'un hyperplan euclidien.

Un hyperplan Naïf [Rev95, AAS97] est défini de la façon suivante :

Définition 11 (Hyperplan Naïf [Rev95]) *L'hyperplan Naïf de paramètres $(c_0, \dots, c_n) \in \mathbb{R}^{n+1}$ est l'ensemble des points $p = (p_1, \dots, p_n) \in \mathbb{Z}^n$ vérifiant*

$$-\frac{\max_{1 \leq i \leq n} |c_i|}{2} \leq c_0 + \sum_{i=1}^n c_i p_i < \frac{\max_{1 \leq i \leq n} |c_i|}{2}$$

Un exemple de droite Naïve en dimension 2 est montré sur la figure 1.9.

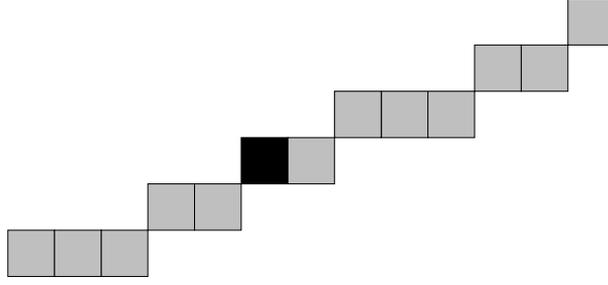


FIG. 1.9 – Droite Naïve de paramètres $(1, 3, -7)$. Le pixel coloré en noir est le pixel de centre $(0, 0)$.

Définition 12 (Hyperplan Naïf fermé [And00]) *L'hyperplan Naïf fermé de paramètres $(c_0, \dots, c_n) \in \mathbb{R}^{n+1}$ est l'ensemble des points $p = (p_1, \dots, p_n) \in \mathbb{Z}^n$ vérifiant*

$$-\frac{\max_{1 \leq i \leq n} |c_i|}{2} \leq c_0 + \sum_{i=1}^n c_i p_i \leq \frac{\max_{1 \leq i \leq n} |c_i|}{2}$$

Remarque 2 Soit $B_{d_1}(c, r)$ la boule fermée de centre $c = (c_1, \dots, c_n) \in \mathbb{Z}^n$ et de rayon $r \in \mathbb{R}$ définie par

$$B_{d_1}(c, r) = \left\{ p = (p_1, \dots, p_n) \in \mathbb{R}^n \mid \sum_{i=1}^n |p_i - c_i| \leq r \right\}$$

La discrétisation d'un hyperplan euclidien à l'aide du modèle Naïf fermé peut être obtenue géométriquement de la manière suivante : considérons l'espace discret pavé par des boules B_{d_1} de rayon $\frac{1}{2}$. Alors, la discrétisation Naïve fermée d'un hyperplan euclidien H est l'ensemble des points discrets dont la boule associée B_{d_1} est coupée par H . Il en est de même dans le cas du modèle Naïf, excepté lorsque l'hyperplan euclidien contient un sommet de boule. Dans ce cas, seuls certains points discrets associés à une boule contenant ce sommet sont conservés. La figure 1.10 montre un exemple de discrétisation à la fois Naïve et Naïve fermée.

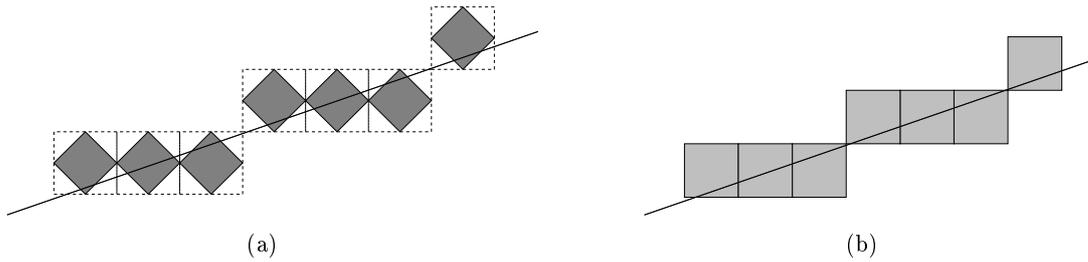


FIG. 1.10 – Exemple de discrétisation Naïve et Naïve fermée d’une droite euclidienne en dimension 2. (a) Pavage de l’espace par des losanges réguliers (boules B_{d_1} de rayon $\frac{1}{2}$). (b) Droite Naïve et Naïve fermée obtenue.

1.3.3 Le modèle Supercouverture

Dans cette section, nous nous intéressons au modèle Supercouverture [COK95]. Nous donnons tout d’abord la définition d’un hyperplan Supercouverture dans l’espace discret classique. Puis nous abordons le cas des espaces non classiques, en indiquant notamment de quelle manière le modèle Supercouverture a été étendu dans le cadre des grilles irrégulières isothétiques.

1.3.3.1 Supercouverture dans les espaces discrets classiques

Un hyperplan discret Supercouverture [ANF97a, ANF97b] est défini de la façon suivante :

Définition 13 (Hyperplan Supercouverture [And00]) *L’hyperplan Supercouverture de paramètres $(c_0, \dots, c_n) \in \mathbb{R}^{n+1}$ est l’ensemble des points $p = (p_1, \dots, p_n) \in \mathbb{Z}^n$ vérifiant*

$$-\frac{\sum_{i=1}^n |c_i|}{2} \leq c_0 + \sum_{i=1}^n c_i p_i \leq \frac{\sum_{i=1}^n |c_i|}{2}$$

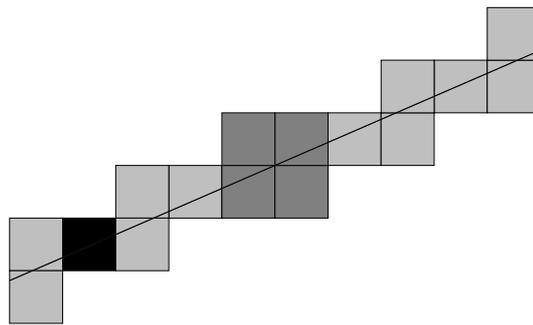


FIG. 1.11 – Droite Supercouverture de paramètres $(0, 3, -7)$. Le pixel coloré en noir est le pixel de centre $(0, 0)$. Nous pouvons remarquer la présence d’une 2-bulle (pixels gris foncés).

Remarque 3 *La discrétisation Supercouverture d’un hyperplan euclidien peut être, à l’instar de la discrétisation Naïve, caractérisée géométriquement. En effet, la discrétisation Supercouverture*

d'un hyperplan euclidien dans l'espace discret classique consiste simplement en l'ensemble des hypervoxels qui sont coupés par cet hyperplan. Un exemple de droite Supercouverture en dimension 2 est montré sur la figure 1.11.

De par sa définition, un hyperplan Supercouverture peut éventuellement présenter des particularités appelées k -bulles, $k \in \llbracket 1, n \rrbracket$, et ce dans le cas où l'hyperplan euclidien contient un sommet d'hypervoxel. Nous voyons sur la figure 1.11 un exemple de 2-bulle.

1.3.3.2 Supercouverture dans d'autres espaces discrets

Dans le cadre de grilles irrégulières isothétiques, D. CŒURJOLLY définit formellement dans [Cœu05] la discrétisation Supercouverture d'une droite euclidienne. De manière similaire à la définition de la discrétisation Supercouverture dans les espaces discrets classiques, la discrétisation Supercouverture d'une droite dans une grille irrégulière isothétique est l'ensemble des pavés de la grille coupés par la droite. Nous pouvons voir un exemple de discrétisation Supercouverture d'une droite dans une grille irrégulière isothétique sur la figure 1.12.

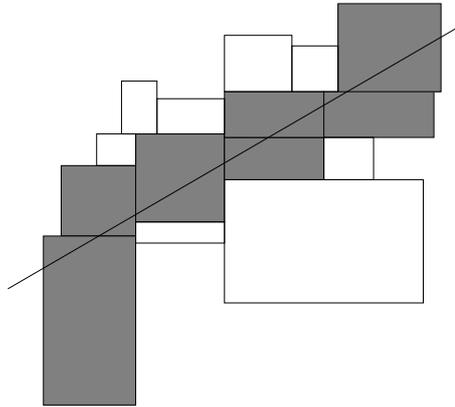


FIG. 1.12 – Discrétisation Supercouverture d'une droite euclidienne dans une grille irrégulière isothétique (figure inspirée de [Cœu05]).

Remarque 4 *Le modèle de discrétisation Supercouverture peut de manière similaire être appliqué à tout espace discret régulier ou non. Dans ce cas, la discrétisation Supercouverture d'un hyperplan euclidien sera décrite par l'ensemble des pavés de l'espace discret considéré qui sont coupés par l'hyperplan. Dans la suite, nous appellerons cette méthode de discrétisation la méthode de **discrétisation Supercouverture universelle**. De plus, la discrétisation d'un hyperplan euclidien à l'aide de cette méthode sera appelé un **hyperplan U-Supercouverture**.*

1.3.4 Le modèle Standard

Dans cette section, nous nous intéressons au modèle Standard [And03]. La discrétisation Standard d'une primitive euclidienne est très proche de sa discrétisation Supercouverture, à ceci près qu'elle ne présente pas de bulles. De plus, tout comme avec le modèle Supercouverture, il est possible de décrire analytiquement la discrétisation Standard des objets euclidiens linéaires.

Dans la suite, nous donnons tout d'abord les définitions analytiques des points et hyperplans Standard. Nous donnons ensuite la description d'une droite Standard en dimension 3. Enfin, nous donnons la description analytique d'un segment de droite discrète Standard.

1.3.4.1 Points et hyperplans Standard

Définition 14 (Point Standard [And03]) Soit p un point de coordonnées $(p_1, \dots, p_n) \in \mathbb{R}^n$. La discrétisation Standard de p est l'unique point discret de coordonnées $(x_1, \dots, x_n) \in \mathbb{Z}^n$ vérifiant

$$p_1 - \frac{1}{2} \leq x_1 < p_1 + \frac{1}{2}, p_2 - \frac{1}{2} \leq x_2 < p_2 + \frac{1}{2}, \dots, \text{ et } p_n - \frac{1}{2} \leq x_n < p_n + \frac{1}{2}$$

Un hyperplan Standard dans l'espace discret classique de dimension n est défini de la manière suivante :

Définition 15 (Hyperplan Standard [And03]) L'hyperplan Standard de paramètres $(c_0, \dots, c_n) \in \mathbb{R}^{n+1}$ est l'ensemble des points $p = (p_1, \dots, p_n) \in \mathbb{Z}^n$ vérifiant

$$-\frac{\sum_{i=1}^n |c_i|}{2} \leq c_0 + \sum_{i=1}^n c_i p_i < \frac{\sum_{i=1}^n |c_i|}{2}$$

où $c_1 \geq 0$, ou $c_1 = 0$ et $c_2 \geq 0$, ou \dots , ou $c_1 = c_2 = \dots = c_{n-1} = 0$ et $c_n \geq 0$.

Deux exemples de droite et de plan Standard sont montrés sur la figure 1.13.

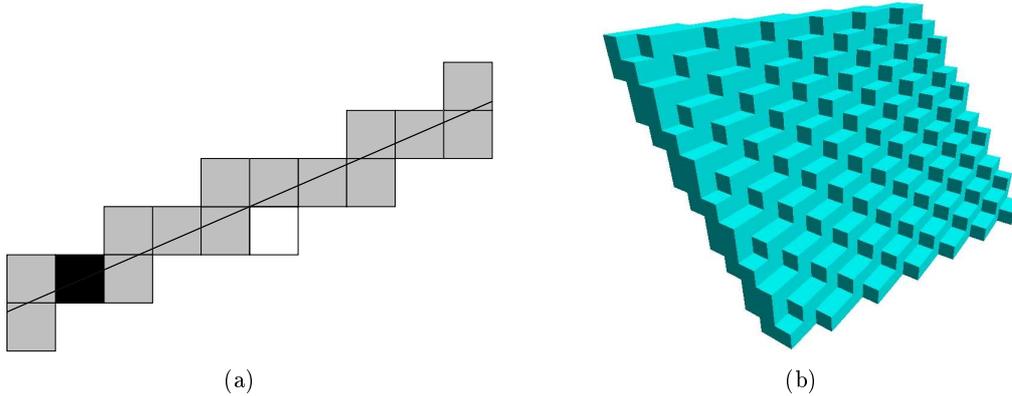


FIG. 1.13 – Hyperplans Standard en dimensions 2 et 3 : (a) Droite Standard de paramètres $(0, 3, -7)$. Le pixel coloré en noir est le pixel de centre $(0, 0)$. Notons que le pixel blanc n'appartient pas à la droite Standard. (b) Plan Standard de paramètres $(0, 3, -1, 2)$.

1.3.4.2 Droite Standard en dimension 3

Définissons à présent un droite Standard en dimension 3. De manière informelle, une *droite Standard en dimension 3* (voir Figure 1.14) est une droite discrète 2-connexe dont les trois projections orthogonales sont des droites Standard en dimension 2 (voir [And03] pour plus de détails).

Remarque 5 La discrétisation Standard d'un hyperplan euclidien, ainsi que celle d'une droite euclidienne en dimension 3, consiste en tous les hypervoxels qui sont coupés par l'hyperplan, excepté quand l'hyperplan passe par un sommet d'hypervoxel. Dans ce cas, certains hypervoxels ne font pas partie de la discrétisation Standard de l'hyperplan, et ce puisque une inégalité dans la définition 15 est stricte.

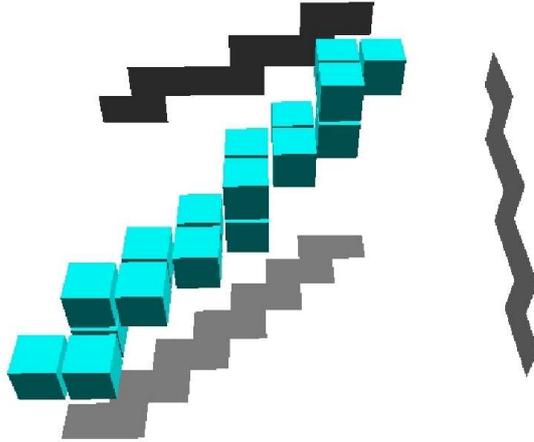


FIG. 1.14 – Exemple de droite Standard en dimension 3 et ses trois projections orthogonales.

1.3.4.3 Description analytique Standard d'un segment

A l'aide du modèle Standard, il est possible d'obtenir une description analytique de divers objets discrets [And00]. Nous donnons ici un exemple de discrétisation Standard d'un segment S en dimension 2 (voir Figure 1.15). Cette description nous sera en particulier utile dans le chapitre 4 de ce mémoire.

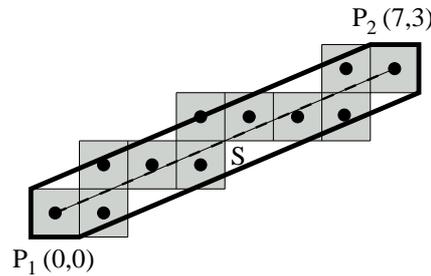


FIG. 1.15 – Exemple de segment Standard et sa description analytique (matérialisée par le polygone en traits continus gras). Les pixels en gris correspondent à la discrétisation Standard de S (trait pointillé).

La description analytique de la discrétisation Standard de S est déduite de celles de ses extrémités $P1$ et $P2$, ajoutées à celle de la droite contenant S . Ces inéquations sont les suivantes :

- Pour la droite : $-5 \leq 3x - 7y < 5$
- Pour $P1$: $-\frac{1}{2} \leq x < \frac{1}{2}$ et $-\frac{1}{2} \leq y < \frac{1}{2}$
- Pour $P2$: $7 - \frac{1}{2} \leq x < 7 + \frac{1}{2}$ et $3 - \frac{1}{2} \leq y < 3 + \frac{1}{2}$

La description analytique de la discrétisation Standard de S est alors composée des 6 inéquations suivantes (suite à la suppression de 4 inéquations) :

$$\left\{ \begin{array}{l} -5 \leq 3x - 7y < 5 \\ -\frac{1}{2} \leq x \\ -\frac{1}{2} \leq y \\ x < 7 + \frac{1}{2} \\ y < 3 + \frac{1}{2} \end{array} \right.$$

1.4 Conclusion

Dans ce chapitre, nous avons fait quelques rappels concernant le domaine de la géométrie discrète. Nous avons en particulier présenté diverses notions relatives à différents types d'espaces discrets, réguliers et irréguliers, dans lesquels nous serons amenés à évoluer dans la suite de ce mémoire.

Nous avons de plus rapidement présenté quelques modèles de discrétisation existants à la fois pour les espaces discrets classiques et pour les espaces irréguliers. Nous y ferons notamment référence dans les chapitres 2 et 3 de ce mémoire. Dans le chapitre 4, nous nous intéresserons exclusivement au modèle Standard, qui est le modèle avec lequel nous avons travaillé dans le cadre de la conception de notre logiciel de modélisation.

Chapitre 2

Une *préimage généralisée* pour la reconnaissance de primitives discrètes

Sommaire

2.1	Introduction	19
2.2	Représentation par dualité : un espace de paramètres	21
2.2.1	Représentation duale d'un point	21
2.2.1.1	Définitions	21
2.2.1.2	Propriétés	22
2.2.2	Objets duaux	23
2.2.2.1	Définitions et propriétés	23
2.2.2.2	Dual d'un polytope	25
2.2.3	Application à l'espace discret classique	29
2.2.3.1	Description du dual d'un hypervoxel	29
2.2.3.2	Exemples en dimensions 2 et 3	30
2.2.4	Cas d'une droite en dimension 3	31
2.3	La notion de préimage généralisée	33
2.3.1	Définition	33
2.3.2	Préimage généralisée d'hypervoxels	37
2.3.2.1	Étude de la préimage généralisée d'un ensemble de pixels 1-connexe	37
2.3.2.2	Cas d'un ensemble de voxels 2-connexe	38
2.4	Reconnaissance d'hyperplans discrets	39
2.4.1	Algorithme de reconnaissance	39
2.4.2	Exemples d'applications	39
2.4.2.1	Hyperplans Supercouverture	39
2.4.2.2	Hyperplans Standard	41
2.4.2.3	Modèles Naïfs	41
2.5	Conclusion	41

2.1 Introduction

Dans ce chapitre, nous nous intéressons à la reconnaissance de primitives discrètes. Ce problème est le suivant : étant donné un ensemble de points discrets, existe-t-il une primitive discrète

contenant cet ensemble ?

De nombreuses méthodes de reconnaissance de primitives discrètes ont été proposées depuis quelques dizaines d’années. En particulier, le problème de la reconnaissance d’hyperplans discrets [DRR95, Kim84, DR95] a été largement étudié, principalement en dimensions 2 et 3 (voir [KR04b, BCK07] pour un panorama des algorithmes de reconnaissance), avec plusieurs approches, telles que la programmation linéaire [Buz02, FST96, Buz03], des méthodes issues de la géométrie algorithmique [KS91, GDRZ05], ou des algorithmes basés sur le calcul d’une *préimage* [VC00, Siv04, VC99, Vit99]. Enfin, très peu d’articles traitent le problème en dimension quelconque [BD05, ST91]. Une étude de la complexité de certains de ces algorithmes est proposée dans [CB06].

Les diverses méthodes proposées peuvent se classer dans deux grandes catégories : les méthodes de décision simple, qui répondent uniquement à la question suivante : “les points discrets donnés appartiennent-ils à une primitive discrète ?”, et les méthodes fournissant en sus les paramètres des primitives reconnues. Elles sont de plus spécifiques à un modèle de discrétisation ou à un espace discret particulier. Dans ce chapitre, nous nous intéressons aux méthodes fournissant les paramètres des primitives reconnues.

Parmi ces méthodes, nous distinguons les méthodes basées sur le calcul d’une *préimage*. De manière simplifiée, cette *préimage* [DS84, McI85] est un objet géométrique défini dans un espace de paramètres dans lequel chaque point de la *préimage* est associé à un hyperplan de l’espace euclidien, et qui fournit l’ensemble des paramètres des primitives reconnues. Dans la suite, nous appellerons cette *préimage* la *préimage classique*. Quelques algorithmes de calcul de cette *préimage* ont été proposés en dimensions 2 et 3 [Cœu02, VC00, Vit99, VC99, Siv04], pour les modèles Naïf et Standard. Ils sont incrémentaux, indépendants de la connexité et de l’ordre d’ajout des points discrets.

L’inconvénient majeur de cette *préimage*, telle qu’elle est définie classiquement est que celle-ci peut être calculée pour des primitives situées dans une certaine portion de l’espace, typiquement un octant en dimension 2 et un 48ème d’espace en dimension 3. La reconnaissance de primitives non situées dans ces portions d’espace se fait donc moyennant une succession d’opérations géométriques simples, telles que des symétries. De plus, ceci induit un autre problème lorsque l’on souhaite déterminer et représenter dans un même espace les primitives solutions de plusieurs primitives discrètes.

Nous proposons ici un nouvel algorithme de reconnaissance de primitives discrètes. Cet algorithme permet la reconnaissance d’hyperplans discrets U-Supercouverture et est en particulier basé sur le calcul d’une *préimage* particulière, que nous appelons *préimage généralisée*. Cette *préimage* est, tout comme la *préimage classique*, calculée dans un espace de paramètres, et fournit à partir d’un ensemble de polytopes donnés l’ensemble des hyperplans coupant l’ensemble de polytopes. L’algorithme de reconnaissance est de plus incrémental, indépendant de l’ordre d’ajout et de la connexité des polytopes. Enfin, nous verrons que cet algorithme peut être aisément appliqué à la reconnaissance de diverses primitives discrètes, telles que les hyperplans Standard, Naïf ou Supercouverture, ainsi que dans le cadre des grilles isothétiques irrégulières.

Dans ce chapitre, nous présentons tout d’abord l’espace de paramètres dans lequel est définie notre *préimage généralisée*, et décrivons en particulier la représentation d’un polytope dans cet espace. Nous introduisons ensuite la notion de *préimage généralisée* d’un ensemble de polytopes. Nous en déduisons notre algorithme de reconnaissance d’hyperplans U-Supercouverture, et expliquons de quelle manière il peut être étendu à divers modèles de discrétisation.

2.2 Représentation par dualité : un espace de paramètres

Les *espaces de paramètres* sont souvent utilisés dans le domaine du traitement d'images, afin notamment de détecter des droites, ou plus généralement des formes paramétriques, dans une image. Ces espaces sont définis à partir de *transformations* associant un point P de l'espace image (assimilé ici à l'espace euclidien) à un objet géométrique O (droite, courbe, etc ...) dans l'espace de paramètres. Chaque point de cet objet est alors associé par le biais d'une deuxième transformation, à une primitive de l'espace image. Par exemple, dans le cadre de la détection de droites, tout point de O est associé à une droite euclidienne contenant le point P . Ainsi, les alignements de points dans une image sont détectés en recherchant des régions d'intersections entre plusieurs objets de l'espace de paramètres. Les transformations les plus connues sont les transformations (ou transformées) de Hough [Hou62, DH72]. Un aperçu de différentes transformées de Hough existantes est donné par H. MAÎTRE dans [Mai85].

Dans cette thèse, nous avons utilisé un espace de paramètres de dimension n , $\mathcal{P}_n = (O_{\mathcal{P}}, Y_1, \dots, Y_n) \subseteq \mathbb{R}^n$, et défini deux transformations nous permettant d'associer un point de l'espace euclidien à un hyperplan de l'espace de paramètres et vice et versa. Tout au long de la partie 2 de ce mémoire, nous désignerons par $\mathcal{E}_n = (O_{\mathcal{E}}, X_1, \dots, X_n)$ l'espace euclidien de dimension n . Dans la suite, les espaces \mathcal{P}_n et \mathcal{E}_n seront appelés *espace duaux*, chacun d'entre eux étant l'*espace dual* de l'autre.

Dans cette section, nous définissons tout d'abord les deux transformations qui nous permettent d'associer à tout point de \mathcal{E}_n (resp. \mathcal{P}_n) un hyperplan dans son espace dual. Nous définissons ensuite ce qu'est le *dual d'un objet* de \mathcal{E}_n ou \mathcal{P}_n , et décrivons en particulier le dual d'un polytope. Nous appliquons ensuite cette définition aux espaces discrets classiques en décrivant plus précisément le dual d'un hypervoxel. Enfin, nous nous intéressons au cas particulier des droites en dimension 3 en décrivant, dans l'espace dual, l'objet auquel est associé l'ensemble des plans contenant une droite donnée. Nous verrons que cet objet est aussi une droite.

2.2.1 Représentation duale d'un point

Dans cette section, nous définissons les relations entre points et hyperplans que nous avons utilisées, en donnant de plus quelques propriétés qui leur sont liées.

2.2.1.1 Définitions

Dans ce travail, nous utilisons deux fonctions associant un point de \mathcal{E}_n à un hyperplan de \mathcal{P}_n et inversement. Ces deux fonctions sont notées $D_{\mathcal{E}} : \mathcal{E}_n \rightarrow \wp(\mathcal{P}_n)$ et $D_{\mathcal{P}} : \mathcal{P}_n \rightarrow \wp(\mathcal{E}_n)$, où $\wp(\mathcal{E})$ désigne l'ensemble des parties d'un ensemble \mathcal{E} . Nous les définissons de la manière suivante :

Définition 16 Soient $p_{\mathcal{E}} = (x_1, \dots, x_n) \in \mathcal{E}_n$ et $p_{\mathcal{P}} = (y_1, \dots, y_n) \in \mathcal{P}_n$. Alors,

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

et

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

Notations : Dans la suite, nous utiliserons la notation $Dual$ à la place des notations $D_{\mathcal{E}}$ et $D_{\mathcal{P}}$, lorsque les deux fonctions peuvent être indifféremment utilisées.

Remarque 6 *Plusieurs autres transformations associant un point de \mathcal{E}_n (resp. \mathcal{P}_n) à un hyperplan dans l'espace dual, comme par exemple, celle qui associe un point $p = (p_1, \dots, p_n)$ à l'hyperplan d'équation $\sum_{i=1}^n c_i X_i = 0$ (resp. $\sum_{i=1}^n c_i Y_i = 0$) auraient pu être choisies.*

Pas les droites verticales (contrairement à Veelaert) ! Notre choix s'est tout de même porté sur les transformations de la définition 16 car l'espace de paramètres associé a déjà fait l'objet de nombreuses études, principalement en dimension 2, et ce notamment par P. VEELAERT [Vee99]. De plus, la préimage classique [DS84] que nous avons évoquée dans l'introduction de cette partie, et à laquelle nous avons comparé nos travaux (voir Chapitre 2) est définie dans un espace de paramètres similaire.

2.2.1.2 Propriétés

Une conséquence directe de la définition 16 est la suivante :

Propriété 1 *Soient p un point de \mathcal{E}_n ou \mathcal{P}_n , et p' un point de $Dual(p)$. Alors, $Dual(p')$ est un hyperplan contenant p .*

Preuve : Soit $p = (p_1, \dots, p_n)$ un point de \mathcal{E}_n (le cas où p est un point de \mathcal{P}_n se démontre de manière similaire). Soit $p' = (p'_1, \dots, p'_n)$ un point de $Dual(p)$. Puisque $p' \in Dual(p) = \mathcal{D}_{\mathcal{E}}(p)$, nous avons $p'_n = p_n - \sum_{i=1}^{n-1} p_i p'_i$. D'où $p_n = p'_n + \sum_{i=1}^{n-1} p_i p'_i$. Nous déduisons alors de cette dernière équation que p appartient à $Dual(p')$. \square

Dans la suite, nous appellerons *objet* d'un espace $\mathcal{E} \subseteq \mathbb{R}^n$ tout sous-ensemble de points de cet espace.

Définition 17 (Dual d'un objet) *Soit O un objet de \mathcal{E}_n (resp. \mathcal{P}_n). Le dual de O , noté $Dual(O)$, est un objet de l'espace de paramètres \mathcal{P}_n (resp. \mathcal{E}_n) défini par*

$$Dual(O) = \bigcup_{p \in O} Dual(p)$$

Remarque 7 *Nous pouvons déduire de la propriété 1 que chaque point du dual d'un objet O est associé à un hyperplan contenant au moins un point de O . Il s'agit là d'une propriété géométrique très importante sur laquelle nous nous appuyerons souvent dans la suite de l'exposé de nos travaux.*

Proposition 1 *Soient O_1 et O_2 deux objets de \mathcal{E}_n ou \mathcal{P}_n vérifiant $O_1 \subseteq O_2$. Alors,*

$$Dual(O_1) \subseteq Dual(O_2)$$

Preuve : Puisque $O_1 \subseteq O_2$, nous avons $Dual(O_2) = \bigcup_{p \in O_2} Dual(p) = \left[\bigcup_{p \in O_1} Dual(p) \right] \cup \left[\bigcup_{p \in O_2 \setminus O_1} Dual(p) \right]$. D'où $Dual(O_1) \subseteq Dual(O_2)$. \square

Proposition 2 Soient O_1 et O_2 deux objets de \mathcal{E}_n ou \mathcal{P}_n . Alors,

$$Dual(O_1 \cup O_2) = Dual(O_1) \cup Dual(O_2).$$

Preuve : $Dual(O_1 \cup O_2) = \bigcup_{p \in O_1 \cup O_2} Dual(p) = \left[\bigcup_{p \in O_1} Dual(p) \right] \cup \left[\bigcup_{p \in O_2} Dual(p) \right] = Dual(O_1) \cup Dual(O_2).$ \square

Proposition 3 Soient O_1 et O_2 deux objets de \mathcal{E}_n ou \mathcal{P}_n . Alors,

$$Dual(O_1 \cap O_2) \subseteq Dual(O_1) \cap Dual(O_2).$$

Preuve : Puisque $O_1 \cap O_2 \subseteq O_1$ et $O_1 \cap O_2 \subseteq O_2$, nous en déduisons que $Dual(O_1 \cap O_2) \subseteq Dual(O_1)$ et $Dual(O_1 \cap O_2) \subseteq Dual(O_2)$. D'où, $Dual(O_1 \cap O_2) \subseteq Dual(O_1) \cap Dual(O_2)$. \square

2.2.2 Objets duaux

Dans ce travail, nous nous sommes fortement appuyés sur la description du *dual d'un polytope*. Nous donnons dans cette section une description géométrique du dual d'un polytope en dimension n . Cette définition est en particulier uniquement basée sur les coordonnées des sommets du polytope. Mais avant cela, nous introduisons les notions d'extrusions positive et négative d'un point.

2.2.2.1 Définitions et propriétés

Nous définissons ici les *extrusions positive et négative* d'un point de \mathbb{R}^n , et donnons quelques propriétés relatives à ces définitions.

Définition 18 (Extrusions positive et négative d'un point) Soit $p = (x_1, \dots, x_n)$ un point de \mathbb{R}^n . L'*extrusion positive* de p dans \mathbb{R}^n est définie par

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

De la même manière, l'*extrusion négative* de p dans \mathbb{R}^n est définie par

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

Définition 19 (Extrusions positive et négative d'un objet) Soit O un objet de \mathbb{R}^n . L'*extrusion positive* de O , notée O^+ , est définie par

$$O^+ = \bigcup_{p \in O} p^+$$

De même, l'*extrusion négative* de O , notée O^- , est définie par

$$O^- = \bigcup_{p \in O} p^-$$

Les propriétés suivantes se déduisent directement de la définition 18 :

Propriété 2 Soient O_1 et O_2 deux objets de \mathbb{R}^n vérifiant $O_1 \subseteq O_2$. Alors,

$$O_1^+ \subseteq O_2^+$$

De même, $O_1^- \subseteq O_2^-$.

Preuve : La preuve de cette propriété est immédiate. □

Proposition 4 Soient O_1 et O_2 deux objets de \mathbb{R}^n . Alors,

$$(O_1 \cup O_2)^+ = O_1^+ \cup O_2^+$$

De même, $(O_1 \cup O_2)^- = O_1^- \cup O_2^-$.

Preuve : $(O_1 \cup O_2)^+ = \bigcup_{p \in O_1 \cup O_2} p^+ = \left[\bigcup_{p \in O_1} p^+ \right] \cup \left[\bigcup_{p \in O_2} p^+ \right] = O_1^+ \cup O_2^+$. La preuve de $(O_1 \cup O_2)^- = O_1^- \cup O_2^-$ s'obtient de manière similaire. □

Proposition 5 Soient O_1 et O_2 deux objets de \mathbb{R}^n . Alors,

$$(O_1 \cap O_2)^+ \subseteq O_1^+ \cap O_2^+$$

De même, $(O_1 \cap O_2)^- \subseteq O_1^- \cap O_2^-$.

Preuve : Nous savons que $O_1 \cap O_2 \subseteq O_1$ et $O_1 \cap O_2 \subseteq O_2$. Ainsi, puisque $(O_1 \cap O_2)^+ \subseteq O_1^+$ et $(O_1 \cap O_2)^+ \subseteq O_2^+$, nous en déduisons que $(O_1 \cap O_2)^+ \subseteq O_1^+ \cap O_2^+$. La seconde inclusion s'obtient de la même manière. □

Enfin, nous avons la relation suivante :

Proposition 6 Soit p un point de \mathbb{R}^n . Alors,

$$Dual(p)^+ = Dual(p^+)$$

De même, $Dual(p)^- = Dual(p^-)$.

Preuve : Nous nous plaçons ici dans l'espace \mathcal{E}_n . La preuve pour l'espace \mathcal{P}_n est identique. Soit donc $p = (x_1, \dots, x_n) \in \mathcal{E}_n$. Alors, $Dual(p^+) = \mathcal{D}_{\mathcal{E}}(p^+) =$

$$\begin{aligned} \bigcup_{p' \in p^+} Dual(p') &= \bigcup_{p'=(x'_1, \dots, x'_n) \in p^+} \left\{ (y_1, \dots, y_n) \in \mathcal{P}_n \mid y_n = x'_n - \sum_{i=1}^{n-1} x'_i y_i \right\} = \\ &= \left\{ (y_1, \dots, y_n) \in \mathcal{P}_n \mid y_n \geq x_n - \sum_{i=1}^{n-1} x_i y_i \right\} = \bigcup_{p' \in \mathcal{D}_{\mathcal{E}}(p)} p'^+ = \mathcal{D}_{\mathcal{E}}(p)^+ = Dual(p)^+. \end{aligned}$$

La preuve de $Dual(p)^- = Dual(p^-)$ peut être obtenue de la même manière. □

La proposition 6 est illustrée sur la figure 2.1. Nous pouvons observer que le dual d'une demi-droite verticale (extrusion positive ou négative d'un point p) est un demi-espace (extrusion positive ou négative de la droite duale de p).

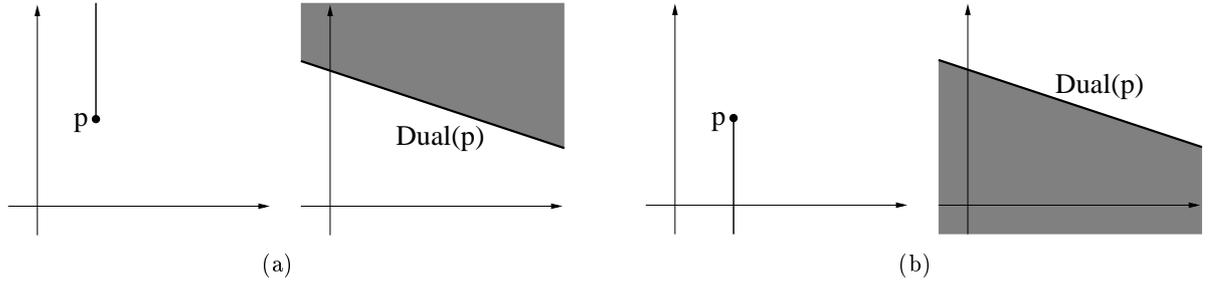


FIG. 2.1 – Extrusions positive et négative d'un point p (demi-droites) et leur objet dual : un demi-espace, (a) Extrusion positive de p , (b) Extrusion négative de p .

2.2.2.2 Dual d'un polytope

Nous définissons dans cette section le dual d'un polytope de \mathcal{E}^n . Le cas d'un polytope de \mathcal{P}^n peut être traité de manière similaire.

Définition 20 (n -polytope) Soit P_c un polytope de dimension n ou n -polytope. Nous supposons qu'il existe un ensemble fini de k demi-espaces $\overline{\mathcal{H}} = \{\overline{H}_1, \dots, \overline{H}_k\}$ tel que $P_c = \bigcap_{i=1}^k \overline{H}_i$, et tel que si H_i est l'hyperplan formant le bord du demi-espace \overline{H}_i (ou **hyperplan porteur** de \overline{H}_i), alors $\forall i \in \llbracket 1, k \rrbracket, H_i \cap P_c \neq \emptyset$.

Notations : Soient P_c un n -polytope, et $\overline{\mathcal{H}}$ l'ensemble de demi-espaces correspondant. Nous définissons trois sous-ensembles de demi-espaces de $\overline{\mathcal{H}}$, notés $\overline{\mathcal{H}}_0, \overline{\mathcal{H}}_+$ et $\overline{\mathcal{H}}_-$, de la façon suivante (+ schémas en 2D et 3D) :

- $\overline{\mathcal{H}}_0$ est l'ensemble des demi-espaces de $\overline{\mathcal{H}}$ définis par une inéquation du type $c_n + \sum_{i=1}^{n-1} c_i X_i \geq 0$ ou du type $c_n + \sum_{i=1}^{n-1} c_i X_i \leq 0$, avec $(c_1, \dots, c_n) \in \mathcal{E}^n$.
- $\overline{\mathcal{H}}_+$ est l'ensemble des demi-espaces de $\overline{\mathcal{H}}$ définis par une inéquation du type $X_n \geq c_n + \sum_{i=1}^{n-1} c_i X_i$, $(c_1, \dots, c_n) \in \mathcal{E}^n$.
- $\overline{\mathcal{H}}_-$ est l'ensemble des demi-espaces de $\overline{\mathcal{H}}$ définis par une inéquation du type $X_n \leq c_n + \sum_{i=1}^{n-1} c_i X_i$, $(c_1, \dots, c_n) \in \mathcal{E}^n$.

De plus, nous notons $\mathcal{H}_0, \mathcal{H}_+$ et \mathcal{H}_- les trois ensembles d'hyperplans porteurs correspondant respectivement aux ensembles de demi-espaces $\overline{\mathcal{H}}_0, \overline{\mathcal{H}}_+$ et $\overline{\mathcal{H}}_-$.

Vérifier l'utilité des \mathcal{H}_0 !

Proposition 7 Soit P_c un n -polytope. Alors,

$$P_c = P_c^+ \cap P_c^-$$

avec

$$P_c^+ = \bigcap_{\overline{H} \in (\overline{\mathcal{H}}_0 \cup \overline{\mathcal{H}}_+)} \overline{H}$$

et

$$P_c^- = \bigcap_{\overline{H} \in (\overline{\mathcal{H}}_0 \cup \overline{\mathcal{H}}_-)} \overline{H}$$

Preuve : Montrons que $P_c^+ = \bigcap_{\overline{H} \in (\overline{\mathcal{H}}_0 \cup \overline{\mathcal{H}}_+)} \overline{H}$. L'égalité $P_c^- = \bigcap_{\overline{H} \in (\overline{\mathcal{H}}_0 \cup \overline{\mathcal{H}}_-)} \overline{H}$ s'obtient de manière similaire.

Soit $p = (p_1, \dots, p_n) \in P_c^+$. Alors, il existe $p' = (p'_1, \dots, p'_n) \in P_c$ tels que pour tout $i \in \llbracket 1, n-1 \rrbracket$,

$$c_i = c'_i \text{ et } c_n = c'_n$$

Ainsi, pour tout $\overline{H} \in \overline{\mathcal{H}}_0$ et pour tout $\overline{H} \in \overline{\mathcal{H}}_+$, nous avons $p \in \overline{H}$. Nous en déduisons que $p \in \bigcap_{\overline{H} \in (\overline{\mathcal{H}}_0 \cup \overline{\mathcal{H}}_+)} \overline{H}$.

Soit à présent $p = (p_1, \dots, p_n) \in \bigcap_{\overline{H} \in (\overline{\mathcal{H}}_0 \cup \overline{\mathcal{H}}_-)} \overline{H}$. Supposons que $p \notin P_c^+$. Alors, pour tout $p' = (p'_1, \dots, p'_n) \in P_c$, il existe $i \in \llbracket 1, n-1 \rrbracket$ tel que $c_i \neq c'_i$ ou $c_n \neq c'_n$. Ainsi, il existe $\overline{H} \in \overline{\mathcal{H}}_0$ ou $\overline{H} \in \overline{\mathcal{H}}_+$ tel que $p \notin \overline{H}$. Nous en déduisons que $p \notin \bigcap_{\overline{H} \in \overline{\mathcal{H}}_0 \cup \overline{\mathcal{H}}_-} \overline{H}$, ce qui contredit notre hypothèse. \square

La figure 2.2 illustre la proposition 7 dans le cas de la dimension 2.

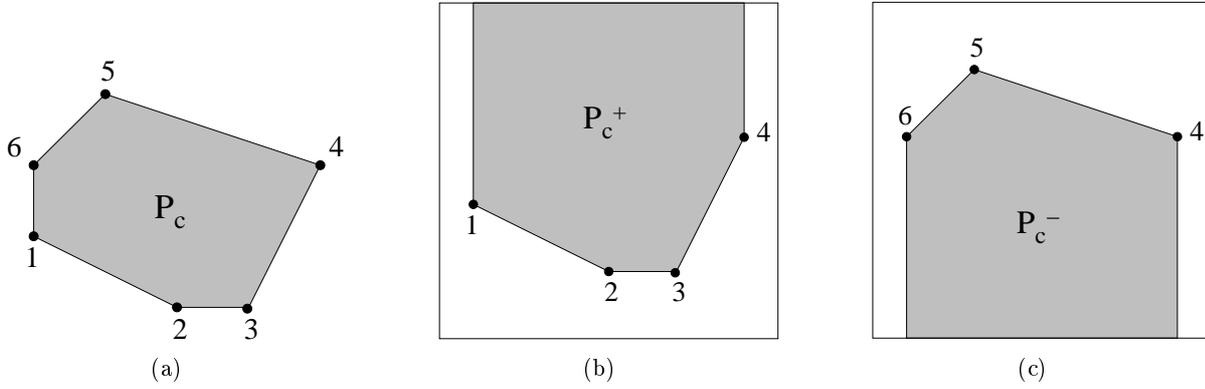


FIG. 2.2 – Extrusions positive et négative d'un polytope de dimension 2 : (a) Un 2-polytope P_c , (b) Extrusion positive de P_c , (c) Extrusion négative de P_c .

Décrivons à présent le dual d'un polytope P_c à partir de ses sommets.

Notations : Soit \mathcal{S} l'ensemble des sommets de P_c . Nous définissons deux sous-ensembles de \mathcal{S} , \mathcal{S}_+ et \mathcal{S}_- , de la façon suivante :

$$\mathcal{S}_+ = \{s \in \mathcal{S} \mid \exists H \in \mathcal{H}_+, s \in H \cap P_c\}$$

et

$$\mathcal{S}_- = \{s \in \mathcal{S} \mid \exists H \in \mathcal{H}_-, s \in H \cap P_c\}$$

Nous pouvons voir sur l'exemple de la figure 2.2 que les sommets numérotés 1, 2, 3 et 4 appartiennent à l'ensemble de sommets \mathcal{S}_+ du polygone P_c . De même, les sommets numérotés 4, 5 et 6 appartiennent à l'ensemble de sommets \mathcal{S}_- .

Le dual d'un polytope peut alors être défini de la façon suivante :

Théorème 1 (Dual d'un polytope) *Soit P_c un n -polytope. Alors,*

$$Dual(P_c) = \left[\bigcup_{s \in \mathcal{S}_+} Dual(s)^+ \right] \cap \left[\bigcup_{s \in \mathcal{S}_-} Dual(s)^- \right]$$

Preuve : Démontrons tout d'abord le lemme suivant :

Lemme 1 *Soit P_c un n -polytope. Alors,*

$$Dual(P_c) = Dual(P_c)^+ \cap Dual(P_c)^-$$

Preuve : Nous savons que $Dual(P_c) \subseteq Dual(P_c)^+$ et $Dual(P_c) \subseteq Dual(P_c)^-$. D'où $Dual(P_c) \subseteq Dual(P_c)^+ \cap Dual(P_c)^-$.

Montrons l'inclusion inverse. Soit $p = (p_1, \dots, p_n) \in Dual(P_c)^+ \cap Dual(P_c)^-$. Alors,

$$\exists p' = (p'_1, \dots, p'_n) \in Dual(P_c) | p \in p'^+$$

De même,

$$\exists p'' = (p''_1, \dots, p''_n) \in Dual(P_c) | p \in p''^-$$

Nous en déduisons que $\forall i \in \llbracket 1, n-1 \rrbracket, p_i = p'_i = p''_i$, et $p'_n \leq p_n \leq p''_n$. Le point p appartient donc au segment vertical $[p', p'']$. Or, le dual de tout point de $[p', p'']$ est par définition un hyperplan parallèle à $Dual(p')$ et $Dual(p'')$, d'équation $X_n = c - \sum_{i=1}^{n-1} p_i X_i$, avec $p'_n \leq c \leq p''_n$. Considérons deux points $p_1 \in P_c \cap Dual(p')$ et $p_2 \in P_c \cap Dual(p'')$. Alors, puisque P_c est convexe, nous en déduisons que pour tout hyperplan H appartenant à $Dual([p', p''])$, nous avons $H \cap [p_1, p_2] \neq \emptyset$. D'où, $Dual(p) \cap P_c \neq \emptyset$, et donc $p \in Dual(P_c)$. \square

Soient à présent \mathcal{C}_+ et \mathcal{C}_- deux sous-ensembles d'objets définis par

$$\mathcal{C}_+ = \{H \cap P_c, H \in \mathcal{H}_+\}$$

et

$$\mathcal{C}_- = \{H \cap P_c, H \in \mathcal{H}_-\}$$

Notons que le cardinal de \mathcal{C}_+ (resp. \mathcal{C}_-), noté $|\mathcal{C}_+|$ (resp. $|\mathcal{C}_-|$), est égal au cardinal de \mathcal{H}_+ (resp. \mathcal{H}_-).

En dimension 2, l'ensemble \mathcal{C}_+ (resp. \mathcal{C}_-) correspond aux segments appartenant au bord de P_c et ayant pour extrémités deux sommets de \mathcal{S}_+ (resp. \mathcal{S}_-). Sur la figure 2.2, \mathcal{C}_+ est donc constitué des segments $[1, 2]$, $[2, 3]$ et $[3, 4]$. De même, \mathcal{C}_- est constitué des segments $[4, 5]$ et $[5, 6]$. En dimension 3, ces deux ensembles sont composés de facettes du bord du polyèdre P_c .

La relation suivante est alors vérifiée :

Lemme 2 *Soit P_c un n -polytope. Alors,*

$$P_c^+ = \bigcup_{C \in \mathcal{C}_+} C^+$$

De même, $P_c^- = \bigcup_{C \in \mathcal{C}_-} C^-$.

Preuve : Nous devons montrer que

$$P_c^+ = \bigcup_{C \in \mathcal{C}_+} C^+ = \bigcup_{i \in \llbracket 1, |\mathcal{C}_+ \rrbracket, H_i \in \mathcal{H}_+} (H_i \cap P_c)^+ = \left[\bigcup_{i \in \llbracket 1, |\mathcal{C}_+ \rrbracket, H_i \in \mathcal{H}_+} H_i \cap P_c \right]^+$$

Tout d'abord, nous avons

$$\bigcup_{i \in \llbracket 1, |\mathcal{C}_+ \rrbracket, H_i \in \mathcal{H}_+} H_i \cap P_c \subseteq P_c$$

D'où,

$$\left[\bigcup_{i \in \llbracket 1, |\mathcal{C}_+ \rrbracket, H_i \in \mathcal{H}_+} H_i \cap P_c \right]^+ \subseteq P_c^+$$

Soit à présent $p \in P_c^+$. Nous savons que $P_c^+ = \bigcap_{\overline{H} \in \overline{\mathcal{H}}_0 \cup \overline{\mathcal{H}}_+} \overline{H}$, ce qui est équivalent à $P_c^+ = \bigcap_{H \in \mathcal{H}_0 \cup \mathcal{H}_+} H^+$. Ainsi, nous en déduisons que pour tout $H_i \in \mathcal{H}_+$, $i \in \llbracket 1, |\mathcal{C}_+ \rrbracket$, il existe $p_i = (p_{i_1}, \dots, p_{i_n}) \in H_i$ tel que $p \in p_i^+$. Soit $p' = (p_{i_1}, \dots, p_{i_{n-1}}, p'_n)$ le point vérifiant $\forall i \in \llbracket 1, |\mathcal{C}_+ \rrbracket, p'_n \geq p_{i_n}$. Alors, par définition de P_c , on a $p' \in P_c$. D'où le résultat. \square

La seconde égalité s'obtient de manière identique. \square

Lemme 3 Soient P_c un n -polytope. Alors,

$$Dual(P_c^+) = \bigcup_{s \in \mathcal{S}_+} Dual(s)^+$$

De même, $Dual(P_c^-) = \bigcup_{s \in \mathcal{S}_-} Dual(s)^-$.

Preuve : Démontrons que $Dual(P_c^+) \subseteq \bigcup_{s \in \mathcal{S}_+} Dual(s)^+$.

Par définition, pour tout sommet s de \mathcal{S}_+ , il existe $C \in \mathcal{C}_+$ tel que $s \in C$. Ainsi, $\bigcup_{s \in \mathcal{S}_+} s \subseteq \bigcup_{C \in \mathcal{C}_+} C$. De plus, $(\bigcup_{s \in \mathcal{S}_+} s)^+ \subseteq (\bigcup_{C \in \mathcal{C}_+} C)^+$. D'où, $\bigcup_{s \in \mathcal{S}_+} s^+ \subseteq \bigcup_{C \in \mathcal{C}_+} C^+$. Or, d'après le lemme 2, $\bigcup_{C \in \mathcal{C}_+} C^+ = P_c^+$. Nous en déduisons que $Dual(\bigcup_{s \in \mathcal{S}_+} s^+) \subseteq Dual(P_c^+)$, et donc que $\bigcup_{s \in \mathcal{S}_+} Dual(s^+) \subseteq Dual(P_c^+)$.

Montrons à présent l'inclusion inverse. Soit $p \in Dual(P_c^+) = Dual(\bigcup_{C \in \mathcal{C}_+} C^+)$. Alors, il existe $C \in \mathcal{C}_+$ tel que $Dual(p) \cap C^+ \neq \emptyset$. Montrons alors qu'il existe un sommet s de C tel que $Dual(p) \cap s^+ \neq \emptyset$. Pour cela, raisonnons par l'absurde, et supposons que pour tout $s \in C$, $Dual(p) \cap s^+ = \emptyset$. Nous savons qu'il existe $H \in \mathcal{H}_+$ tel que $C = H \cap P_c = H \cap \left[\bigcap_{i=1}^k \overline{H}_i \right] = \bigcap_{i=1}^k (H \cap \overline{H}_i)$. Ainsi, si nous considérons l'hyperplan H en tant qu'espace, nous en déduisons que C est un polytope, puisque pour tout i , $\overline{H}_i \cap H$ est un demi-espace dans H , et que donc C est une intersection de demi-espaces. Puisque C est l'enveloppe convexe de ses sommets, nous en déduisons que si $\forall s \in C, Dual(p) \cap s^+ = \emptyset$, alors, $\forall p' \in C, Dual(p) \cap p' = \emptyset$. De plus, on a $C^+ = \bigcup_{p \in C} p^+ = \bigcup_{p \in C, l \in \mathbb{R}_+} p + l\overrightarrow{OX_n} = \bigcup_{l \in \mathbb{R}_+} C + l\overrightarrow{OX_n}$. Ainsi, puisque C est un polytope, $C + l\overrightarrow{OX_n}$ est aussi un polytope et le même raisonnement peut alors être appliqué pour montrer que $\forall p' \in C + l\overrightarrow{OX_n}, Dual(p) \cap p' = \emptyset$. Ce qui contredit notre première hypothèse. Une preuve similaire peut être appliquée pour démontrer que $Dual(P_c^-) = \bigcup_{s \in \mathcal{S}_-} Dual(s)^-$. \square

La preuve du théorème 1 se déduit directement du lemme 3. \square

Le théorème 1 nous fournit une description du dual d'un polytope à partir du dual de ses sommets. Plus précisément, le dual d'un polytope est défini par l'intersection de deux objets, chaque objet étant une union de demi-espaces (voir Figure 2.3). Chacun de ces demi-espaces correspond à l'extrusion positive ou négative de l'hyperplan dual d'un des sommets du polytope. Sur la figure 2.3c, nous pouvons voir la représentation du dual du polygone convexe présent sur la figure 2.2a.

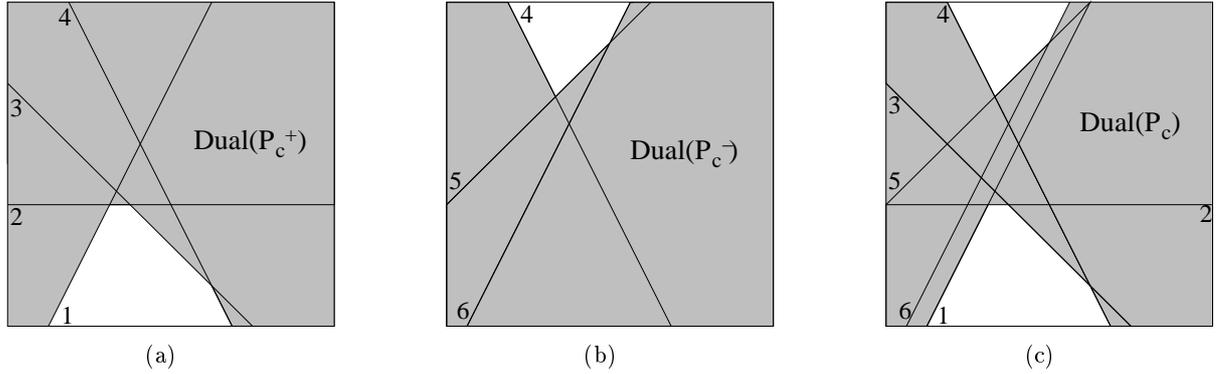


FIG. 2.3 – Dual d'un polygone convexe P_c : (a) Dual de l'extrusion positive de P_c , (b) Dual de l'extrusion négative de P_c , (c) Dual de P_c .

2.2.3 Application à l'espace discret classique

Dans le cadre des espaces discrets classiques, nous nous sommes intéressés à la description du dual des pavés particuliers que sont les hypervoxels. Dans ce qui suit, nous allons voir que le dual d'un hypervoxel peut aussi être décrit comme l'union de plusieurs *polytopes infinis*⁶.

2.2.3.1 Description du dual d'un hypervoxel

Nous décrivons ici le dual d'un hypervoxel à partir d'une union de polytopes. Mais avant cela, nous introduisons quelques notations.

Notations : Soit $\mathcal{L} = \{-1, 1\}$ et soit $\mathcal{R} : \mathcal{L} \rightarrow \{\mathbb{R}_-, \mathbb{R}_+\}$ la fonction définie par $\mathcal{R}(-1) = \mathbb{R}_-$ et $\mathcal{R}(1) = \mathbb{R}_+$. De plus, soit $l = (l_1, \dots, l_{n-1}) \in \mathcal{L}^{n-1}$. Nous notons \mathcal{R}^l le produit cartésien $\mathcal{R}(l_1) \times \dots \times \mathcal{R}(l_{n-1})$. Une partition de \mathbb{R}^n peut donc être définie par

$$\mathbb{R}^n = \bigcup_{l \in \mathcal{L}^{n-1}} (\mathcal{R}^l \times \mathbb{R})$$

Soit H un hypervoxel de centre $(c_1, \dots, c_n) \in \mathbb{Z}^n$. Nous notons s_-^l le sommet de H de coordonnées $(c_1 + \frac{1}{2}l_1, \dots, c_{n-1} + \frac{1}{2}l_{n-1}, c_n - \frac{1}{2})$ et par s_+^{-l} le sommet de coordonnées $(c_1 - \frac{1}{2}l_1, \dots, c_{n-1} - \frac{1}{2}l_{n-1}, c_n + \frac{1}{2})$.

Avec les notations précédentes, nous pouvons déduire du théorème 1 la description du dual d'un hypervoxel suivante :

⁶Nous appelons *polytope infini* un polytope possédant au moins un sommet situé à l'infini. Dans le cas contraire, nous parlerons de *polytope fini*.

Corollaire 1 (Dual d'un hypervoxel) *Soit H un hypervoxel de dimension n . Alors,*

$$Dual(H) = \bigcup_{l \in \mathcal{L}^{n-1}} \left[(\mathcal{R}^l \times \mathbb{R}) \cap Dual(s_-^l)^+ \cap Dual(s_+^l)^- \right]$$

Preuve : Soit H un hypervoxel de \mathcal{E}_n . Nous savons que

$\forall l \in \mathcal{L}^{n-1}, [(\mathcal{R}^l \times \mathbb{R}) \cap Dual(s_-^l)^+ \cap Dual(s_+^l)^-] \subset Dual(s_-^l)^+ \cap Dual(s_+^l)^-$. Ainsi, puisque $Dual(s_-^l)^+ \cap Dual(s_+^l)^- \subset Dual(\mathcal{S}_-)^+ \cap Dual(\mathcal{S}_+)^-$, nous déduisons du théorème 1 que $\forall l \in \mathcal{L}^{n-1}, [(\mathcal{R}^l \times \mathbb{R}) \cap Dual(s_-^l)^+ \cap Dual(s_+^l)^-] \subset Dual(H)$.

Prouvons à présent l'inclusion inverse. Soit $p = (p_1, \dots, p_n) \in Dual(H)$. Alors, il existe $l = (l_1, \dots, l_{n-1}) \in \mathcal{L}^{n-1}$ tel que $p \in \mathcal{R}^l$. De plus, nous déduisons du théorème 1 que

$$\forall s = (s_1, \dots, s_n) \in \mathcal{S}_-, p_n \geq \sum_{i=1}^{n-1} -s_i p_i + s_n$$

et

$$\forall s = (s_1, \dots, s_n) \in \mathcal{S}_-, p_n \leq \sum_{i=1}^{n-1} -s_i p_i + s_n$$

Cependant, puisque $p \in \mathcal{R}^l$, et donc $p_i \in \mathcal{R}(l_i)$, nous en déduisons que $-(c_i + \frac{1}{2}l_i)p_i \leq -(c_i - \frac{1}{2}l_i)p_i$. Ainsi, p vérifie les inéquations suivantes :

$$p_n \leq \sum_{i=1}^{n-1} -(c_i - \frac{1}{2}l_i)p_i + c_n + \frac{1}{2}$$

et

$$p_n \geq \sum_{i=1}^{n-1} -(c_i + \frac{1}{2}l_i)p_i + c_n - \frac{1}{2}$$

Nous en déduisons donc que $p \in Dual(s_-^l)^+ \cap Dual(s_+^l)^-$. □

Le corollaire 1 décrit le dual d'un hypervoxel comme l'union de 2^{n-1} polytopes infinis délimités par les hyperplans duaux des sommets de l'hypervoxel, et par les hyperplans verticaux associés aux axes du repère de l'espace (par exemple, en dimension 3, ces hyperplans sont les plans (O, X_1, X_3) et (O, X_2, X_3)). La figure 2.4 illustre cette propriété en dimensions 2 et 3.

2.2.3.2 Exemples en dimensions 2 et 3

Dans cette section, nous montrons quelques exemples de duaux d'hypervoxels pour les dimensions 2 et 3, en précisant les relations existant entre les sommets des hypervoxels et les hyperplans délimitant le bord de leur dual.

Dual d'un pixel

Les correspondances exactes entre les sommets d'un pixel et les droites délimitant son dual sont indiquées dans le tableau 2.1. Nous voyons que pour chaque portion de l'espace de paramètres ($Y_1 \leq 0$ et $Y_1 \geq 0$), deux droites délimitent la partie du dual du pixel correspondante.

La figure 2.5 montre quelques exemples de duaux de pixels. La correspondance entre les sommets des pixels et les droites délimitant leur dual est illustrée par la numérotation.

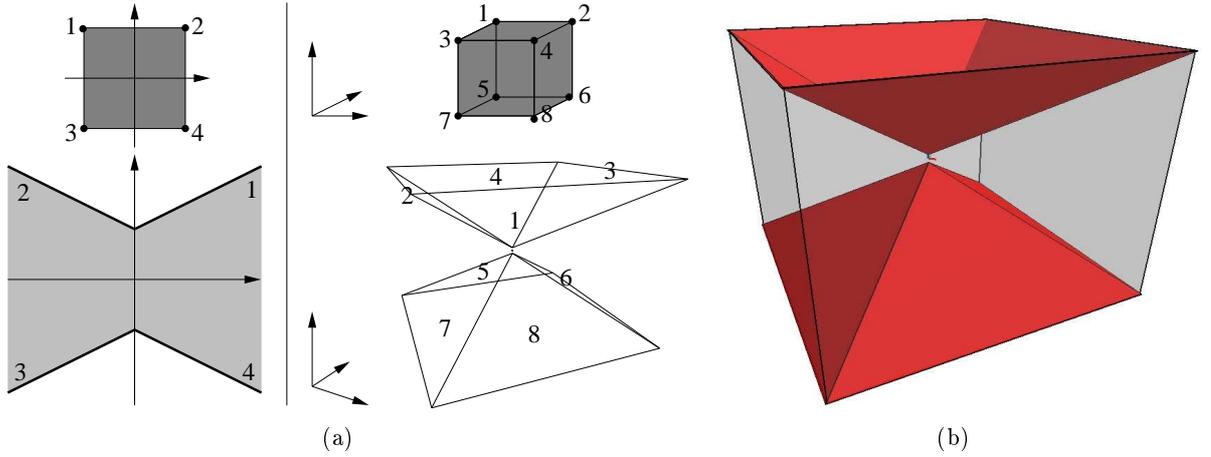


FIG. 2.4 – Exemples de duals d’hyperpixels en dimensions 2 et 3. La numérotation illustre la correspondance entre les sommets d’un pixel (resp. voxel) et les droites (resp. plans) constituant le bord de son dual : (a) Dual du pixel de centre $(0, 0)$, (b) Dual du voxel de centre $(0, 0, 0)$.

TAB. 2.1 – Relations entre le dual d’un pixel de centre $c = (c_1, c_2)$ et ses sommets.

	$Y_1 \leq 0$	$Y_1 \geq 0$
\mathcal{H}_-	$(c_1 + \frac{1}{2}, c_2 + \frac{1}{2})$	$(c_1 - \frac{1}{2}, c_2 + \frac{1}{2})$
\mathcal{H}_+	$(c_1 - \frac{1}{2}, c_2 - \frac{1}{2})$	$(c_1 + \frac{1}{2}, c_2 - \frac{1}{2})$

Dual d’un voxel

Les correspondances entre les sommets d’un voxel et les plans délimitant son dual sont indiquées dans le tableau 2.2. Nous voyons que pour chacune des quatre portions de l’espace de paramètres décrites, deux plans délimitent la partie du dual du voxel correspondante.

Quelques exemples de duals de voxels sont montrés sur les figures 2.6, 2.7 et 2.8.

2.2.4 Cas d’une droite en dimension 3

Dans le cadre de notre algorithme de reconstruction de volumes discrets (voir Chapitre 3.3), nous avons eu besoin de déterminer l’ensemble des plans contenant une droite euclidienne d en dimension 3. Nous appelons cet ensemble les *plans solutions* de la droite d , et le notons P_S . Nous

TAB. 2.2 – Relations entre le dual d’un voxel de centre $c = (c_1, c_2, c_3)$ et ses sommets.

	$Y_1 \leq 0, Y_2 \leq 0$	$Y_1 \leq 0, Y_2 \geq 0$
H_-	$(c_1 + \frac{1}{2}, c_2 + \frac{1}{2}, c_3 + \frac{1}{2})$	$(c_1 + \frac{1}{2}, c_2 - \frac{1}{2}, c_3 + \frac{1}{2})$
H_+	$(c_1 - \frac{1}{2}, c_2 - \frac{1}{2}, c_3 - \frac{1}{2})$	$(c_1 - \frac{1}{2}, c_2 + \frac{1}{2}, c_3 - \frac{1}{2})$
	$Y_1 \geq 0, Y_2 \leq 0$	$Y_1 \geq 0, Y_2 \geq 0$
H_-	$(c_1 - \frac{1}{2}, c_2 + \frac{1}{2}, c_3 + \frac{1}{2})$	$(c_1 - \frac{1}{2}, c_2 - \frac{1}{2}, c_3 + \frac{1}{2})$
H_+	$(c_1 + \frac{1}{2}, c_2 - \frac{1}{2}, c_3 - \frac{1}{2})$	$(c_1 + \frac{1}{2}, c_2 + \frac{1}{2}, c_3 - \frac{1}{2})$

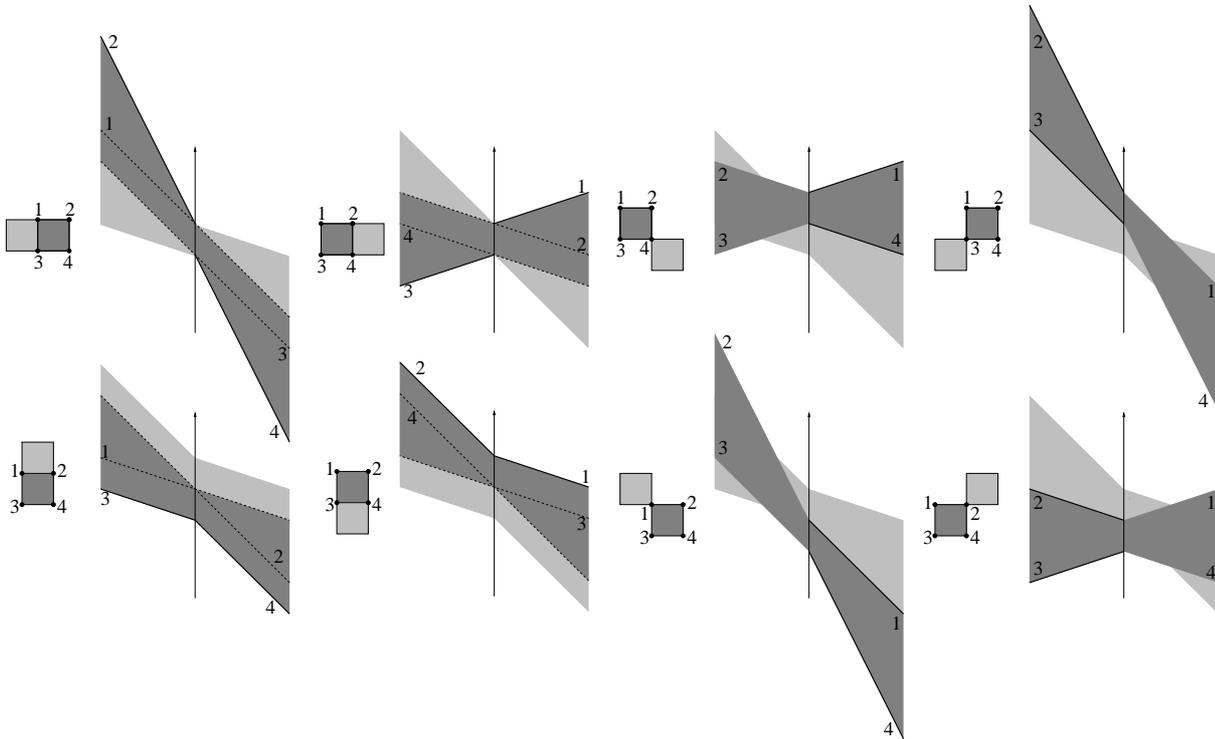


FIG. 2.5 – Quelques exemples de duaux de pixels. Le pixel coloré en gris clair, et commun à toutes les illustrations, est le pixel de centre $(1, -1)$.

le décrivons dans l'espace dual de la manière suivante :

Définition 21 (Plans solutions) Soit d une droite de \mathcal{E}_3 ou \mathcal{P}_3 . L'ensemble des plans solutions P_S de d est défini par

$$P_S(d) = \bigcap_{p \in d} \text{Dual}(p)$$

Propriété 3 Les duaux de k points alignés sont k plans sécants en une même droite.

A partir de la propriété 3, nous déduisons les paramètres de la droite correspondant aux plans solutions d'une droite en dimension 3.

Proposition 8 Soit d une droite de \mathcal{E}_3 définie par un point $p = (c_1, c_2, c_3)$ et un vecteur directeur $\vec{v} = (v_1, v_2, v_3) \neq (0, 0, v_3)$. La droite d' représentant les plans solutions de d est définie en fonction de \vec{v} par le point p' et le vecteur directeur \vec{v}' suivants :

Si $v_1 = 0$	Si $v_2 = 0$	Si $v_1 \neq 0$ et $v_2 \neq 0$
$p' = (0, \frac{v_3}{v_2}, c_3 - \frac{c_2 v_3}{v_2})$	$p' = (\frac{v_3}{v_1}, 0, c_3 - \frac{c_1 v_3}{v_1})$	$p' = (0, \frac{v_3}{v_2}, c_3 - \frac{c_2 v_3}{v_2})$
$\vec{v}' = (\frac{c_3}{c_1} - \frac{c_2 v_3}{c_1 v_2}, 0, \frac{c_2 v_3}{v_2} - c_3)$	$\vec{v}' = (0, \frac{c_3}{c_2} - \frac{c_1 v_3}{c_2 v_1}, \frac{c_1 v_3}{v_1} - c_3)$	$\vec{v}' = (\frac{v_3}{v_1}, -\frac{v_3}{v_2}, \frac{c_2 v_3}{v_2} - \frac{c_1 v_3}{v_1})$

Preuve : Soient $p_1 = (c_1, c_2, c_3)$ et $p_2 = (c_1 + v_1, c_2 + v_2, c_3 + v_3)$ deux points appartenant à d . Le but est de déterminer deux points de d' . Les plans duaux de p_1 et p_2 ont pour équations respectives

$$Y_3 = -c_1 Y_1 - c_2 Y_2 + c_3 \quad (2.1)$$

et

$$Y_3 = -(c_1 + v_1)Y_1 - (c_2 + v_2)Y_2 + c_3 + v_3 \quad (2.2)$$

Nous avons alors la relation suivante :

$$-Y_1v_1 - Y_2v_2 + v_3 = 0 \quad (2.3)$$

Pour obtenir les points recherchés, il suffit de déterminer deux solutions évidentes de l'équation 2.3 telles que $Y_1 = 0, Y_2 = \frac{v_3}{v_2}$ et $Y_3 = 0, Y_1 = \frac{v_3}{v_1}$. Une solution pour Y_3 peut être directement déduite de l'équation 2.1. Ces deux premiers points solutions ne sont valables que dans le cas où $v_1 \neq 0$ et $v_2 \neq 0$. Si $v_1 = 0$, on en déduit d'après l'équation 2.3 que $Y_2 = \frac{v_3}{v_2}$. Il suffit alors de déterminer deux points qui sont des solutions évidentes de l'équation 2.1, telles que $Y_1 = 0, Y_3 = c_3 - \frac{c_2v_3}{v_2}$ et $Y_3 = 0, Y_1 = \frac{c_3}{c_1} - \frac{c_2v_3}{c_1v_2}$. De la même manière, deux points solutions peuvent être trouvés dans le cas où $v_2 = 0$. \square

Proposition 9 *Soit d une droite de \mathcal{P}_3 définie par un point $p = (c_1, c_2, c_3)$ et un vecteur directeur $\vec{v} = (v_1, v_2, v_3) \neq (0, 0, v_3)$. La droite d' représentant les plans solutions de d est définie en fonction de \vec{v} par le point p' et le vecteur directeur \vec{v}' suivants :*

$Si\ v_1 = 0$	$Si\ v_2 = 0$	$Si\ v_1 \neq 0\ et\ v_2 \neq 0$
$p' = (0, -\frac{v_3}{v_2}, c_3 - \frac{c_2v_3}{v_2})$	$p' = (-\frac{v_3}{v_1}, 0, c_3 - \frac{c_1v_3}{v_1})$	$p' = (0, -\frac{v_3}{v_2}, c_3 - \frac{c_2v_3}{v_2})$
$\vec{v}' = (\frac{c_2v_3}{c_2v_2} - \frac{c_3}{c_1}, 0, \frac{c_2v_3}{v_2} - c_3)$	$\vec{v}' = (0, \frac{c_1v_3}{c_2v_1} - \frac{c_3}{c_2}, \frac{c_1v_3}{v_1} - c)$	$\vec{v}' = (-\frac{v_3}{v_1}, \frac{v_3}{v_2}, \frac{c_2v_3}{v_2} - \frac{c_1v_3}{v_1})$

Preuve : La preuve est similaire à celle de la proposition 8. \square

2.3 La notion de préimage généralisée

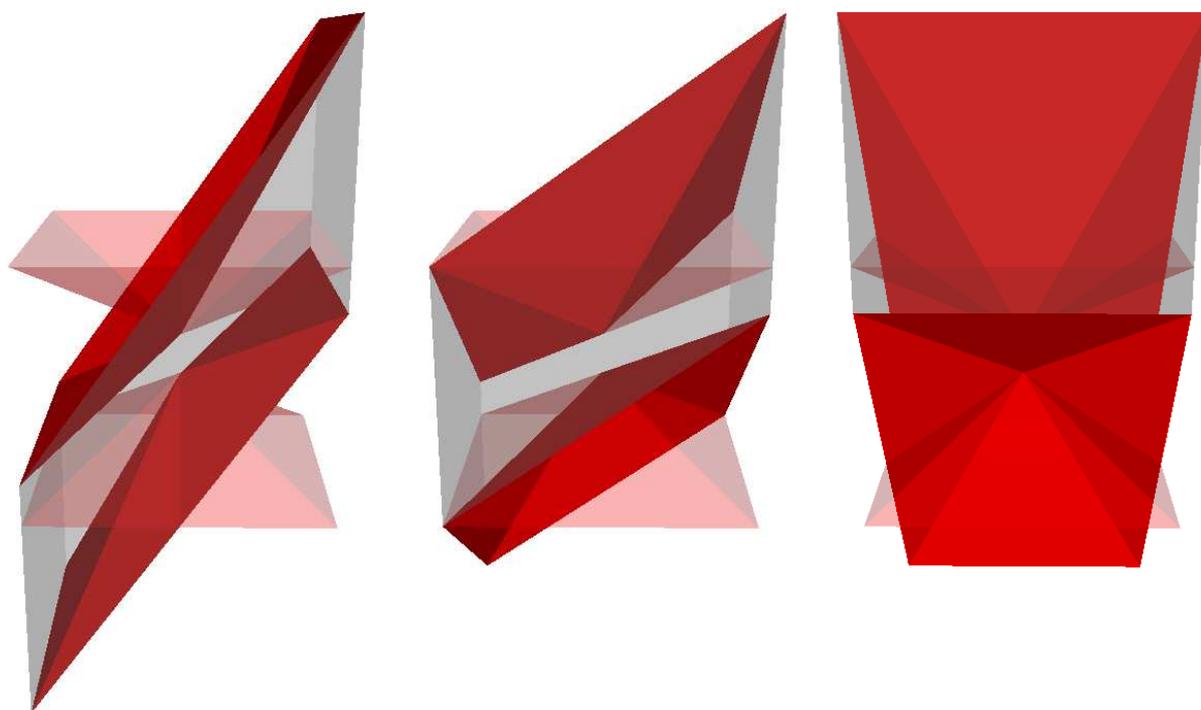
Dans cette section, nous donnons tout d'abord la définition de la préimage généralisée d'un ensemble de polytopes. Nous illustrons cette définition en décrivant les différentes formes que peut prendre la préimage généralisée d'un ensemble de pixels, et en montrant succinctement quelques exemples de préimages généralisées de voxels.

2.3.1 Définition

Dans cette thèse, nous avons défini la notion de préimage généralisée d'un ensemble de polytopes. Cette préimage est un objet de l'espace de paramètres tel que chaque point de cet objet est associé à un hyperplan coupant tous les polytopes donnés.

Définition 22 (Préimage généralisée) *Soit $\mathcal{P} = \{P_1, \dots, P_k\}$ un ensemble de k n -polytopes, $k \in \mathbb{N}^*$. La préimage généralisée $\mathbb{G}_{\mathcal{P}}$ de l'ensemble de polytopes \mathcal{P} est définie par*

$$\mathbb{G}_{\mathcal{P}}(\mathcal{P}) = \bigcap_{i=1}^k Dual(P_i)$$

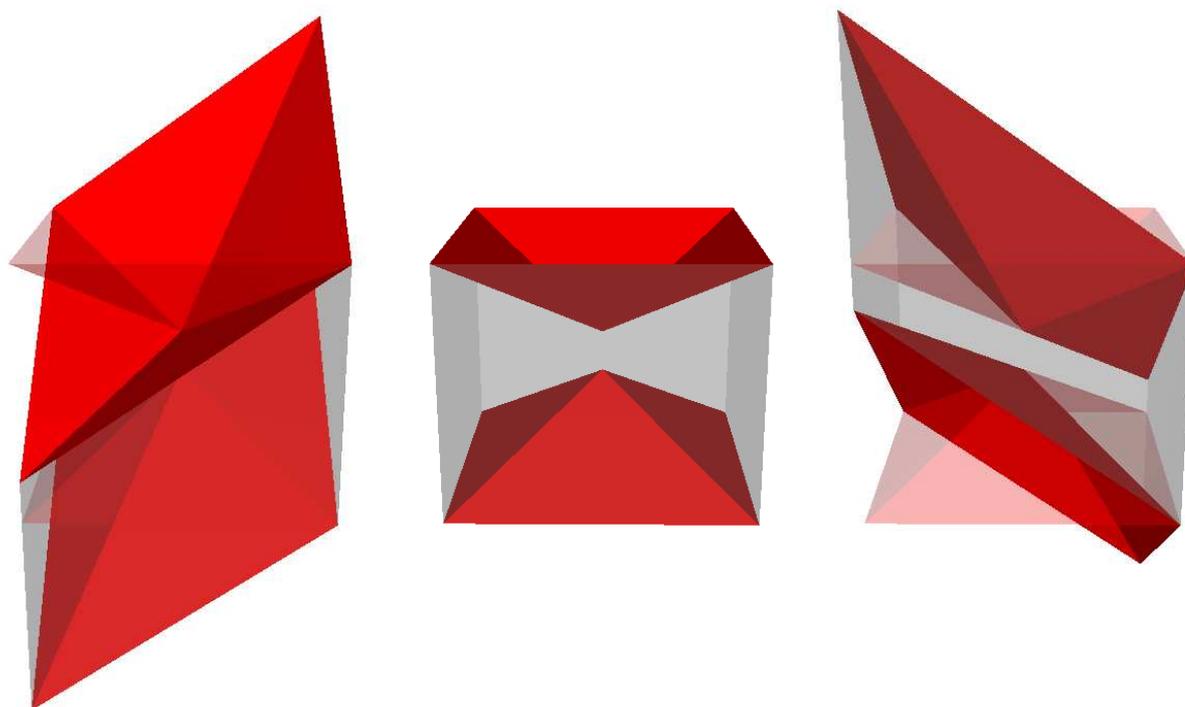


Dual du voxel $(-1, -1, 0)$

Dual du voxel $(-1, 0, 0)$

Dual du voxel $(-1, 1, 0)$

FIG. 2.6 – Exemples de duals de voxels.

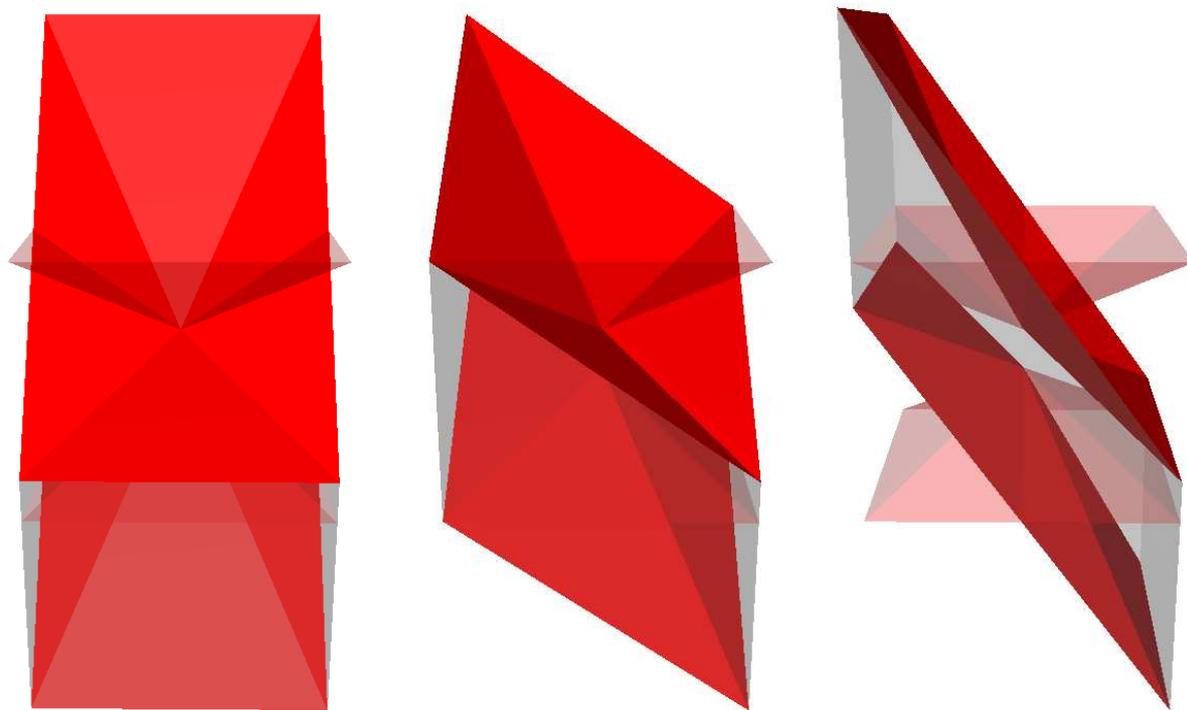


Dual du voxel $(0, -1, 0)$

Dual du voxel $(0, 0, 0)$

Dual du voxel $(0, 1, 0)$

FIG. 2.7 – Exemples de duaux de voxels.



Dual du voxel $(1, -1, 0)$

Dual du voxel $(1, 0, 0)$

Dual du voxel $(1, 1, 0)$

FIG. 2.8 – Exemples de duals de voxels.

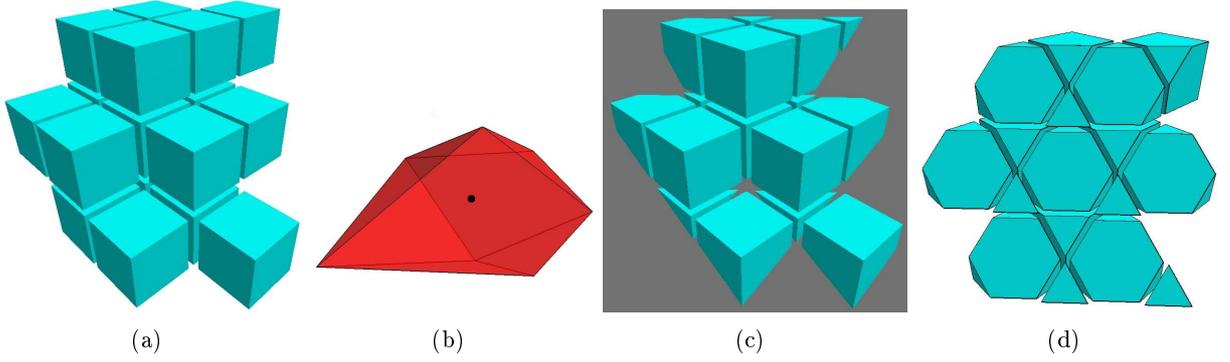


FIG. 2.9 – Exemple de préimage généralisée en dimension 3 : (a) Un ensemble \mathcal{V} de dix-sept voxels, (b) Préimage généralisée correspondante $\mathbb{G}_P(\mathcal{V})$. Son barycentre B est matérialisé par un point noir, (c) et (d) Résultat de l'intersection entre \mathcal{V} et le plan dual de B . Nous pouvons remarquer que le plan correspondant à $Dual(B)$ coupe bien l'ensemble des voxels de \mathcal{V} .

Le dual d'un polytope étant lui-même un polytope, nous déduisons de la définition 22 que la préimage généralisée d'un ensemble de polytopes est soit vide, soit elle aussi un polytope, puisqu'elle est décrite comme l'intersection d'un ensemble de polytopes. Nous pouvons voir un exemple de préimage généralisée en dimension 3 sur la figure 2.9b. Les voxels considérés sont représentés sur la figure 2.9a.

De plus, la définition de cette préimage présente plusieurs avantages tels que le fait d'être totalement indépendant de la position des polytopes considérées, ainsi que de leurs relations de connexité ou encore de l'ordre dans lequel ils sont considérés.

2.3.2 Préimage généralisée d'hypervoxels

Nous illustrons ici la définition de la préimage généralisée donnée dans la section précédente en nous intéressant plus particulièrement aux préimages généralisées d'un ensemble de pixels 1-connexe et d'un ensemble de voxels 2-connexes.

2.3.2.1 Étude de la préimage généralisée d'un ensemble de pixels 1-connexe

Afin de décrire plus précisément la préimage généralisée d'un ensemble de pixels, nous nous plaçons dans le cadre du corollaire 1 du chapitre précédent qui décrit le dual d'un pixel comme l'union de deux polygones infinis.

Lorsqu'elle n'est pas vide, la préimage généralisée d'un ensemble de pixels peut avoir quatre formes différentes (polygones finis ou infinis), chaque forme dépendant directement de la position des pixels les uns par rapport aux autres. Les quatre formes possibles sont les suivantes :

- Deux polygones infinis (voir Figure 2.10) : cette forme de la préimage généralisée correspond à plusieurs cas de figures, tels que deux pixels 1-voisins alignés horizontalement, ou encore k pixels alignés verticalement, $k \in \mathbb{N}^*$.
- Un polygone infini (voir Figure 2.11) : la préimage généralisée d'un ensemble de pixels 1-connexe sera composée d'un unique polygone infini dans le cas de deux rangées de pixels alignés verticalement de part et d'autre d'une même droite verticale (hormis dans le cas de deux pixels alignés horizontalement décrit précédemment).

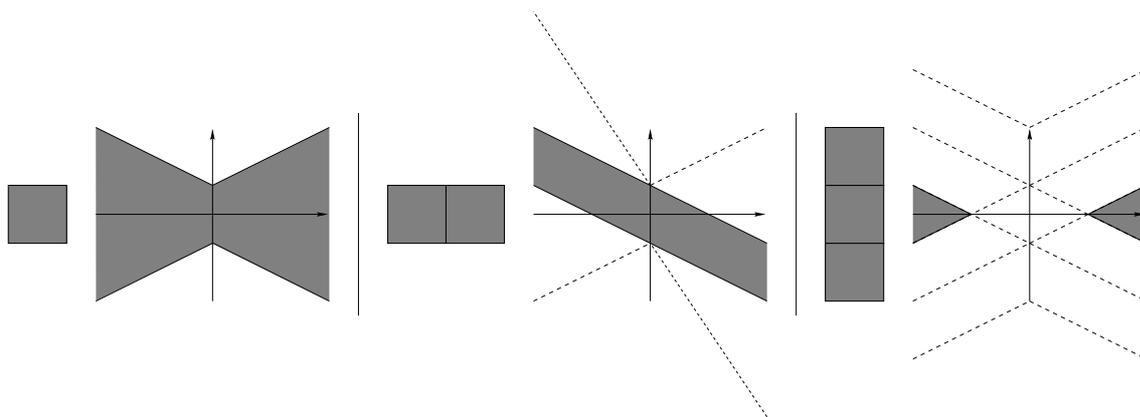


FIG. 2.10 – Préimage généralisée de pixels : deux polygones infinis.

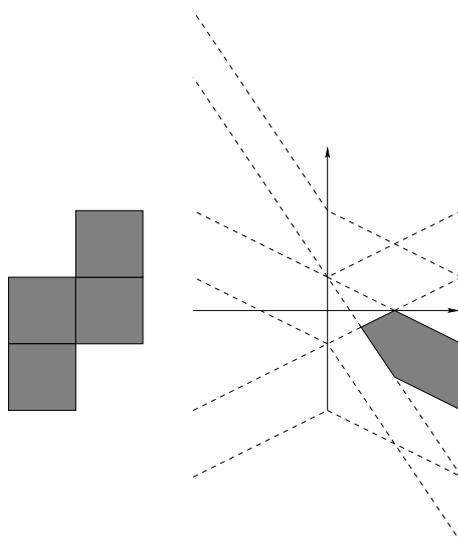


FIG. 2.11 – Préimage généralisée de pixels : un polygone infini.

- Deux polygones finis (voir Figure 2.12) : ce cas de figure correspond à un ensemble de k pixels alignés horizontalement, $k \geq 3$.
- Un polygone fini (voir Figure 2.14) : ce dernier cas de figure correspond à toute configuration de pixels autre que les trois précédemment décrites. En particulier, dans ce cas précis, la préimage généralisée d'un ensemble de pixels est exactement identique à la préimage classique.

2.3.2.2 Cas d'un ensemble de voxels 2-connexe

Tout comme la préimage généralisée de pixels, la préimage généralisée d'un ensemble de voxels peut avoir différentes formes en fonction des différentes positions des voxels. Cependant, le nombre de cas possibles étant très important, nous ne les décrirons pas ici. Nous présentons simplement quelques exemples de préimages généralisées de voxels sur les figures 2.9 et 2.13.

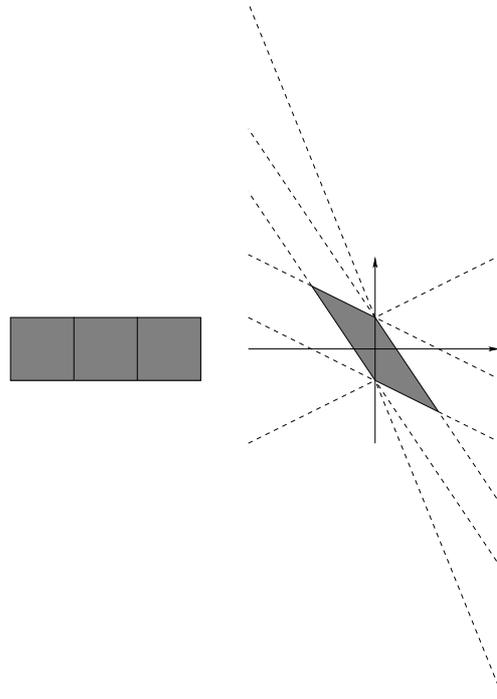


FIG. 2.12 – Préimage généralisée de pixels : deux polygones finis.

2.4 Reconnaissance d'hyperplans discrets

Dans cette section, nous présentons notre algorithme de reconnaissance d'hyperplans U-Supercouverture. Puis, nous donnons quelques exemples d'applications, et notamment dans le cadre des espaces discrets classiques.

2.4.1 Algorithme de reconnaissance

Soient \mathbb{D}^n un espace discret de dimension n et \mathcal{P} un ensemble de pavés de cet espace. La reconnaissance d'un hyperplan discret U-Supercouverture est effectuée simplement en calculant la préimage généralisée de \mathcal{P} . Si cette préimage est vide, l'ensemble de pavés n'appartient pas à un hyperplan discret. Dans le cas contraire, l'ensemble de pavés appartient à un hyperplan discret, et la préimage généralisée fournit l'ensemble des hyperplans euclidiens solutions.

L'algorithme de reconnaissance est le suivant :

2.4.2 Exemples d'applications

L'algorithme 1 peut être directement appliqué aux différents modèles de discrétisation classiques, tels que les modèles Supercouverture, Standard ou Naïfs. Nous décrivons rapidement dans cette partie de quelle manière adapter notre algorithme à ces différents modèles.

2.4.2.1 Hyperplans Supercouverture

Dans le cadre des espaces discrets classiques, la reconnaissance d'hyperplans Supercouverture se fait simplement en utilisant l'algorithme 1 dans lequel les pavés sont remplacés par des

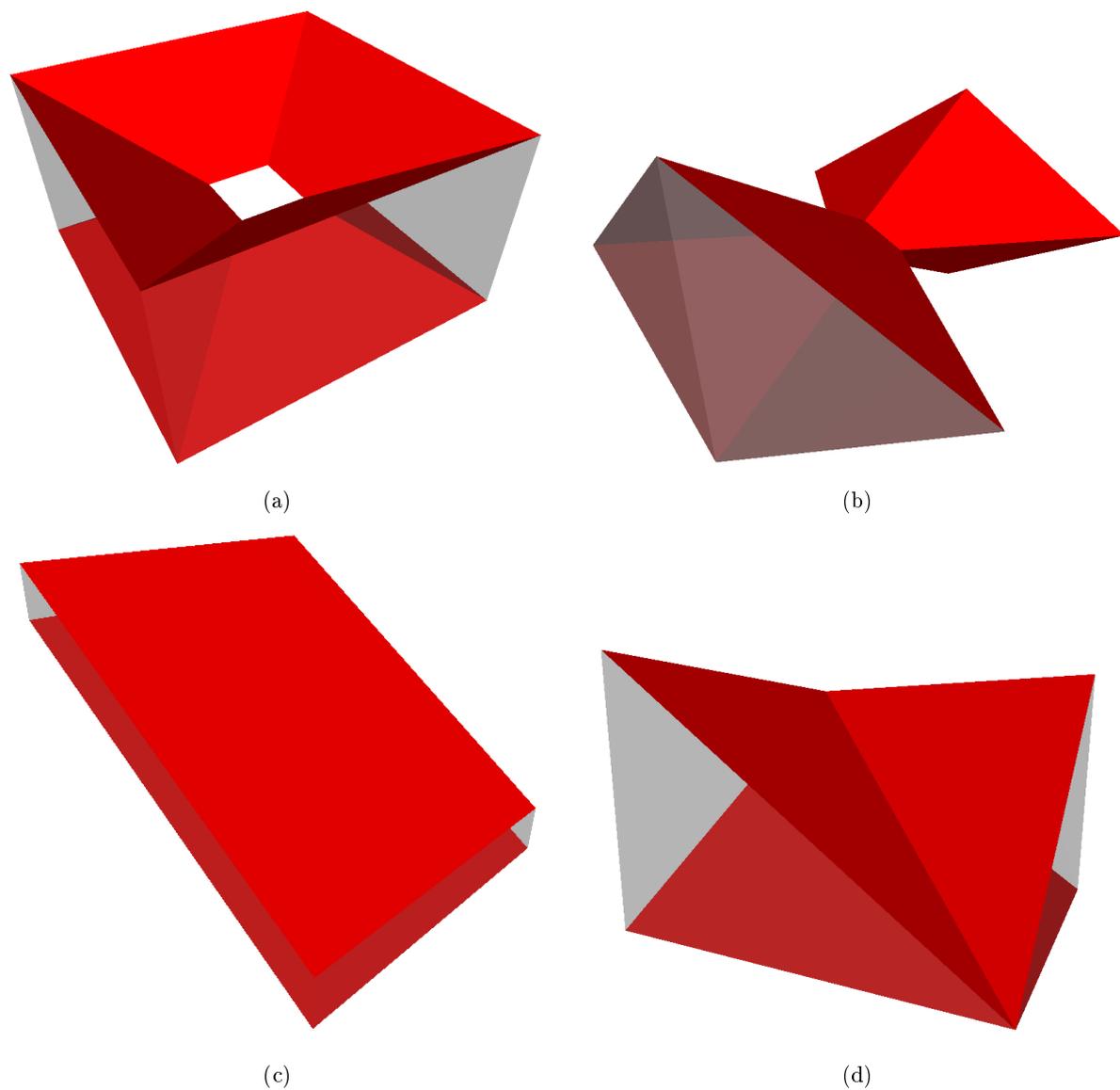


FIG. 2.13 – Exemples de préimages généralisées en dimension 3 : (a) Préimage généralisée de trois voxels alignés sur l'axe X_3 , (b) Préimage généralisée de trois voxels alignés sur l'axe X_2 , (c) Préimage généralisée de quatre voxels formant un carré sur le plan (O, X_1, X_2) , (d) Préimage généralisée de quatre voxels formant un carré sur le plan (O, X_2, X_3) .

Algorithme 1 : Reconnaissance d'hyperplans discrets U-Supercouverture

Données : Un ensemble \mathcal{P} de k pavés P_1, \dots, P_k .

début
 $\mathbb{G}_P \leftarrow Dual(P_1);$
 $i \leftarrow 2;$
tant que $\mathbb{G}_P \neq \emptyset$ **et** $i \leq k$ **faire**
 $\mathbb{G}_P \leftarrow \mathbb{G}_P \cap Dual(P_i);$
 $i \leftarrow i + 1;$
si $\mathbb{G}_P \neq \emptyset$ **alors**
 \mathcal{P} appartient à un hyperplan discret.

sinon
 \mathcal{P} n'appartient pas à un hyperplan discret.

fin

hypervoxels. Un exemple de calcul de la préimage généralisée d'un ensemble de pixels est montré sur la figure 2.14.

De même, dans le cadre des grilles irrégulières isothétiques, la reconnaissance de droites discrètes se fait en appliquant l'algorithme 1 aux pavés correspondant à ces espaces particuliers.

2.4.2.2 Hyperplans Standard

Dans le cas du modèle Standard, le fait que la préimage généralisée de l'ensemble d'hypervoxels considérés soit non vide, et non réduite à son bord, suffit à déterminer l'appartenance des hypercubes à un hyperplan Standard. En outre, certains points du bord de la préimage généralisée ne sont pas associés à un hyperplan solution. Une méthode simple permettant de choisir un hyperplan solution consiste donc à choisir un point de la préimage généralisée ne se trouvant pas sur son bord.

2.4.2.3 Modèles Naïfs

Enfin, considérons le cas des modèles Naïf et Naïf fermé. Pour ces deux modèles, la distance utilisée est la distance d^1 (voir Section 1.3.2). Le problème de la reconnaissance d'hyperplans Naïfs ou Naïfs fermés se ramène alors à déterminer s'il existe des hyperplans euclidiens qui coupent les boules définies pour la distance d^1 et ayant pour centre un point discret. Ceci peut être aisément effectué en calculant la préimage généralisée de ces boules.

2.5 Conclusion

Dans ce chapitre, nous avons commencé par définir l'*espace de paramètres* dans lequel nous avons été amenés à travailler durant cette thèse. Chaque point de cet espace est associé à un hyperplan de l'espace euclidien et réciproquement. A partir de cette définition, nous avons défini ce que nous appelons le *dual* d'un objet O , tel que chaque point de ce dual est associé à un hyperplan passant par au moins un point de O . Nous avons alors donné une description géométrique du dual d'un polytope à partir des duals de ses sommets, ce dual étant décrit comme l'intersection de deux unions finies de demi-espaces. Nous en avons ensuite déduit une description du dual d'un polytope particulier, l'hypervoxel, comme étant l'union de plusieurs polytopes

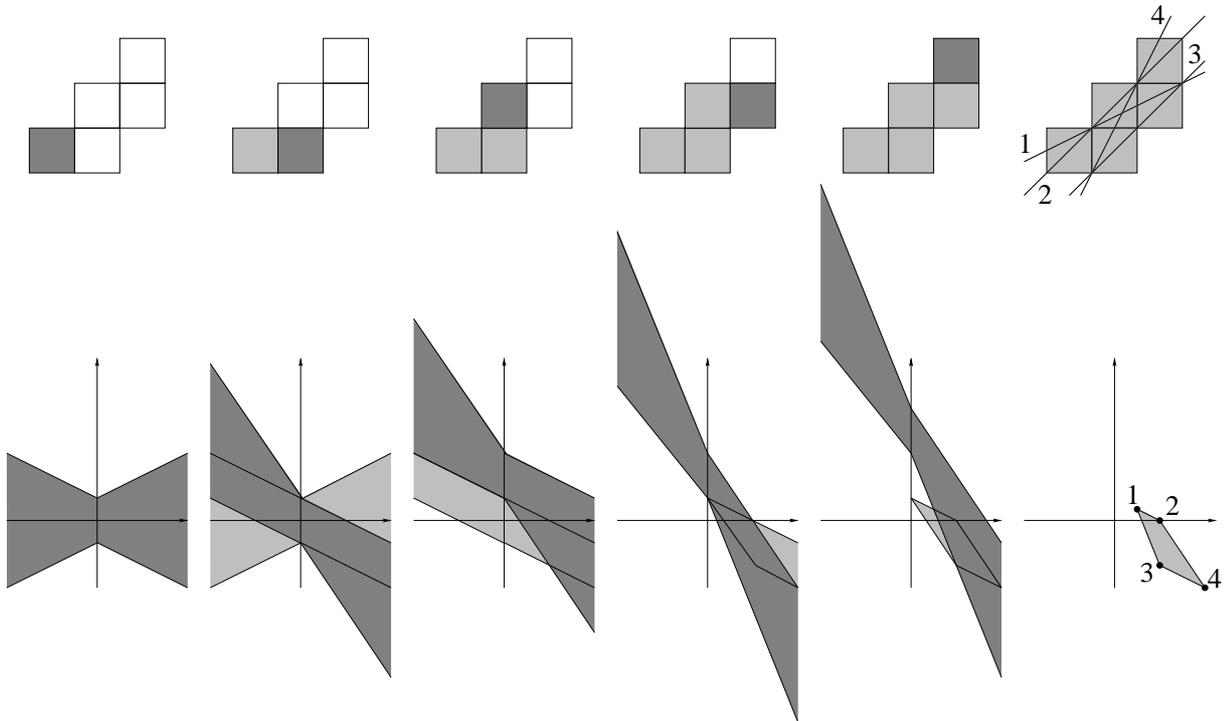


FIG. 2.14 – Exemple de calcul d'une préimage généralisée de pixels.

infinis. Enfin, nous avons abordé le cas des droites en dimension 3 en décrivant l'ensemble des plans contenant cette droite sous la forme d'une droite dans l'espace dual.

Dans un deuxième temps, nous avons défini la *préimage généralisée* d'un ensemble de polytopes. Cette préimage est définie dans l'espace de paramètre précédemment évoqué à partir des duaux des polytopes, et correspond à l'ensemble des hyperplans euclidiens coupant tous les polytopes. Nous avons de plus donné une autre description de la préimage d'un ensemble d'hypervoxels.

Nous avons alors déduit de cette définition un algorithme de reconnaissance d'hyperplans U-Supercouverture. Nous avons de plus expliqué de quelle manière cet algorithme peut être utilisé pour la reconnaissance d'hyperplans Supercouverture, Standard et Naïfs.

Dans tous les cas, notre algorithme est incrémental puisque la préimage généralisée des polytopes associés aux points discrets considérés est calculée au fur et à mesure de l'ajout des points discrets. De plus, les points discrets peuvent être ajoutés dans n'importe quel ordre.

La description de la préimage généralisée d'un ensemble d'hypervoxels, ainsi que celle de la préimage généralisée de polygones ont été présentées dans deux conférences internationales avec actes [DA06a, DA06c, DCA06].

Chapitre 3

Reconstruction analytique inversible de courbes et volumes discrets

Sommaire

3.1	Introduction	44
3.2	Reconstruction inversible en dimension 2	46
3.2.1	Méthodes de reconstruction proposées par R. BRETON [Bre03]	47
3.2.1.1	Méthode avec joints [BSDA03]	47
3.2.1.2	Méthode sans joints [SBDA05]	48
3.2.1.3	Bilan	49
3.2.2	Méthode étendue	50
3.2.2.1	Principe	50
3.2.2.2	Reconnaissance des segments discrets	52
3.2.2.3	Polygonalisation de la courbe	53
3.2.3	Résultats	54
3.2.3.1	Méthode étendue	54
3.2.3.2	Comparaison avec les méthodes avec et sans joints	56
3.2.4	Reconstruction de courbes non simples	56
3.2.5	Conclusion et perspectives	59
3.3	Reconstruction de courbes et volumes discrets en dimension 3	63
3.3.1	Données traitées	64
3.3.2	Principe	64
3.3.2.1	Surfel de départ et méthodes de propagation	65
3.3.2.2	Ordre de reconnaissance	67
3.3.3	Reconnaissance de plan contrainte	67
3.3.3.1	Contrainte de sommet	67
3.3.3.2	Contrainte d'arête	67
3.3.3.3	Choix du plan	68
3.3.4	Polygonalisation de face	68
3.3.4.1	Reconstruction d'une 2-courbe discrète planaire	69
3.3.4.2	Ensemble de solution de surfels vide	72
3.3.5	Résultats	72
3.3.6	Conclusion et perspectives	78
3.4	Conclusion du chapitre	85

3.1 Introduction

Dans ce chapitre, nous nous intéressons au problème de la reconstruction de données discrètes. Il s'agit là du problème inverse de la discrétisation, à savoir, comment obtenir une représentation "continue" d'un ensemble discret ? A titre d'exemple, nous pouvons voir sur la figure 3.1b un objet en dimension 3 obtenu à l'aide d'un algorithme de reconstruction appliqué sur l'objet original de la figure 3.1a⁷.



FIG. 3.1 – Exemple d'objet reconstruit à partir d'images : (a) *Stanford Bunny*, (b) Reconstruction obtenue à partir de plusieurs clichés de l'objet original.

Le problème de la reconstruction de données discrètes est un problème classique en informatique graphique, auquel plusieurs solutions ont été proposées. La méthode la plus connue est celle des *Marching Cubes* [LC87]. Cette méthode, proposée initialement en dimension 3, et adaptée ensuite à la dimension 2, est basée sur l'étude de configurations locales de voxels. Le nombre de facettes générées par cette méthode est en particulier très élevé, et ce à cause du caractère local de la reconstruction. Ainsi, cette méthode fournit un résultat visuel de très bonne qualité et très proche de l'objet discret d'origine. En contrepartie, l'inconvénient majeur de cette méthode est que la taille mémoire nécessaire au stockage de tels objets est particulièrement importante. De nombreuses méthodes de simplification d'objets de type *Marching Cubes* ont donc été proposées [LLVT03, Che95, HHVW96, Nie03, Nie04] dans le but de diminuer le nombre de facettes de l'objet final. De même, d'autres méthodes, là encore basées sur l'étude de configurations locales de voxels, ont été proposées [MSS94, HWC⁺05, LM00, KL02].

Notre approche se situe plutôt dans la continuité des approches suivies par la communauté française en géométrie discrète ces dernières années, c'est-à-dire une approche basée non pas sur l'étude des configurations locales de voxels, mais sur la reconnaissance de morceaux de droites et plans discrets constituant les objets. L'idée est d'obtenir directement un objet constitué de grands segments de droites ou facettes sans avoir besoin de recourir à une phase de simplification. Cette approche a aussi pour but d'essayer d'obtenir au final des objets avec un nombre inférieur de

⁷Image extraite de la page web <http://graphics.stanford.edu/data/3Dscanrep/>.

facettes par rapport aux méthodes basées sur l'étude de configurations locales de voxels.

Dans cette approche, les méthodes proposées, que nous appelons méthodes de *reconstruction discrète analytique*, sont en général composées de deux étapes :

- Une étape de reconnaissance visant à décomposer l'objet discret en plusieurs morceaux discrets (segments de droites en dimension 2 ou morceaux de plans en dimension 3),
- Une étape de polygonalisation consistant à obtenir des polygones euclidiens (segments de droites ou facettes en dimensions 2 et 3) à partir des morceaux discrets reconnus.

Cette méthode est illustrée sur la figure 3.2, pour la dimension 2.

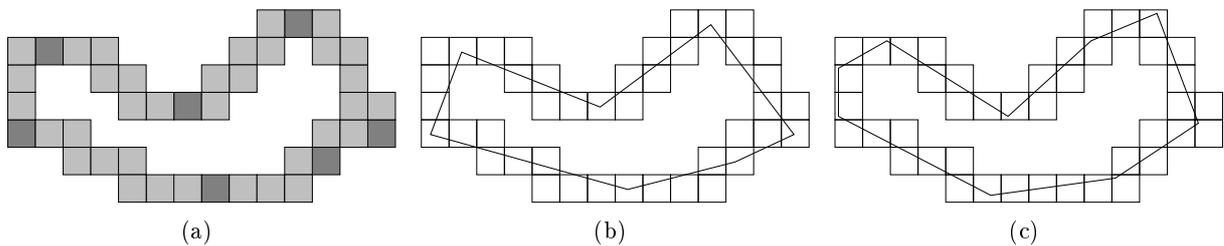


FIG. 3.2 – Reconstruction d'une 1-courbe discrète : (a) Résultat de l'étape de reconnaissance de segments discrets. Les pixels gris foncés appartiennent à deux segments discrets différents, (b) Une reconstruction possible (chaque segment discret reconnu est remplacé par un segment euclidien), (c) Exemple de reconstruction inversible pour le modèle Standard.

Dans nos travaux, nous nous sommes principalement intéressés à l'étude de méthodes de reconstruction discrète analytique. Nous souhaitons en particulier que la reconstruction obtenue vérifie plusieurs critères :

- Le premier critère est la réversibilité de la reconstruction (voir Figure 3.2c). En effet, nous souhaitons que la discrétisation de l'objet obtenu suivant un modèle de discrétisation donné, dans notre cas le modèle Standard, soit identique à l'objet discret de départ. La propriété de réversibilité assure alors que notre reconstruction est proche des données discrètes originelles. C'est une propriété qui est souvent perdue lors des étapes de simplifications qui suivent les méthodes de reconstruction basées sur l'étude de configurations locales de voxels, telles que la méthode des Marching Cubes.
- Le deuxième critère est que l'objet reconstruit doit être topologiquement cohérent avec l'objet de départ. Nous désirons en particulier que la reconstruction respecte le genre, le nombre de composantes connexes, etc ..., des objets euclidiens.

En dimension 2, plusieurs méthodes répondant à nos critères fournissent des résultats très convenables [BSDA03, SBDA05, LI03, CZ06]. Par ailleurs, en dimension 3, seule la méthode de reconstruction proposée par D. CŒURJOLLY ET AL. [CGS04, CDJS06], qui est une méthode "hybride" consistant à simplifier un objet de type Marching Cubes à l'aide d'outils de reconnaissance discrète, répond à nos critères.

Dans cette thèse, nous proposons deux nouvelles méthodes de reconstruction analytique discrète en dimensions 2 et 3 fournissant une reconstruction inversible pour le modèle Standard et topologiquement correcte.

Nous présentons d'une part une nouvelle méthode de reconstruction de 1-courbes discrètes ouvertes ou fermées, simples ou non simples, qui améliore notamment les résultats obtenus par les deux méthodes proposées dans [Bre03, BSDA03, SBDA05]. En particulier, le nombre moyen de segments générés est inférieur à ceux générés par les deux méthodes sus-évoquées.

D'autre part, nous proposons une nouvelle méthode qui reconstruit la surface d'un volume discret 2-connexe en une surface polygonale. Cette méthode est la première méthode de reconstruction analytique vérifiant les deux critères demandés proposée à ce jour.

3.2 Reconstruction inversible en dimension 2

Dans cette section, nous nous intéressons au problème de la reconstruction inversible de courbes discrètes classiques en dimension 2. Notre but est d'obtenir, à partir d'une 1-courbe discrète, une représentation polygonale de cette courbe. De plus, nous souhaitons que la reconstruction soit inversible pour le modèle Standard, c'est à dire que la discrétisation suivant le modèle Standard de la courbe polygonale obtenue soit identique à la courbe discrète d'origine. D'une manière simplifiée, cette propriété est équivalente au fait que la courbe polygonale reconstruite coupe tous les pixels de la courbe discrète. Enfin, nous souhaitons pouvoir reconstruire des courbes discrètes simples ouvertes ou fermées (voir Figure 3.3a), ou encore des courbes discrètes non simples, c'est à dire des courbes ayant un ou plusieurs pixels communs avec d'autres courbes (voir Figure 3.3b).

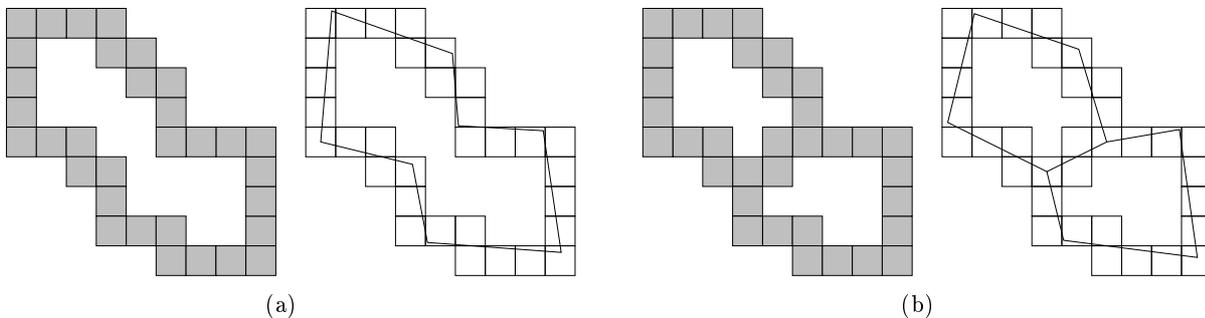


FIG. 3.3 – Exemples de reconstructions inversibles possibles de 1-courbes discrètes simples et non simples : (a) Courbe simple fermée, (b) Courbe non simple.

Parmi les diverses méthodes de reconstruction en dimension 2 existantes, nous nous sommes particulièrement intéressés aux deux méthodes de reconstruction proposées par R. BRETON dans sa thèse [Bre03]. Ces deux méthodes permettent la reconstruction inversible pour le modèle Standard de 1-courbes discrètes simples. Cependant, la première méthode, appelée *méthode avec joints* [BSDA03], présente l'inconvénient de nécessiter deux traitements successifs de la courbe discrète, ce qui présente un temps d'exécution important. Le problème majeur de cette méthode est toutefois qu'elle peut difficilement être utilisée pour reconstruire des courbes non simples et qu'elle ne s'étend difficilement aux dimensions supérieures.

Une deuxième méthode, appelée *méthode sans joint* [SBDA05], a par conséquent été proposée. Cette dernière possède de nombreux avantages par rapport à la précédente car elle ne

nécessite qu'un seul parcours de la courbe discrète, permet de gérer des courbes non simples et peut s'étendre aux dimensions supérieures. Néanmoins, cette deuxième méthode présente l'inconvénient de générer un nombre de segments nettement supérieur à celui généré par la première méthode avec joints.

Pour pallier ce dernier inconvénient, nous avons travaillé sur une nouvelle méthode de reconstruction de courbes discrètes 1-connexes, inspirée de la méthode sans joint, mais qui permet de limiter le nombre de segments générés. Nous avons de plus cherché une manière de l'utiliser pour la reconstruction de courbes non simples.

Dans la suite, nous présentons tout d'abord rapidement les principes des deux méthodes proposées par R. BRETON. Puis, nous décrivons notre méthode de reconstruction, baptisée *méthode étendue*. Nous donnons ensuite quelques résultats obtenus avec les trois méthodes et les comparons entre eux. Enfin, nous présentons rapidement une méthode de reconstruction de courbes discrètes non simples.

3.2.1 Méthodes de reconstruction proposées par R. BRETON [Bre03]

Dans sa thèse, R. BRETON propose deux méthodes de reconstruction analytique inversible de courbes discrètes classiques. Ces deux méthodes, appelées *méthode avec joints* et *méthode sans joint*, reconstruisent des 1-courbes simples, ouvertes ou fermées, et fournissent une reconstruction inversible pour le modèle Standard.

Elles sont basées sur les deux processus suivants :

- La reconnaissance de segments discrets constituant la courbe. Pour les deux méthodes, l'algorithme de reconnaissance utilisé est l'algorithme de reconnaissance basé sur le calcul de la préimage classique de points discrets proposé par J. VITTONI [Vit99]. Initialement, cet algorithme permet la reconnaissance de droites Naïves mais est néanmoins applicable à la reconnaissance de droites Standard grâce à l'utilisation d'une transvection des points discrets [Via96]. L'algorithme de calcul de la préimage a par ailleurs été étendu au cas des droites Standard par I. SIVIGNON [Siv04].
- Le remplacement des segments discrets reconnus par un ensemble de segments euclidiens, de manière à ce que la discrétisation Standard de la courbe polygonale reconstruite soit identique à la courbe discrète originale.

Nous présentons rapidement ces deux méthodes dans les sections suivantes.

3.2.1.1 Méthode avec joints [BSDA03]

Le but de cette première méthode est de tenter de reconstruire une courbe polygonale constituée d'un très faible nombre de segments. Cette méthode est appliquée en deux étapes dont le principe est détaillé ci-après.

La première étape consiste tout d'abord à effectuer une reconnaissance « au plus loin » de segments discrets composant la courbe, c'est à dire essayer de reconnaître des segments de taille maximale en partant d'un point discret donné. Chaque segment reconnu doit de plus avoir un pixel commun avec les segments le précédant et le suivant dans la courbe discrète. Pour ce faire, un premier point discret, ainsi qu'un sens de parcours de la courbe sont arbitrairement choisis.

Puis, un premier segment est reconnu en ajoutant un à un les points discrets consécutifs au point de départ, et en calculant la préimage correspondante. Lorsque cette dernière est vide, le dernier point discret ajouté lors de la reconnaissance ne fait pas partie du segment discret reconnu et une nouvelle reconnaissance est alors effectuée à partir du dernier pixel du segment reconnu. Le processus est réitéré jusqu'à atteindre la fin de la courbe (cas d'une courbe ouverte) ou le point de départ (cas d'une courbe fermée).

Ensuite, pour chaque segment discret reconnu, une *droite euclidienne solution* (i.e. dont la discrétisation Standard contient le segment discret) est déterminée à partir de la préimage calculée pour le segment. Cependant, le choix des droites solutions effectué avec cette méthode ne garantit pas la réversibilité de la reconstruction obtenue. En effet, deux droites euclidiennes solutions choisies pour deux segments discrets consécutifs ne se coupent pas nécessairement à l'intérieur du pixel commun aux deux segments (voir Figure 3.4c).

Une deuxième étape, consistant à ajouter des segments euclidiens (appelés joints) à la courbe afin d'obtenir une reconstruction inversible, est alors effectuée (voir Figure 3.4d).

Cependant, comme cette méthode effectue les étapes de reconnaissance et de reconstruction en deux parcours séparés, elle ne permet que difficilement le traitement de courbes non simples, ce qui est un problème majeur. De plus, le nombre de segments reconstruits se voit augmenté par l'ajout de joints.

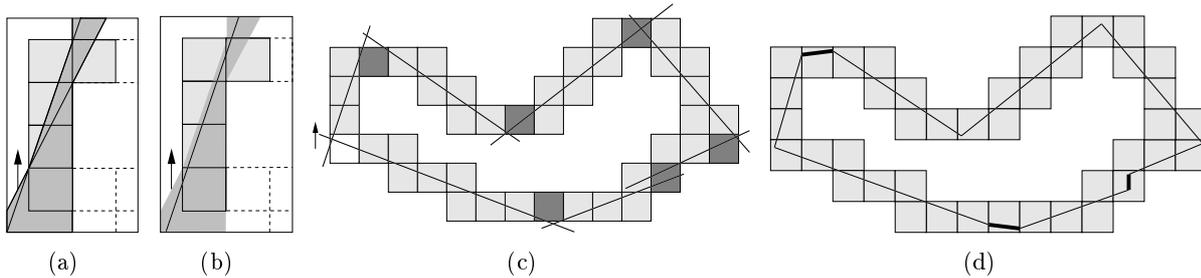


FIG. 3.4 – Exemple de reconstruction avec joints : (a) Ensemble des droites solutions (en gris foncé sur la figure) pour le segment discret reconnu, (b) Choix d'une droite solution, (c) Résultat du processus de reconnaissance, (d) Courbe polygonale obtenue après l'ajout de joints.

3.2.1.2 Méthode sans joints [SBDA05]

Une deuxième méthode, appelée méthode sans joints, a donc été proposée afin de pallier cet inconvénient. Le but de cette méthode était d'effectuer une reconstruction inversible d'une courbe discrète 1-connexe en ne parcourant qu'une seule fois la courbe, tout en essayant d'obtenir des segments de grande taille. La reconstruction de la courbe se fait donc en même temps que la reconnaissance des segments discrets, et la discrétisation Standard de chaque segment euclidien reconstruit doit être identique au segment discret. Autrement dit, chaque segment reconstruit doit couper tous les pixels du segment discret reconnu correspondant.

Le principe de cette méthode est de « guider » la reconnaissance de chaque segment discret de manière à ce que le choix de la droite euclidienne solution garantisse la réversibilité de la courbe reconstruite. Plus précisément, le but est de s'assurer qu'à chaque reconnaissance de

segment discret, il existe une droite solution qui coupe la droite solution choisie pour le segment précédemment reconnu à l'intérieur de leur pixel commun.

Pour cela, tout comme pour la méthode avec joints, un point discret de départ est choisi dans la courbe, ainsi qu'un sens de parcours. Un point euclidien est alors choisi à l'intérieur du pixel correspondant. La reconnaissance du premier segment se fait en ajoutant un à un les pixels consécutifs au pixel de départ et en calculant la préimage des points discrets. Mais contrairement à la méthode avec joints, la reconnaissance ne s'arrête pas lorsque la préimage est vide, mais lorsqu'il n'existe plus de droite solution qui passe par le point fixé (voir Figure 3.5a). Cette vérification est effectuée dans l'espace des paramètres en calculant l'intersection entre la droite duale du point fixé et la préimage. Si l'intersection est vide, il n'existe plus de droite solution vérifiant la condition souhaitée, et la reconnaissance s'arrête puisque le dernier point discret ajouté n'appartient pas au segment reconnu.

Une droite solution passant par le point fixé est alors choisie (voir Figure 3.5b). Dans l'espace des paramètres, ceci se traduit par le choix d'un point appartenant à l'intersection entre la droite duale du point fixé et la préimage calculée pour le segment reconnu. Enfin, un segment euclidien dont une extrémité est le point fixé est calculé pour le segment reconnu. La deuxième extrémité est calculée en faisant l'intersection entre la droite solution et le dernier pixel reconnu, et en choisissant un point sur cette intersection, comme par exemple le milieu du segment correspondant à l'intersection. La reconnaissance et la reconstruction des segments suivants s'effectuent de la même manière que pour le premier segment.

Un traitement particulier est cependant nécessaire pour reconstruire le dernier segment de la courbe dans le cas d'une courbe fermée. En effet, le dernier segment reconstruit doit pouvoir être raccordé au point fixé dans le premier pixel de la courbe. Il est donc nécessaire, lors de l'ajout du dernier point discret, de s'assurer que l'unique droite passant à la fois par les premier et dernier points fixés de la courbe est une droite solution pour le segment reconnu. Cette contrainte supplémentaire se traduit dans l'espace des paramètres par le fait que l'intersection entre les deux droites duales des points fixés doit se trouver à l'intérieur de la préimage calculée. Dans le cas contraire, le dernier point ajouté n'appartient pas au segment reconnu.

Cette deuxième méthode présente l'avantage de ne s'effectuer qu'en un seul parcours de la courbe discrète. L'avantage principal réside toutefois dans le fait qu'en introduisant le concept de "contrainte de sommet" (contraignant une droite euclidienne reconstruite à passer par un sommet fixé) cette méthode permet de gérer le problème des sommets de degrés supérieurs à 2 et donc de reconstruire des courbes non simples. Un autre avantage non négligeable est que cette méthode peut être étendue aux dimensions supérieures alors que l'extension des "joints" en dimension 3 est très difficile. Cependant, les résultats obtenus sont peu satisfaisants. En effet, tout d'abord, le nombre de segments reconstruits est en général nettement supérieur au nombre de segments obtenus avec la méthode avec joints. De plus, les courbes reconstruites sont visuellement peu satisfaisantes.

3.2.1.3 Bilan

Les deux méthodes proposées sont aisées à mettre en oeuvre et fournissent des résultats relativement convenables. Cependant, face au problème posé par le nombre important de segments générés par la méthode sans joint, nous avons essayé d'améliorer ce dernier algorithme. Pour cela, nous nous sommes intéressés au choix du point de départ. En effet, il nous a semblé que le fait de fixer un point à chaque reconstruction de segment euclidien était en grande partie responsable du nombre élevé de segments reconstruits, et ce car le nombre de droites solutions passant par un seul point est très faible. Ainsi, nous avons tenté de trouver une méthode moins

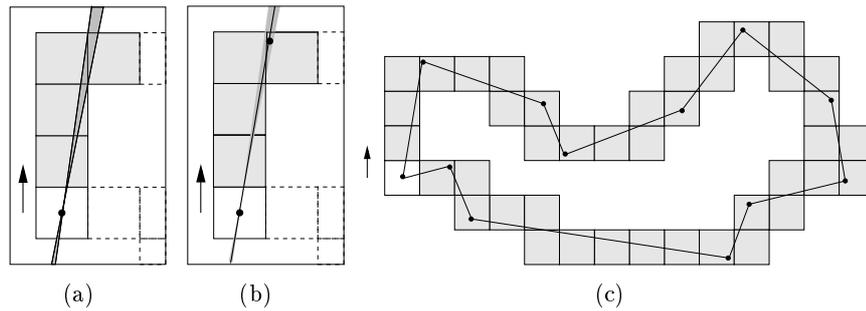


FIG. 3.5 – Exemple de reconstruction sans joints : (a) Ensemble des droites solutions contenant le point fixé (en gris foncé sur la figure), (b) Choix d’une droite solution et d’un nouveau point fixé, (c) Résultat.

contraignante que la méthode sans joints, permettant ainsi de diminuer le nombre de segments reconstruits tout en ne nécessitant qu’un seul traitement de la courbe. Deux approches ont alors été envisagées, que nous présentons dans la section suivante.

3.2.2 Méthode étendue

Dans cette section, nous présentons un nouvel algorithme de reconstruction inversible de 1-courbes discrètes basé sur l’algorithme de reconnaissance de droites Standard présenté dans le chapitre précédent.

Dans ce qui suit, nous donnons tout d’abord rapidement le principe de notre algorithme, puis nous détaillons les étapes de reconnaissance et de polygonalisation qui le composent. Enfin, nous comparons les résultats obtenus avec notre méthode avec ceux obtenus en utilisant les méthodes avec joint [BSDA03] et sans joint [SBDA05] présentées dans la section précédente.

3.2.2.1 Principe

Le principe de cette méthode est tout comme la méthode sans joints de contraindre la reconnaissance. Cependant, lors de la reconnaissance des segments discrets, non pas une solution mais un ensemble de solutions possibles pour le segment reconnu est calculé. Le choix d’une solution est alors effectué lors d’une deuxième étape.

Nous avons suivi le raisonnement suivant : au lieu de considérer uniquement un point fixé à l’intérieur du pixel de départ de la reconnaissance, considérons à présent l’ensemble des points du pixel. Ainsi, dans le cas du premier segment discret à reconnaître, la reconnaissance continue tant qu’il existe une droite solution coupant le pixel de départ. Il s’agit donc là de la méthode avec joints appliquée au premier segment à reconnaître. Dans les autres cas de reconnaissance de segment, l’ensemble de points considérés dans le pixel se restreint alors aux points atteints par les droites solutions calculées pour le segment précédemment reconnu.

La méthode que nous utilisons est donc composée de deux phases : une phase de reconnaissance contrainte et une phase de polygonalisation.

La phase de reconnaissance (voir Figures 3.6a et 3.6b) consiste à décomposer la courbe discrète en segments de droite discrète, tout en calculant pour chaque segment reconnu l’ensemble des droites coupant les pixels du segment. Cependant, l’algorithme de reconnaissance est contraint car nous souhaitons que les droites coupant les pixels coupent aussi un ensemble de points

particuliers situés dans le premier pixel du segment à reconnaître. La nature de ces points, ainsi que l'algorithme de reconnaissance contraint sont détaillés dans la section 3.2.2.2.

Lorsque la phase de reconnaissance est terminée, la phase de polygonalisation est effectuée (voir Figure 3.6c). La méthode utilisée pour reconstruire la ligne polygonale est la suivante : l'algorithme commence à partir du dernier pixel reconnu de la courbe. Un point euclidien est choisi dans ce pixel parmi l'ensemble des points coupés par les droites euclidiennes qui ont été calculées pour ce segment durant la phase de reconnaissance. Puis une droite contenant ce point est choisie parmi les droites solutions, et un nouveau point est fixé dans l'autre pixel extrémité du segment reconnu. Le processus est alors répété jusqu'à atteindre le premier pixel de la courbe. Cette méthode est décrite plus en détails dans la section 3.2.2.3.

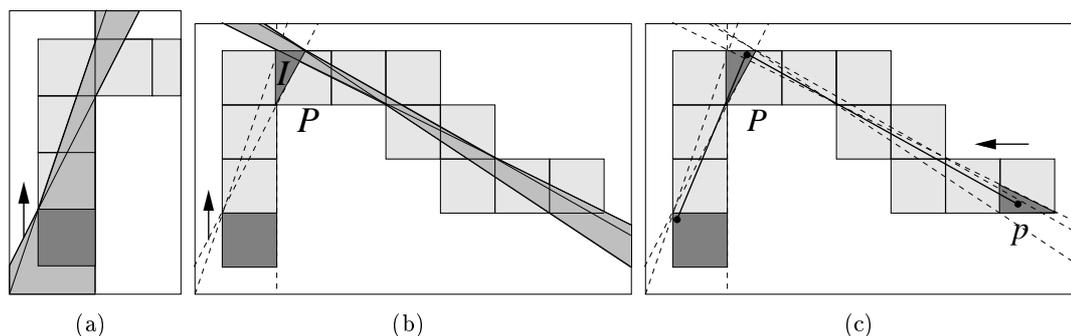


FIG. 3.6 – Principe de notre méthode de reconstruction. (a) Reconnaissance du premier segment. Le pixel de départ est en gris foncé. La flèche noire indique la direction de reconnaissance. (b) Reconnaissance des segments suivants. (c) Étape de polygonalisation. La flèche noire indique la direction de polygonalisation.

Pixel de départ et sens de reconnaissance

Nous donnons ici les choix effectués concernant le pixel de départ et le sens de parcours de la courbe discrète. Dans ce travail, nous utilisons les deux conventions proposées pour les courbes ouvertes et fermées par I. SIVIGNON ET AL. dans [SBDA05] :

- Pour une courbe simple ouverte : le point discret de départ est l'extrémité de la courbe ayant la plus petite abscisse et la plus petite ordonnée. Le sens de reconnaissance est alors induit par le choix du point de départ.
- Pour une courbe simple fermée : le point de départ est le point de la courbe ayant la plus petite abscisse et la plus petite ordonnée. Le sens de reconnaissance est le sens des aiguilles d'une montre.

Notons que ces choix nous assurent que deux courbes discrètes identiques à une translation près, seront reconstruites de manière identique. Par contre, ce ne sera pas le cas de deux courbes identiques à une rotation près.

De plus, nous n'avons pas dans ces travaux fait d'étude poussée concernant ces choix et leur influence sur les courbes polygonales obtenues. Nous avons simplement constaté, suite à un jeu de tests, que le nombre de segments reconnus varie fortement en fonction du sens de parcours adopté. Nous n'avons cependant pas trouvé de méthode permettant de déterminer la direction de reconnaissance la plus appropriée pour une courbe donnée, i.e. pour laquelle le nombre de segments reconnus est moindre.

3.2.2.2 Reconnaissance des segments discrets

Nous détaillons ici la méthode utilisée pour effectuer la première étape de notre algorithme. Le but de celle-ci est d'effectuer la reconnaissance de segments discrets sur une courbe en tenant compte du segment précédemment reconnu. Pour cela, nous devons nous assurer qu'au moins une des droites se trouvant dans l'ensemble des droites solutions du segment courant possède un point d'intersection avec au moins une droite solution du segment précédent, et que ce point se trouve dans le pixel commun au deux segments discrets.

Cependant, au lieu de considérer l'ensemble des droites solutions calculées pour le segment discret précédemment reconnu, nous considérons l'ensemble des points euclidiens atteints par les droites solutions dans le dernier pixel reconnu. Nous appelons cet ensemble de points le *polygone solution* du segment reconnu. Ainsi, lors de la reconnaissance suivante, le but est de vérifier qu'il existe au moins une droite solution qui coupe le polygone ainsi obtenu.

Notons de plus que pour chaque segment reconnu, un ensemble de points est aussi atteint dans le premier pixel reconnu par l'ensemble des droites solutions. Nous appelons cet ensemble le *polygone initial* du segment. Notons que ces deux polygones sont des polygones convexes (nous ne démontrerons pas cette propriété ici).

Dans l'espace des paramètres, cette contrainte se traduit par le fait que l'intersection entre la préimage généralisée du segment en cours de reconnaissance et le dual du polygone solution est non vide.

Ainsi, si nous considérons une 1-courbe discrète C constituée de k pixels P_1, \dots, P_k ordonnés, $k \in \mathbb{N}^*$, l'algorithme se déroule de la manière suivante (voir Algorithme 2). Nous commençons par reconnaître le premier segment S de C en considérant comme polygone solution le premier pixel (voir Figure 3.7a). Ensuite, nous ajoutons les pixels suivants jusqu'à ce que l'ensemble des droites solutions de S devienne vide. Dans ce cas, l'ajout du dernier pixel implique que S n'est plus un segment discret Standard. Nous conservons alors l'ensemble des droites solutions calculé avant cet ajout et nous démarrons une nouvelle reconnaissance à partir du pixel précédent.

Pour effectuer la reconnaissance du segment suivant, nous commençons par calculer le dual du polygone solution du dernier segment reconnu. Le polygone obtenu est alors utilisé pour initialiser l'ensemble des droites solutions du nouveau segment (voir Figure 3.7b). La reconnaissance fonctionne ensuite de la même manière que précédemment et s'arrête lorsque tous les segments ont été reconnus, i.e. lorsque tous les pixels de la courbe C ont été traités (voir Figure 3.7c).

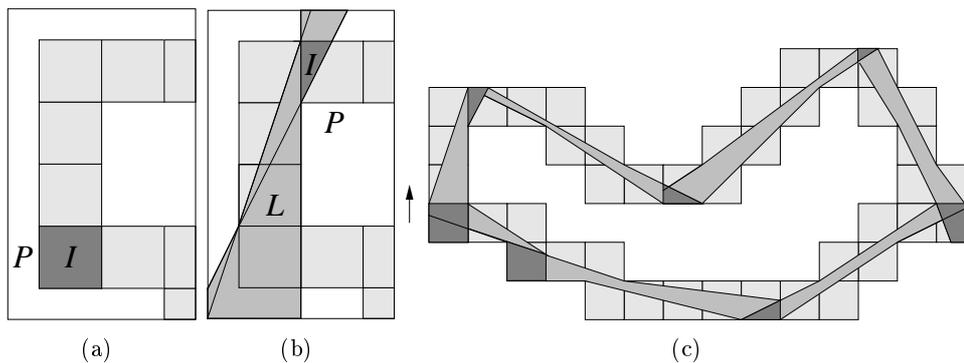


FIG. 3.7 – Illustration du processus de reconnaissance. (a) Contrainte de la reconnaissance du premier segment (I sur la figure). Pixel de départ (P). (b) Contrainte de la reconnaissance du deuxième segment. (c) Ensemble de droites euclidiennes résultant du processus de reconnaissance de courbe discrète.

Algorithme 2 : Reconnaissance de segments de droites discrètes.

Données : Un ensemble \mathcal{P} de n pixels ordonnés P_1, \dots, P_n .

```

début
   $i \leftarrow 1$ ;
   $I \leftarrow P_1$ ;
  tant que  $i \leq n$  faire
     $S \leftarrow \{P_i\}$ ;
     $\mathbb{G}_P \leftarrow Dual(I)$ ;
    tant que  $\mathbb{G}_P \neq \emptyset$  et  $i < n$  faire
       $i \leftarrow i + 1$ ;
       $S \leftarrow S \cup \{P_i\}$ ;
       $\mathbb{G}_{P_{temp}} \leftarrow \mathbb{G}_P$ ;
       $\mathbb{G}_P \leftarrow \mathbb{G}_P \cap Dual(P_i)$ ;
    si  $\mathbb{G}_P = \emptyset$  alors
       $\mathbb{G}_P \leftarrow \mathbb{G}_{P_{temp}}$ ;
       $S \leftarrow S - \{P_i\}$ ;
       $i \leftarrow i - 1$ ;
     $I \leftarrow Dual(\mathbb{G}_P) \cap P_i$ ;
fin
  
```

3.2.2.3 Polygonalisation de la courbe

Cette étape correspond au choix d'une reconstruction parmi les différentes courbes polygonales possibles fournies par l'étape de reconnaissance. Nous décomposons cette étape en 3 parties : la reconstruction du premier segment de la courbe, la reconstruction d'un segment « intermédiaire » et enfin, la reconstruction du dernier segment.

Reconstruction du premier segment euclidien : un premier point euclidien correspondant à un premier sommet de la courbe polygonale devant être obtenue est tout d'abord fixé dans le dernier pixel reconnu. En effet, la reconstruction de la courbe est effectuée dans le sens inverse de la reconnaissance. Notons que dans le cas d'une courbe ouverte, le dernier pixel reconnu est le pixel extrémité qui n'est pas le point de départ, alors que dans le cas d'une courbe fermée, il s'agit du pixel de départ.

Plus précisément, dans le cas d'une courbe ouverte, le point est choisi à l'intérieur du dernier polygone solution calculé. La position de ce point est choisie arbitrairement, comme par exemple le barycentre du polygone solution, celui-ci étant convexe. Afin de choisir simplement ce point et de garantir la réversibilité de la reconstruction pour le modèle Standard, il est cependant préférable de ne pas choisir ce point sur le bord du polygone solution. Dans le cas d'une courbe fermée, le point est choisi à l'intérieur de l'intersection entre le polygone solution du dernier segment reconnu et le polygone initial du premier segment reconnu si l'intersection n'est pas vide. Notons que cette intersection est de même convexe, ce qui nous autorise à choisir son barycentre pour premier sommet de la courbe.

Ensuite, une droite passant par le point fixé est choisie dans l'ensemble des droites solutions du dernier segment reconnu. Ceci est effectué en calculant l'intersection entre la préimage généralisée du segment reconnu et la droite duale du point fixé, puisque nous savons que le dual de n'importe quel point sur cette intersection est une droite passant par le sommet fixé. Un point est alors choisi sur cette intersection qui nous fournit la droite solution. Notons qu'en fonction de la forme de la préimage généralisée, l'intersection entre une droite et la préimage peut être soit un segment,

soit une demi-droite, soit deux demi-droites. Un traitement adapté est alors nécessaire afin de choisir un point sur cette intersection.

Enfin, un nouveau point euclidien est fixé dans l'intersection entre le polygone initial du segment reconnu et la droite choisie. Nous avons choisi ici le centre du segment résultant de l'intersection.

Reconstruction des segments intermédiaires : la reconstruction des autres segments, hormis le dernier segment, se fait simplement en répétant les processus de choix d'une droite solution et de choix d'un point sur cette droite.

Reconstruction du dernier segment : dans le cas d'une courbe ouverte, le dernier segment est reconstruit de la même manière que les segments intermédiaires. Par contre, dans le cas d'une courbe fermée, la reconstruction de ce segment se fait en vérifiant si la droite passant par le premier et le dernier sommets fixés fait partie des droites solutions du premier segment reconnu. Ceci est effectué dans l'espace des paramètres en vérifiant que le point résultant de l'intersection entre les deux droites duales des deux points se trouve bien à l'intérieur de la préimage généralisée du segment. Si c'est le cas, le dernier segment peut être reconstruit entre ces deux points. Sinon, le segment est reconstruit comme un segment intermédiaire, et un deuxième segment doit être ajouté entre le dernier et le premier sommet calculé.

La polygonalisation de la courbe C (voir Algorithme 3) est effectuée en partant du dernier pixel reconnu P de C dans le sens inverse du sens de reconnaissance (voir Figure 3.8). Notons que si C est une courbe fermée alors P est le pixel de départ de la reconnaissance. Un premier point p est choisi à l'intérieur de l'intersection I entre P et l'ensemble des droites $L = Dual(\mathbb{G}_P(S))$ calculé pour le dernier segment discret reconnu S . Si C est fermée alors p est choisi dans l'intersection I' entre I et l'ensemble de droites calculé pour le premier segment reconnu (si I' n'est pas vide). Puisque I (resp. I') est un polygone convexe, nous pouvons par exemple choisir le barycentre de I (resp. I') comme point p . Puis, une droite l contenant p est choisie dans L . Ceci est effectué dans l'espace de paramètres de la façon suivante : l'intersection entre $Dual(L) = \mathbb{G}_P(S)$ et $Dual(p)$ est calculée. Le dual de chaque point dans cette intersection est une droite contenant p . Un point est donc choisi dans cette intersection. Enfin, l'intersection entre l et le premier pixel reconnu de S est effectuée, et le milieu du segment obtenu est choisi comme le nouveau premier point de la reconstruction du segment suivant. Le processus est répété jusqu'à ce que le point de départ de la courbe soit atteint.

Dans le cas d'une courbe fermée, la reconstruction du dernier segment est doublement contrainte, puisque le premier point fixé se trouve à l'intérieur du pixel de départ. Soit p ce point et p_f le premier point fixé du premier segment reconnu. S'il existe une droite contenant p et p_f , le dernier segment est reconstruit. Dans le cas contraire, un autre point p_l est fixé dans le pixel de départ et un segment est ajouté entre p_l et p . La figure 3.8 montre un exemple de courbe reconstruite.

3.2.3 Résultats

Dans cette section, nous présentons tout d'abord les résultats que nous avons obtenus avec notre méthode. Puis nous les comparons avec les méthodes avec et sans joint.

3.2.3.1 Méthode étendue

Les résultats obtenus avec la méthode étendue sont visuellement de très bonne qualité.

Nous pouvons voir sur la figure 3.9 un exemple de reconstruction de courbe discrète dans le cas d'une image. La courbe reconstruite est constituée des points discrets situés sur la frontière

Algorithme 3 : Polygonalisation de la courbe

Données : Un ensemble S de k segments reconnus connexes S_1, \dots, S_k avec $S_i = (P_{i_f}, \dots, P_{i_l})$ et k ensembles de droites solutions L_1, \dots, L_k .

début

$i \leftarrow k$;

 Choisir un point réel p_{i_f} in P_{i_f} ;

$p_f \leftarrow p_{i_f}$;

tant que $i > 1$ **faire**

 Choisir une droite l dans L_i contenant p_{i_f} ;

$s \leftarrow l \cap P_{i_f}$;

 Choisir un point réel p_{i_l} sur s ;

 Ajouter un segment entre p_{i_f} et p_{i_l} ;

$p_{i-1_f} \leftarrow p_{i_l}$;

$i \leftarrow i - 1$;

 Chercher une droite l dans L_1 qui contient p_{1_f} et p_f ;

si l *n'existe pas* **alors**

 Choisir une droite l dans L_1 qui contient p_{1_f} ;

$s \leftarrow l \cap P_{1_f}$;

 Choisir un nouveau point réel p_{1_l} sur s ;

 Ajouter un segment entre p_{1_f} et p_{1_l} et entre p_{1_l} et p_f ;

sinon

 Ajouter un segment entre p_{1_f} et p_f ;

fin

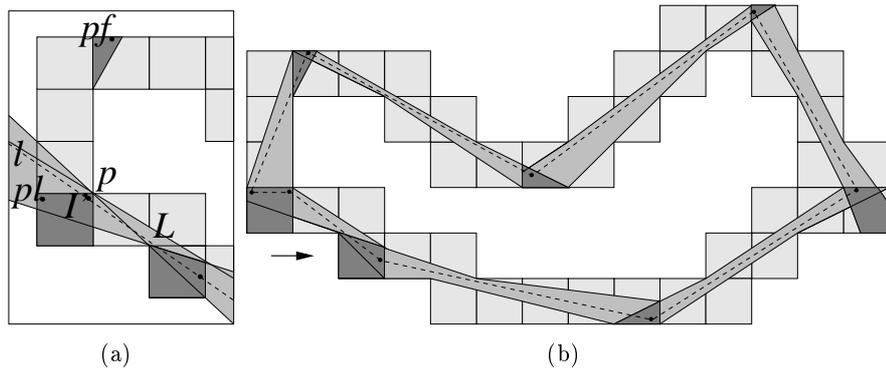


FIG. 3.8 – Résultat du processus de polygonalisation (en pointillés sur la figure). Le sens de reconstruction est indiqué par la flèche noire.

entre les deux régions colorées en noir et en blanc. L'espace discret associé à l'image est donc l'espace discret classique dual. Nous pouvons vérifier sur la figure 3.9c que la courbe reconstruite, correspondant à la frontière entre les deux régions de la figure 3.9b coupe bien l'ensemble des pixels associés au points de la courbe discrète (voir Figure 3.9a).

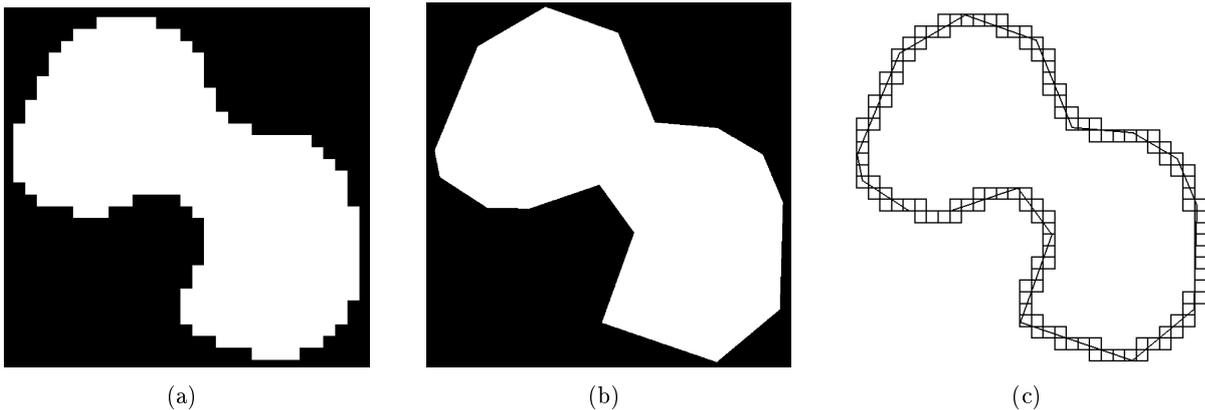


FIG. 3.9 – Résultat du processus de reconstruction : (a) Image originale, (b) Objet obtenu, (c) La courbe polygonale obtenue coupe bien tous les pixels de la courbe discrète.

D'autres exemples de reconstructions de courbes discrètes obtenues avec la méthode étendue sont montrés sur les figures 3.10, 3.11, 3.12 et 3.13.

3.2.3.2 Comparaison avec les méthodes avec et sans joints

Le tableau 3.1 montre quelques résultats obtenus par les trois différentes méthodes de reconstruction (avec joints, sans joint et étendue) en termes de nombre de segments de droite obtenus. Nous pouvons voir que notre méthode fournit un nombre inférieur de segments que les méthodes avec et sans joint.

3.2.4 Reconstruction de courbes non simples

Dans cette section, nous proposons une méthode permettant la reconstruction de courbes non simples. Le principe est le suivant : tout d'abord, un point euclidien est fixé à l'intérieur

TAB. 3.1 – Comparaison entre les trois méthodes de reconstruction. Les nombres de segments des courbes reconstruites sont donnés.

Image	Nombre de points discrets de la frontière entre régions	Méthode avec joints	Méthode sans joint	Méthode étendue
	116	22	24	13
	356	50	62	39
	424	57	66	38
	574	71	104	57
	852	99	130	84

de chaque pixel de la courbe de *degré*⁸ supérieur ou égal à 3. Comme point fixé, nous avons arbitrairement choisi le point discret associé à ces pixels. Ensuite, toutes les courbes discrètes comprises entre deux points fixés sont reconstruites en utilisant la méthode étendue, en prenant en compte la contrainte matérialisée par les points fixés aux deux extrémités des courbes. Cela revient à reconstruire une courbe discrète ouverte contrainte par un point à ses deux extrémités. Nous appelons ces courbes des courbes *doublement contraintes*. Cette contrainte intervient lors de la reconnaissance des premier et dernier segments de la courbe de la manière suivante :

- Reconnaissance du premier segment discret : la préimage généralisée des pixels du segment est calculée tant que son intersection avec la droite duale du point fixé n'est pas vide. Lorsque celle-ci est vide, la dernière intersection non vide entre la préimage généralisée calculée et la droite duale correspond aux nouvelles droites solutions permettant le calcul d'un nouveau polygone solution pour débiter la reconnaissance suivante.
- Reconnaissance du dernier pixel de la courbe : lors de l'ajout du dernier pixel, la reconnaissance du dernier segment discret de la courbe est valide si et seulement si l'intersection entre la préimage généralisée, le dual du polygone solution et la droite duale du point fixé est non vide. Si ce n'est pas le cas, la reconnaissance du segment s'arrête sur le pixel précédant le dernier pixel de la courbe.

La reconstruction d'une courbe non simple se fait alors en parcourant les différentes courbes doublement contraintes constituant la courbe non simple. Un point de départ de la reconnaissance et un sens de parcours de la courbe ont donc été choisis.

- Le pixel de départ est le pixel de degré supérieur ou égal à trois dont le point discret associé est d'abscisse le plus petit et d'ordonnée la plus petite.

⁸Nous appelons degré d'un pixel P le nombre de pixels 1-voisins de P .

- Le parcours de la courbe se fait par propagation à l'intérieur de la courbe à partir du pixel de départ (parcours en largeur de la courbe). Les courbes à reconstruire sont considérées dans l'ordre suivant : tout d'abord, la courbe située à droite, puis en haut, puis à gauche, puis en bas.

Les choix effectués pour le pixel de départ et la direction de propagation sont bien entendus arbitraires. Le pixel de départ doit tout de même être un point de degré supérieur ou égal à trois pour simplifier la reconstruction.

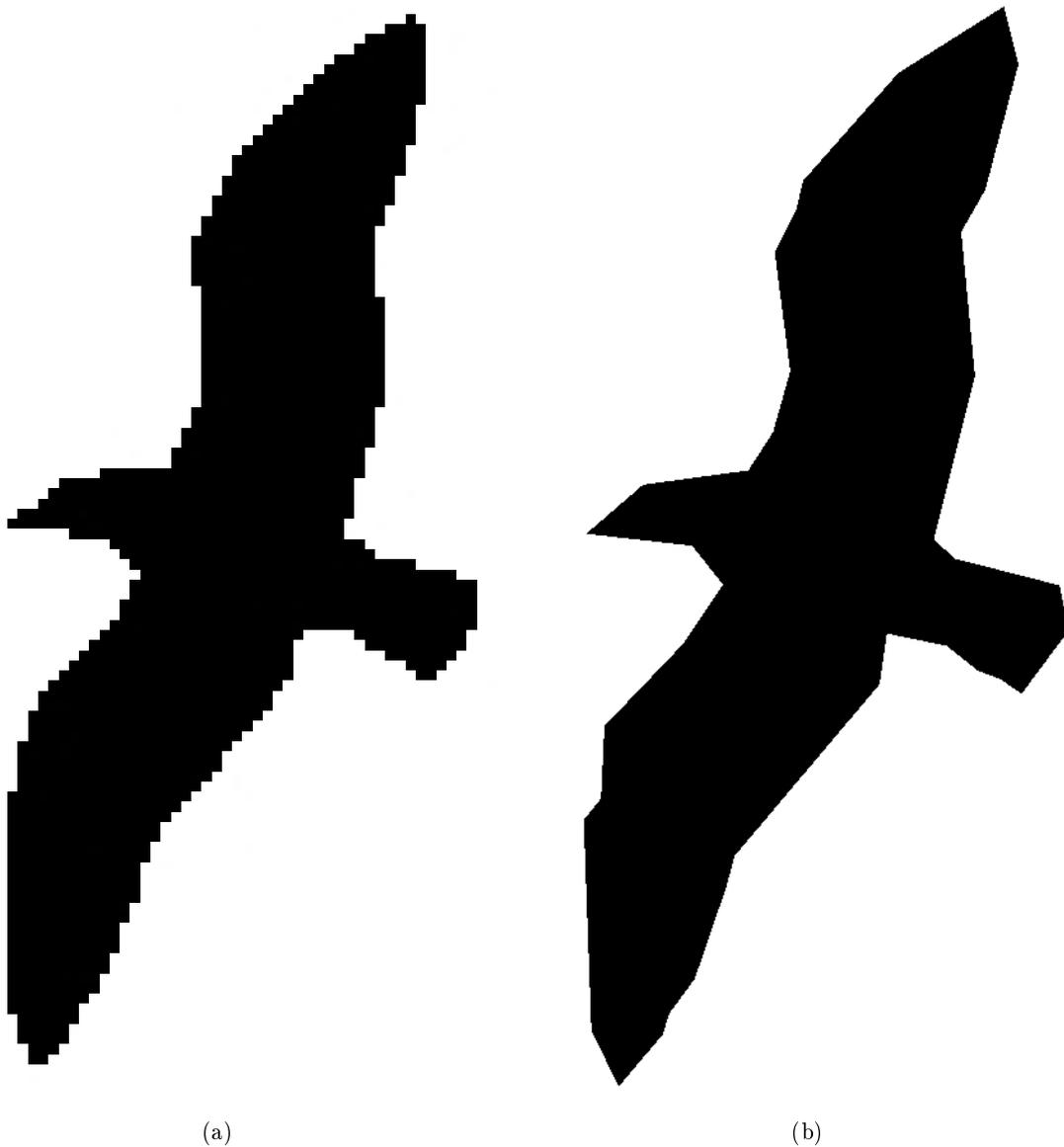


FIG. 3.10 – Exemple de courbe discrète reconstruite avec la méthode étendue : (a) Image d'origine : la frontière entre les régions noire et blanche est constituée de 178 points discrets, (b) Image résultante : la frontière entre les régions noire et blanche est constituée de 38 segments.

3.2.5 Conclusion et perspectives

Dans cette section, nous nous sommes intéressés au problème de la reconstruction inversible de 1-courbes discrètes classiques. Notre but était de fournir une reconstruction de courbe discrète qui soit en particulier inversible pour le modèle Standard. Nous avons tout d'abord brièvement rappelé les principes des deux algorithmes de reconstruction proposés par R. BRETON[Bre03] dans sa thèse. Nous avons ensuite présenté une nouvelle méthode de reconstruction, appelée *méthode étendue*, qui améliore les deux méthodes précédemment évoquées. En effet, les courbes polygonales obtenues à l'aide de notre algorithme contiennent en moyenne moins de segments que les courbes obtenues à l'aide des deux autres méthodes. Enfin, nous avons rapidement abordé le problème de la reconstruction de courbes non simples en proposant une méthode résolvant ce problème. Cette méthode est notamment utile pour reconstruire les frontières entre régions dans une image.

La reconstruction de la courbe est de plus effectuée au plus loin. Enfin, la reconstruction est effectuée en deux étapes. Cependant, l'ensemble des solutions possibles pour la reconstruction est calculé durant le premier parcours de la courbe, le deuxième parcours consistant simplement à choisir une droite parmi les droites solutions calculées durant le premier parcours.

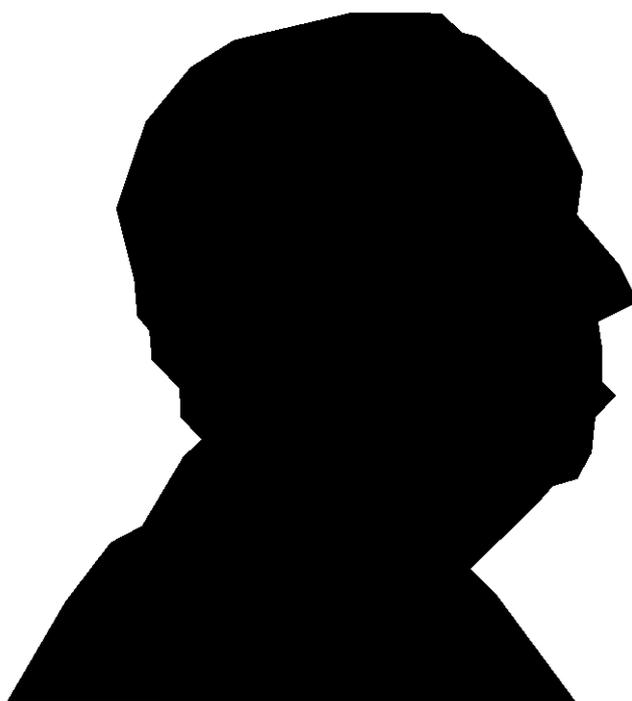
Une des principales perspectives de ce travail est l'extension de cette méthode à d'autres types d'espaces, réguliers ou non. En effet, l'étape de reconnaissance de notre méthode est principalement basée sur le calcul de la préimage généralisée. Cette dernière étant définie pour n'importe quel type de pavé, notre méthode de reconstruction peut être aisément appliquée à la reconstruction de 1-courbes dans des espaces discrets quelconques, dès lors que le problème posé est la détermination d'une courbe polygonale coupant l'ensemble des pavés de la courbe discrète.

Une deuxième évolution de notre algorithme est l'étude de son extension à la reconstruction inversible de volumes discrets, et notamment inversible pour le modèle Standard. Nous verrons dans le chapitre suivant que ce problème est loin d'être trivial, mais qu'il mérite une attention particulière.

Cet algorithme de reconstruction a en particulier fait l'objet d'une publication [DA06c].

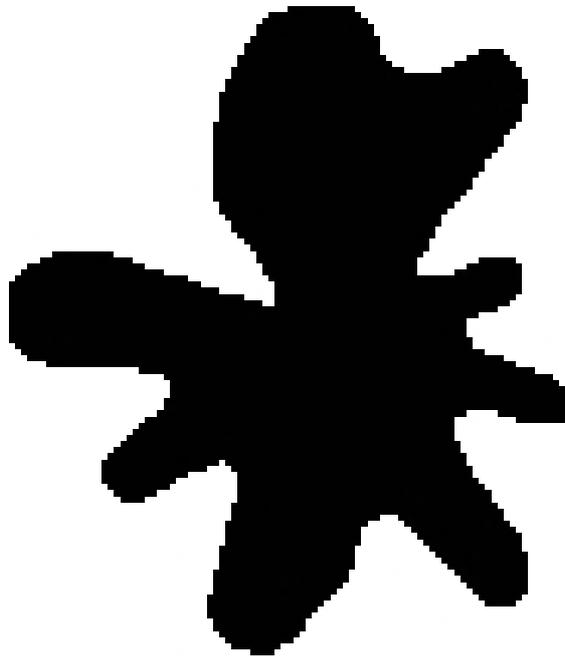


(a)

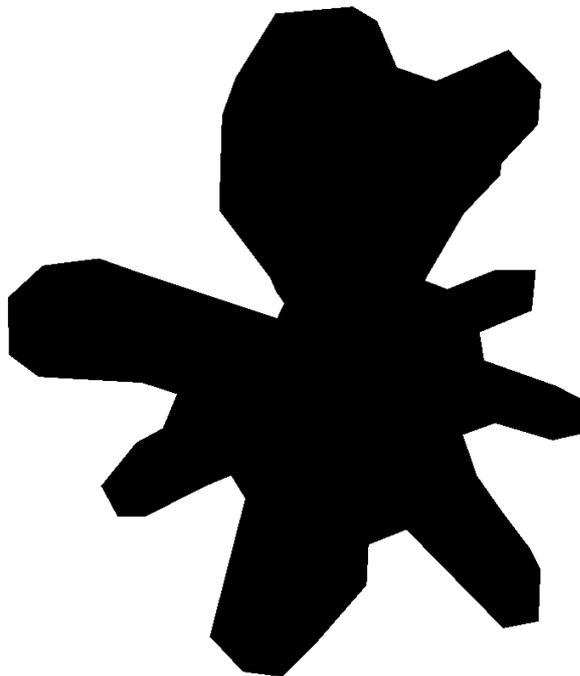


(b)

FIG. 3.11 – Exemple de courbe discrète reconstruite avec la méthode étendue : (a) Image d'origine : la frontière entre les régions noire et blanche est constituée de 212 points discrets, (b) Image résultante : la frontière entre les régions noire et blanche est constituée de 38 segments.

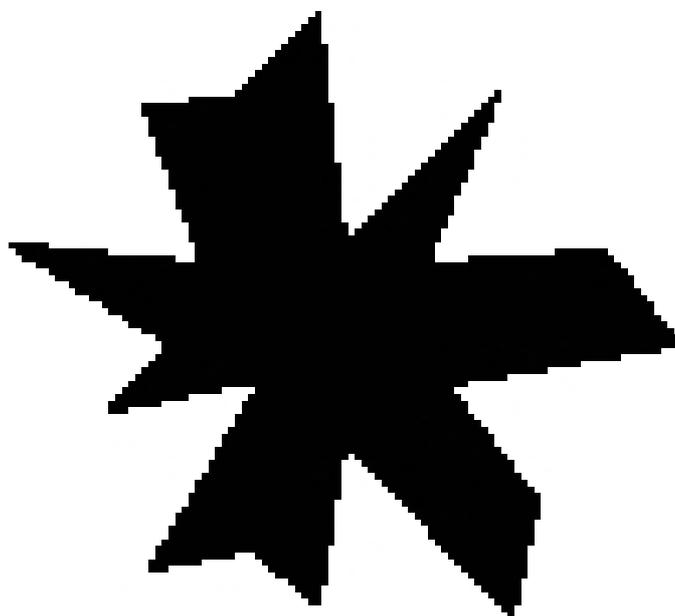


(a)

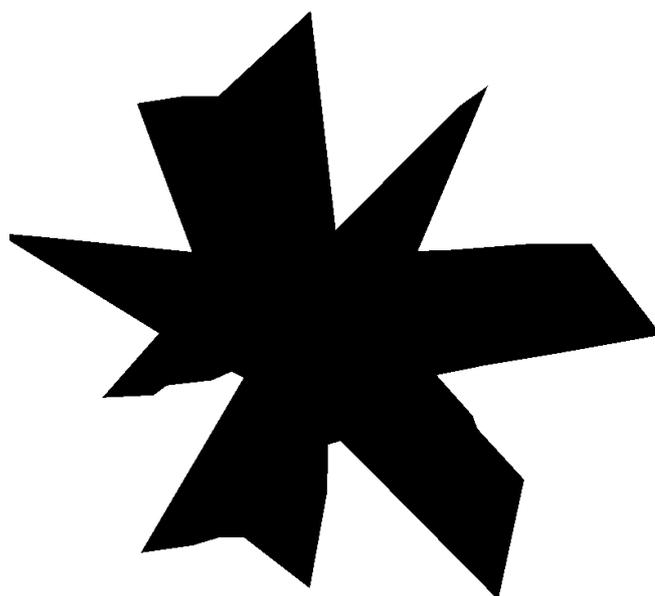


(b)

FIG. 3.12 – Exemple de courbe discrète reconstruite avec la méthode étendue : (a) Image d'origine : la frontière entre les régions noire et blanche est constituée de 302 points discrets, (b) Image résultante : la frontière entre les régions noire et blanche est constituée de 64 segments.



(a)



(b)

FIG. 3.13 – Exemple de courbe discrète reconstruite avec la méthode étendue : (a) Image d'origine : la frontière entre les régions noire et blanche est constituée de 305 points discrets, (b) Image résultante : la frontière entre les régions noire et blanche est constituée de 39 segments.

3.3 Reconstruction de courbes et volumes discrets en dimension 3

Dans cette section, nous nous intéressons au problème de la reconstruction analytique du bord de volumes discrets 2-connexes en dimension 3. Le problème posé est le suivant : étant donné un volume discret en dimension 3, comment obtenir une représentation polygonale de son bord ?

Dans nos travaux, nous cherchons en particulier à obtenir une reconstruction inversible pour le modèle Standard, c'est à dire que la discrétisation de l'objet reconstruit suivant le modèle Standard est identique à l'objet discret initial. De plus, nous souhaitons obtenir un objet qui soit topologiquement cohérent avec l'objet discret d'origine.

Parmi les méthodes de reconstruction analytique de volumes discrets qui ont été proposées dans la communauté de géométrie discrète [BF94, Pap99, SDC05, KS01], nous nous sommes plus particulièrement intéressés aux méthodes proposées par I. SIVIGNON dans sa thèse [Siv04]. En effet, hormis une méthode basée sur un algorithme de simplification d'objets de type Marching Cubes [CGS04], elle a proposé deux méthodes de reconstruction analytique utilisant l'algorithme de reconnaissance de plans discrets Standard basé sur le calcul de la préimage classique :

- La première méthode consiste tout d'abord à segmenter la surface de l'objet discret en face discrètes. Puis, un plan solution est choisi dans la préimage calculée pour chaque face reconnue. Enfin, l'intersection entre les différents plans euclidiens définit un polyèdre correspondant à une reconstruction possible de l'objet discret. L'inconvénient de cette méthode est que la surface obtenue n'est pas nécessairement inversible puisque les plans euclidiens choisis peuvent se couper en dehors de l'objet discret.

- Le principe de la deuxième méthode [SDC05] est tout d'abord de segmenter la surface du volume discret en morceaux de plans discrets Standard. Puis chaque morceau discret obtenu est reconstruit en une face polygonale dont la discrétisation standard est identique à la face discrète, en utilisant un algorithme de polygonalisation de courbes planaires en dimension 3.

Contrairement à la première méthode proposée, cette méthode fournit bien une reconstruction inversible. Par contre, la reconstruction obtenue n'est pas topologiquement correcte puisque les différentes facettes polygonales ne sont pas connexes.

Dans cette section, nous présentons un algorithme de reconstruction du bord d'un volume discret 2-connexe basé sur la méthode de reconnaissance de plans Standard que nous avons présentée dans le chapitre 2 de ce mémoire. Le résultat obtenu est inversible pour le modèle Standard et topologiquement cohérent avec l'objet discret de départ. La méthode que nous avons utilisée est fortement inspirée de l'algorithme de reconstruction "sans joint" présenté dans le chapitre précédent, ainsi que des algorithmes de reconstruction proposés par I. SIVIGNON dans sa thèse.

Dans la suite, nous décrivons tout d'abord les volumes discrets que nous considérons ce travail, ainsi que le principe de notre méthode. Puis nous détaillons les deux étapes nécessaires à la reconstruction que sont la reconnaissance de plans et la polygonalisation de face discrète. Enfin, nous présentons quelques résultats obtenus avec notre méthode.

3.3.1 Données traitées

Notre algorithme s'applique au bord B_V (voir Figure 3.14b) d'un volume discret 2-connexe V (voir Figure 3.14a). Nous supposons que chaque sommet de voxel qui se trouve sur B_V est un point discret. Quatre points discrets incidents à une même face de voxel forment un *surfel* (voir Figure 3.14c).

Notre but est de reconstruire une surface polygonale dont la discrétisation suivant le modèle Standard est égale à tous les points discrets de B_V . En d'autres termes, si nous considérons chaque point discret de B_V comme le centre d'un nouveau voxel (voir Figure 3.14d), la surface obtenue doit couper chaque voxel de ce nouveau volume creux 2-connexe (voir Figure 3.14e et 3.14f). Nous appelons ce volume le *contour en voxels* du volume discret V . Dans la suite, le terme surfel référencera arbitrairement soit quatre points discrets, soit les quatre voxels associés.

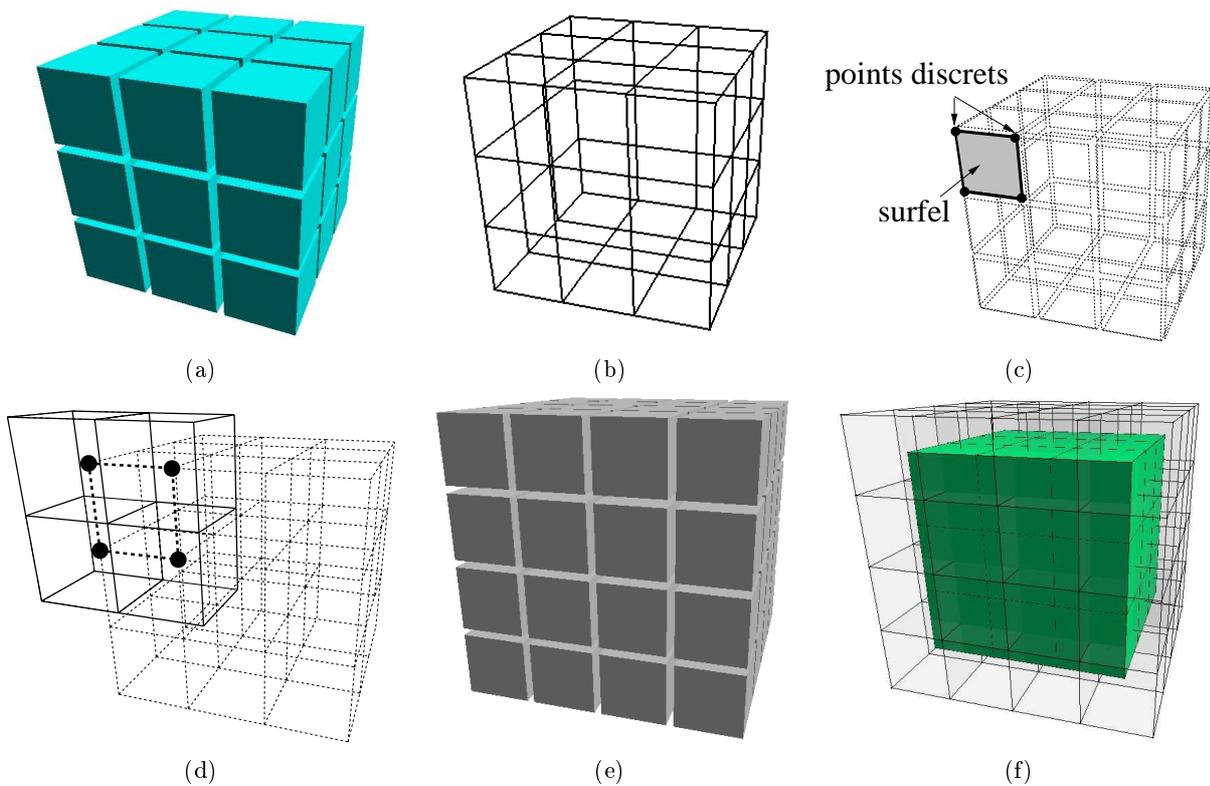


FIG. 3.14 – Données traitées : (a) Objet discret d'origine V . (b) et (c) Bord B_V de V composé de points discrets. (d), (e) et (f) Objets discrets équivalents à B_V . L'objet euclidien obtenu doit couper chaque voxel de (e).

3.3.2 Principe

Étant donné un contour de voxels V_c (voir Section 3.3.1), le but est d'obtenir une surface polygonale dont la discrétisation Standard est égale à V_c . Notre algorithme fonctionne de la manière suivante (voir Algorithme 4) : tout d'abord, un surfel S est choisi dans V_c et un premier morceau de plan discret Standard est reconnu en partant de S (voir Figure 3.15a) en utilisant l'algorithme de reconnaissance décrit dans le chapitre 2. Notons que nous ne choisissons pas seulement un voxel de départ pour effectuer la reconnaissance, mais les quatre voxels constituant

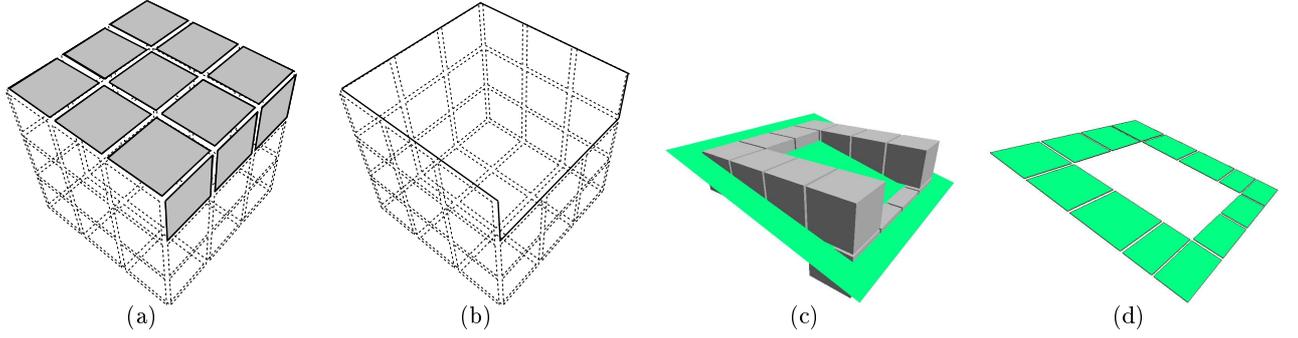


FIG. 3.15 – Illustration de la méthode de reconstruction : (a) Surfels reconnus, (b) Extraction du bord de la courbe discrète (en gras), (c) Une courbe discrète 2-connexes planaire C et le plan euclidien choisi P , (d) Résultat de l'intersection entre P et C .

S . En effet, nous voulons que chaque voxel du bord du plan reconnu ait au moins deux 2-voisins. Ainsi, les voxels doivent être reconnus quatre par quatre au lieu d'un par un. Quand le processus de reconnaissance s'arrête, un plan euclidien P est choisi dans l'ensemble des solutions calculé et le bord C du plan reconnu, qui est une courbe discrète 2-connexes est extrait (voir Figure 3.15b). Le processus de polygonalisation de face est alors appliqué sur C afin d'obtenir le contour d'une face euclidienne contenue dans P et dont la discrétisation Standard est identique au morceau de plan discret reconnu (voir Figure 3.15c).

Un surfel non traité est alors choisi et les processus de reconnaissance et de polygonalisation appliqués à nouveau. Ces deux processus sont réitérés jusqu'à ce que tous les surfels soient traités. Cependant, le processus de reconnaissance peut être contraint. En effet, pendant cette étape, le prochain surfel C_s à être reconnu peut être adjacent à des faces discrètes déjà reconstruites. Ainsi, les voxels de C_s peuvent être coupés par une face ou plus, et donc appartiennent à la discrétisation Standard du sommet ou de l'arête d'une face déjà reconstruite. Dans le but d'effectuer une reconstruction topologiquement correcte, notre but est d'obtenir une face polygonale qui contient ces sommets ou arêtes. Pour assurer cette propriété, des *contraintes de sommets* (un voxel de C_s est la discrétisation d'un sommet d'une face reconstruite) et des *contraintes d'arêtes* (un voxel de C_s appartient à la discrétisation d'une arête de face) doivent être prises en compte durant le processus de reconnaissance. Cela signifie qu'un surfel est reconnu si et seulement si ses contraintes n'amènent pas à un ensemble de solutions vide, i.e. il existe au moins un plan euclidien coupant tous les surfels reconnus et contenant toutes leurs contraintes. De plus, nous voulons qu'un voxel soit contraint par au plus un sommet ou une arête.

Les processus de reconnaissance contrainte, et de polygonalisation sont détaillés dans les sections 3.3.3 et 3.3.4.

Afin d'assurer l'unicité de la reconstruction, le choix d'un surfel de départ, ainsi qu'un ordre de reconnaissance des surfels pour chaque face est nécessaire. Différents choix possibles sont discutés dans les deux sections suivantes.

3.3.2.1 Surfel de départ et méthodes de propagation

Le surfel de départ S de la première reconnaissance de plan est l'unique surfel parallèle au plan $(0, X_1, X_2)$ contenant l'unique point discret $p = (x_p, y_p, z_p)$ vérifiant (voir Figure 3.16a) :

$$\forall V = (x_V, y_V, z_V) \in V_c - \{p\}, x_p < x_V$$

Algorithme 4 : Reconstruction analytique Standard

en entrée : Un contour de voxels V_c .
en sortie : Un maillage polygonal dont la discrétisation Standard est égale à V_c .
 S : Surfel ;
 P : Plan ;
 C : Courbe discrète 2-connexe ;
début
 tant que *il existe des surfels non traités* **faire**
 Choisir un surfel de départ S ;
 $(C, P) \leftarrow \text{RECONNAÎTRE_PLAN}(V_c, S)$;
 $\text{POLYGONALISER_COURBE}(C, P)$;
fin

ou

$$x_p = x_V \text{ et } y_p < y_V$$

ou

$$x_p = x_V, y_p = y_V \text{ et } z_p < z_V$$

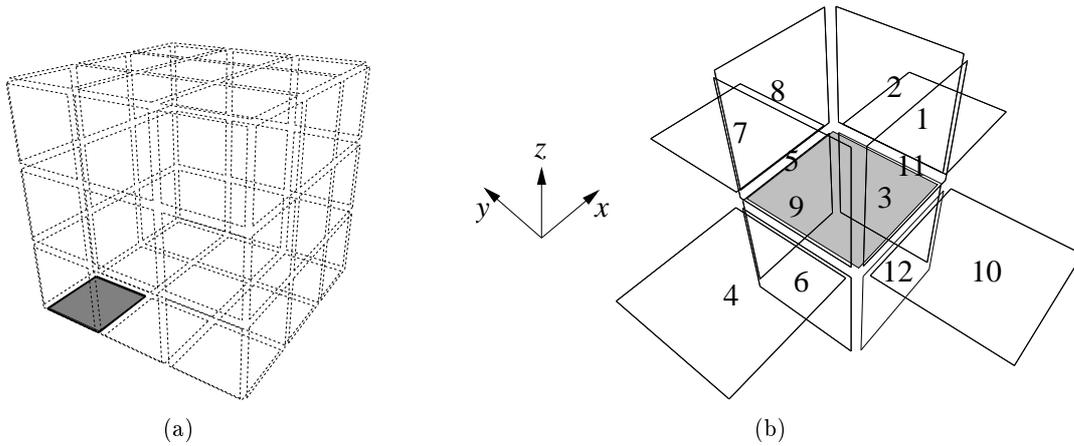


FIG. 3.16 – Surfel de départ et ordre de reconnaissance. (a) Surfel de départ choisi pour la première reconnaissance de plan. (b) Ordre d’ajout des surfels pendant le processus de reconnaissance.

Plusieurs stratégies peuvent être utilisées pour déterminer le premier surfel à reconnaître dans les étapes de reconnaissance de plan suivantes. Par exemple, nous pouvons choisir le surfel le plus ou le moins contraint, c’est à dire le surfel ayant le plus grand ou le plus petit nombre de contraintes (nous supposons qu’une contrainte de sommet (resp. arête) correspond à une (resp. deux) contrainte(s)). Nous pouvons aussi choisir en premier un surfel non contraint, puis, s’il ne reste aucun autre surfel non contraint à traiter, choisir le moins contraint. Dans ce travail, nous avons essayé quatre stratégies, que nous appelons dans la suite des *méthodes de propagation* :

- Méthode 1 : Avec cette méthode, le surfel de départ doit être contraint par au moins une contrainte. De plus, parmi ces surfels, nous choisissons le surfel le moins contraint.

Dans le cas où plusieurs surfels ont le même nombre de contraintes, nous en choisissons un arbitrairement.

- Méthode 2 : Cette méthode consiste à choisir en priorité les surfels adjacents aux premières facettes reconstruites. Par exemple, supposons que la première facette soit reconstruite. Dans ce cas, le surfel de départ suivant doit être adjacent à cette facette. Après reconstruction de la deuxième facette, le surfel de départ suivant doit être adjacent à la première facette reconstruite, ou, dans le cas où il n'existe aucun surfel de ce type, il doit être adjacent à la deuxième facette. Et ainsi de suite, jusqu'à ce qu'il n'y ait plus de surfel à traiter.
- Méthode 3 : Ici, le surfel de départ choisi est un surfel choisi arbitrairement parmi les surfels non contraints. Lorsqu'il n'y a plus de surfel non contraint, nous choisissons un des surfels les moins contraints, et ainsi de suite jusqu'à ce que tous les surfels soient traités.
- Méthode 4 : Cette dernière méthode consiste simplement à choisir comme surfel de départ un des surfels les plus contraints de la surface discrète.

3.3.2.2 Ordre de reconnaissance

Pendant le processus de reconnaissance, si S est le dernier surfel reconnu, les surfels suivants à être reconnus sont choisis dans l'ordre montré sur la figure 3.16b. Le choix de l'ordre de reconnaissance est très important puisque la forme des plans reconnus dépend fortement de cet ordre.

3.3.3 Reconnaissance de plan contrainte

Lorsque nous reconstruisons différents polygones, des contraintes apparaissent. En effet, les arêtes et les sommets des polygones reconstruits voisins doivent être pris en compte. La première face discrète est reconnue sans contraintes. Ce n'est pas le cas pour celles qui suivent puisque les sommets ou les arêtes de faces déjà construites peuvent contraindre le processus de reconnaissance (voir Algorithme 5). Plus précisément, soit $S = \{V_1, V_2, V_3, V_4\}$ le prochain surfel devant être reconnu. Si S est contraint, l'ensemble des solutions fourni par l'algorithme de reconnaissance doit être mis à jour en fonction de la contrainte $Cont(V_i)$ de chaque voxel.

3.3.3.1 Contrainte de sommet

La mise à jour de l'ensemble solution à cause d'une contrainte de sommet est simplement effectuée en faisant l'intersection entre la préimage généralisée et le plan dual de la contrainte. Si le résultat est vide, le processus de reconnaissance s'arrête. Sinon, un plan solution peut être choisi dans l'intersection calculée.

3.3.3.2 Contrainte d'arête

Dans le cas d'une contrainte d'arête, la mise à jour de l'ensemble solution est effectuée en faisant l'intersection entre la préimage généralisée et la droite correspondant aux plans solutions de la droite contenant l'arête (voir Chapitre 3).

Cependant, nous avons vu dans la section ?? que si la droite est parallèle au vecteur $(0, 0, a)$, $a \in \mathbb{R}$, elle n'a pas de plans solutions. Ce problème peut être traité en considérant la projection

Algorithme 5 : Reconnaissance de plan Standard contrainte

RECONNAÎTRE_PLAN

en entrée : Un contour de voxels V_c et un surfel non traité $S = \{V_1, V_2, V_3, V_4\}$

en sortie : Une courbe 2-connexe planeaire C et un plan P

Sol : Ensemble solution ;

début

$P \leftarrow \emptyset$;

$Sol \leftarrow \mathbb{G}_P(S) \cap_{i=1}^4 Dual(Cont(V_i))$;

tant que $Sol \neq \emptyset$ **faire**

 Choisir un plan P dans l'ensemble des solutions.

$S \leftarrow$ Prochain_surfel ;

$Sol \leftarrow Sol \cap_{i=1}^4 Dual(Cont(V_i))$;

Extraire la courbe discrète 2-connexe C à partir du bord des surfels reconnus ;

fin

des voxels déjà reconnus sur le plan $(0, X_1, X_2)$. En effet, la projection sur $(0, X_1, X_2)$ de la discrétisation Standard du plan vertical (i.e. le plan de paramètres $(a, b, 0, d)$, $(a, b, d) \in \mathbb{R}^3$) est une droite Standard. Ainsi, pour déterminer s'il existe un plan euclidien coupant tous les voxels reconnus et la contrainte d'arête, il suffit de déterminer s'il existe une droite qui coupe tous les voxels projetés (i.e. pixels) et qui contient la projection de l'arête (i.e. un point). Ceci est effectué en calculant la préimage généralisée des pixels dans l'espace de paramètres en dimension 2, et donc en effectuant l'intersection entre la droite duale du point et la préimage généralisée obtenue. Si l'intersection est vide, le processus de reconnaissance s'arrête. Sinon, une droite solution peut être choisie dans l'intersection, puis extrudée le long de l'axe des ordonnées, de façon à obtenir un plan solution.

3.3.3.3 Choix du plan

La reconnaissance du plan se fait à l'aide de la méthode décrite dans le chapitre 4. Une fois la préimage généralisée obtenue, nous choisissons de fixer un plan. Ce plan est choisi de la façon suivante. Lorsque la préimage est convexe, nous calculons les coordonnées de son barycentre, puis nous le transformons en une équation de plan dans l'espace euclidien. Lorsque la préimage n'est pas convexe, nous calculons le barycentre de la préimage restreinte à un quart d'espace (qui est obligatoirement convexe), puis nous déterminons le plan euclidien correspondant de la même manière.

3.3.4 Polygonalisation de face

Quand le processus de reconnaissance s'arrête, l'ensemble solution est soit vide, soit non vide. S'il n'est pas vide, un plan solution P a été choisi dans l'ensemble et le bord des surfels reconnus extrait. La courbe 2-connexe correspondante $C = \{V_1, \dots, V_n\}$ est alors reconstruite en fonction de P afin d'obtenir une facette contenue par P . Si l'ensemble solution est vide, cela signifie que le premier surfel à reconnaître était contraint de telle sorte qu'aucun plan contenant toutes les contraintes existe. Un traitement particulier est alors nécessaire. Nous décrivons ces deux cas dans les sections suivantes.

3.3.4.1 Reconstruction d'une 2-courbe discrète plane

Dans cette section, nous nous attaquons au problème suivant : étant donnée une 2-courbe discrète plane C (voir Figure 3.17a), c'est à dire une courbe qui appartient à un plan Standard, et étant donné un plan euclidien P qui coupe tous les voxels de la courbe, comment calculer une courbe polygonale plane se trouvant sur P et telle que sa discrétisation Standard est égale à C ?

Principe

Soit P le plan déterminé lors de l'étape de reconnaissance (voir Figure 3.17b). Tout d'abord, l'intersection entre P et la courbe discrète est effectuée afin d'obtenir une courbe plane composée de faces ayant des formes variées (voir Figure 3.17c), i.e. une courbe irrégulière. Puis, une courbe polygonale plane se trouvant sur P et coupant toutes les faces de la courbe irrégulière est calculée (voir Figure 3.17d).

Notre algorithme est donc composé de deux étapes. Tout d'abord, le calcul de l'intersection entre la courbe discrète et le plan choisi. Puis, le calcul d'une polyligne coupant tous les pavés de la courbe irrégulière.

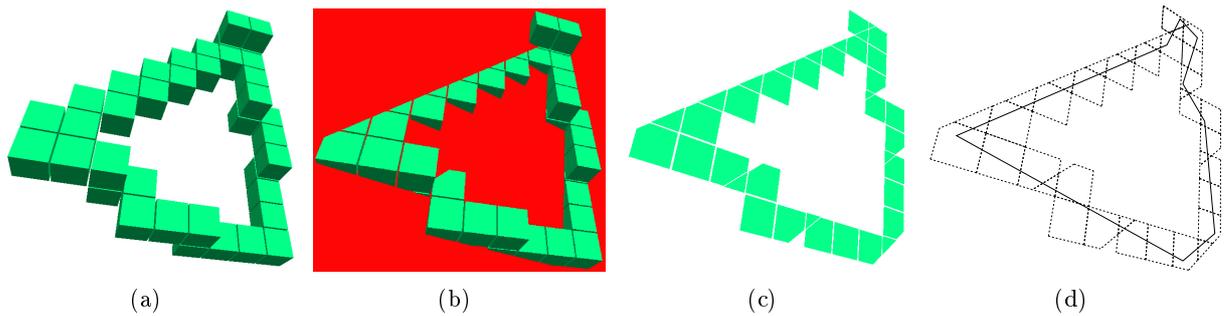


FIG. 3.17 – Exemple de polygonalisation de courbe : (a) Une courbe discrète, (b) Un plan coupant tous les voxels, (c) Intersection entre le plan et la courbe, (d) Un résultat possible.

Calcul de l'intersection entre un plan euclidien et une 2-courbe discrète

Afin de calculer l'intersection entre C et P (voir Algorithme 6), nous effectuons de manière incrémentale des opérations d'intersection entre P et les différentes cellules topologiques constituant les voxels de la courbe (faces et arêtes).

Tout d'abord, tous les sommets de la courbe sont triés en fonction de leur position par rapport à P (au dessus, en dessous ou dans le plan). Trois sous-ensembles de sommets différents notés \oplus , \ominus et \ominus sont alors obtenus. A partir de ces ensembles, nous pouvons déterminer les arêtes de la courbe qui sont coupées par P . En effet, ces arêtes possèdent une extrémité dans l'ensemble \oplus , et une autre dans l'ensemble \ominus .

L'intersection entre ces arêtes et P peut alors être calculée de manière à créer de nouveaux sommets (voir Figure 3.18a). Notons que le choix du plan P implique que toute intersection entre une arête et P est nécessairement réduite à un point. En effet, P est choisi de telle manière à ne pas contenir de sommet de la courbe C , et donc ne contient pas d'arête.

Puis, pour chaque face de voxel, si deux sommets ont été créés dans la face, une nouvelle arête est créée entre ces deux sommets (voir Figure 3.18b).

Algorithme 6 : Opération d'intersection entre une 2-courbe discrète classique et un plan euclidien

en entrée : Une 2-courbe discrète C et un plan P .
en sortie : Une 1-courbe irrégulière.

pour chaque sommet S de C faire
 | $pos \leftarrow$ évaluer la position de S par rapport à P ;
 | marquer S avec $\oplus, \ominus, \omin�$ en fonction de pos ;

pour chaque arête A de C faire
 | **si sommets de A sont marqués par \oplus et \ominus alors**
 | $I \leftarrow$ calculer le point d'intersection entre A et P ;
 | $S \leftarrow$ insérer un nouveau sommet dans A à la position I ;
 | marquer S avec $\omin�$;

pour chaque face F de C faire
 | **si F contient deux sommets S_1, S_2 marqués avec $\omin�$ alors**
 | insérer une nouvelle arête dans F entre S_1 et S_2 ;

pour chaque voxel V de C faire
 | **si V contient de nouvelles arêtes alors**
 | $F \leftarrow$ créer une nouvelle face en convexe à partir des nouvelles arêtes;

pour chaque nouvelle face F faire
 | **pour chaque arête A de F faire**
 | $A' \leftarrow$ arête correspondant à A sur le voxel contenant F ;
 | $F' \leftarrow$ trouver la face correspondant à A' à l'intérieur du voxel adjacent;
 | **si F' existe alors**
 | relier F et F' le long de A ;

retourner l'ensemble des nouvelles faces;

Enfin, si de nouvelles arêtes ont été créées sur un même voxel, une nouvelle face convexe est créée à partir de ces arêtes (voir Figure 3.18c).

La dernière étape consiste alors à établir des relations entre les faces nouvellement créées de manière à reproduire les relations d'adjacence existant auparavant entre les différents voxels de la courbe.

Intéressons nous à présent à la complexité de l'algorithme présenté. Puisque chaque cellule de la courbe n'est traitée qu'une seule et unique fois lors du déroulement de cet algorithme, sa complexité est en $\mathcal{O}(i)$, où i correspondant au nombre de cellules de la courbe C . De plus, puisque i et le nombre de voxels k de la courbe sont linéairement dépendants, nous en déduisons que la complexité de cet algorithme est $\mathcal{O}(k)$.

Polygonalisation d'une courbe irrégulière

Une courbe irrégulière planaire C' , ouverte ou fermée, ayant été extraite de l'intersection entre le plan P et la courbe de voxels C , notre but est à présent de reconstruire cette courbe discrète. Nous souhaitons donc obtenir une courbe polygonale planaire coupant tous les pavés de C' . Pour ce faire, diverses méthodes peuvent être utilisées.

Nous pouvons par exemple étendre la méthode de reconstruction de 1-courbe discrète classique *étendue* que nous avons présentée dans le chapitre 3.2 au cas d'une 1-courbe irrégulière. Ceci peut être aisément effectué en considérant non plus la préimage généralisée d'un ensemble

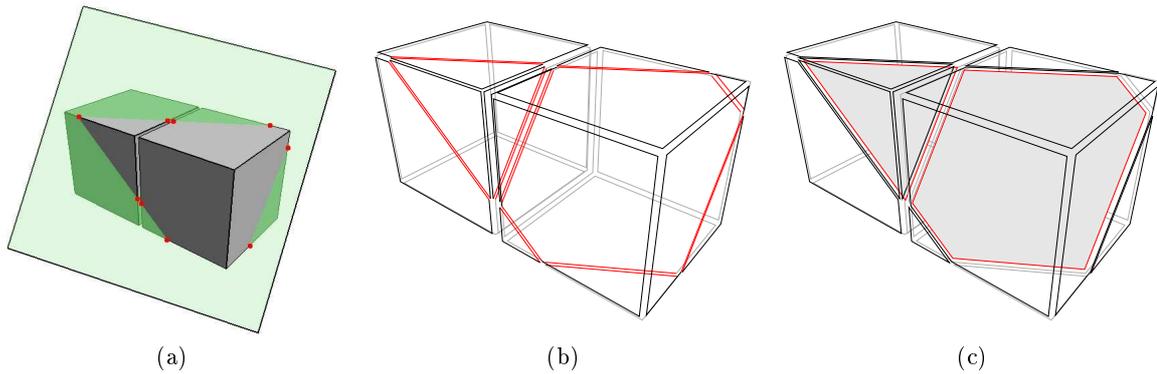


FIG. 3.18 – Illustration de l'algorithme d'intersection : (a) Opération d'intersection entre le plan et les arêtes, (b) Création de nouvelles arêtes, (c) Création de nouvelles faces.

de pixels, mais la préimage généralisée d'un ensemble de pavés quelconques dans l'algorithme de reconnaissance de segments discrets.

Bien que cette méthode fournisse probablement de bons résultats en terme de nombre de segments reconstruits, nous avons choisi d'utiliser une autre méthode dans le cadre de notre algorithme de reconstruction. En effet, dans un souci d'efficacité, nous avons opté pour une méthode de reconstruction de complexité linéaire en le nombre de pavés de la courbe irrégulière.

Cette deuxième méthode est une extension aux espaces irréguliers de l'algorithme de reconstruction proposé par D. CŒURJOLLY ET AL. [CZ06] dans le cadre de grilles irrégulières isothétiques, et qui est basé sur le calcul de *cônes de visibilité*. *

Le principe de notre méthode de polygonalisation est le suivant :

Tout d'abord, un premier pavé P est choisi dans la courbe C , et un point euclidien P_0 est choisi à l'intérieur de P . Nous pouvons par exemple prendre le barycentre de P puisque celui-ci est par construction convexe. Notons que dans le cas d'une courbe ouverte, nous choisissons un des pavés extrémités. Enfin, un sens de parcours de la courbe est lui aussi choisi.

Le parcours de la courbe commence à partir de P et dans le sens précédemment défini. Un premier cône C_0 est alors calculé entre le premier point fixé P_0 et le pavé suivant, qui est adjacent à P . Ce cône est défini par P_0 et l'arête e_0 commune à P et le pavé suivant (voir Figure 3.19a). Puis, un nouveau cône C_1 est calculé entre P_0 et le pavé suivant à partir de l'arête e_1 commune aux deux derniers pavés atteints. L'intersection entre C_0 et C_1 , correspondant au cône de visibilité est calculée, et tant que ce cône n'est pas vide, le processus continue pour les pavés suivants de la courbe (voir Figure 3.19b).

Si le cône de visibilité devient vide, cela signifie qu'aucune droite du cône de visibilité précédemment calculé n'atteint le pavé suivant de la courbe (voir Figure 3.19c). Une droite du cône de visibilité précédemment calculé est alors choisie et un nouveau point euclidien P_1 , correspondant à un nouveau sommet de la courbe polygonale en cours de construction, est fixé sur l'intersection entre cette droite et le pavé précédemment atteint. Le processus de calcul d'un nouveau cône commence alors à nouveau à partir de ce point (voir Figure 3.19d).

Quand le dernier pavé de la courbe est atteint, et dans le cas d'une courbe fermée, il est nécessaire de vérifier si le dernier cône de visibilité calculé contient le premier point fixé dans la courbe, i.e. P_0 . Si c'est le cas, la courbe polygonale est entièrement reconstruite, sinon, un point est fixé dans le pavé précédant P dans la courbe, et un segment est ajouté entre les deux pavés.

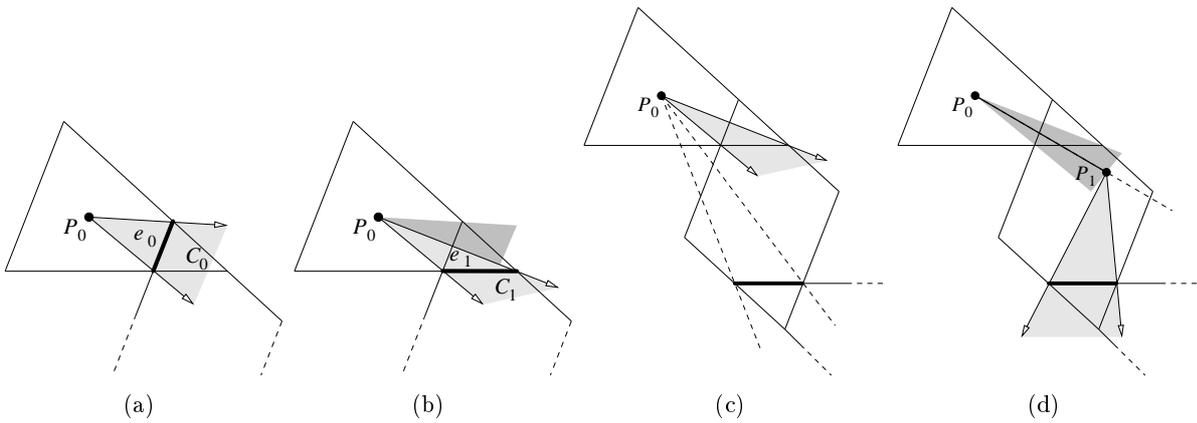


FIG. 3.19 – Illustration de l'algorithme de polygonalisation de 1-courbe irrégulière : (a) Premier cône calculé, (b) Réduction du cône de visibilité en fonction du pavé suivant, (c) Exemple de cône de visibilité vide, (d) Choix d'un nouveau point et calcul d'un nouveau cône.

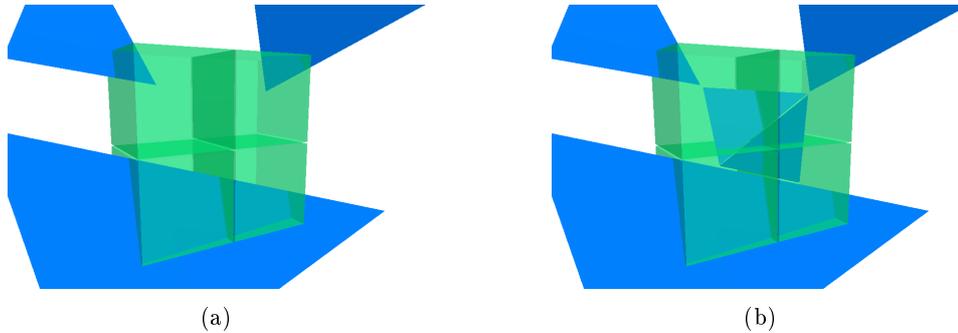


FIG. 3.20 – Traitement particulier. (a) Un surfel contenant trois voxels contraints. (b) Ajout de deux facettes adjacentes.

3.3.4.2 Ensemble de solution de surfels vide

Si l'ensemble solution est vide, cela signifie qu'aucun plan ne contient toutes les contraintes du premier surfel devant être reconnu. Il s'agit d'un surfel contenant trois voxels contraints. La reconstruction d'un tel surfel S est effectuée en ajoutant deux facettes adjacentes (voir Figure 3.20). Dans ce but, un point euclidien est fixé dans chaque voxel de S qui n'est pas contraint par un sommet. Si le voxel est contraint par une arête, le centre du segment résultant de l'intersection entre l'arête et le voxel est choisi. Si le voxel n'est pas contraint, le centre du voxel est choisi comme point fixé. Puis, deux facettes sont ajoutées entre les quatre contraintes de sommet.

3.3.5 Résultats

Nous pouvons voir sur la figure 3.21 quelques résultats obtenus avec notre méthode de reconstruction (voir Figures 3.25, 3.26, 3.27 et 3.28 pour plus d'exemples).

Dans un premier temps, nous pouvons tout d'abord constater que quelle que soit la méthode de propagation utilisée, les surfaces polygonales reconstruites ne sont pas visuellement de très bonne qualité. En effet, la méthode de reconnaissance utilisée étant fortement contrainte, la surface obtenue est très "accidentée". Cet état de fait était cependant prévisible puisque la

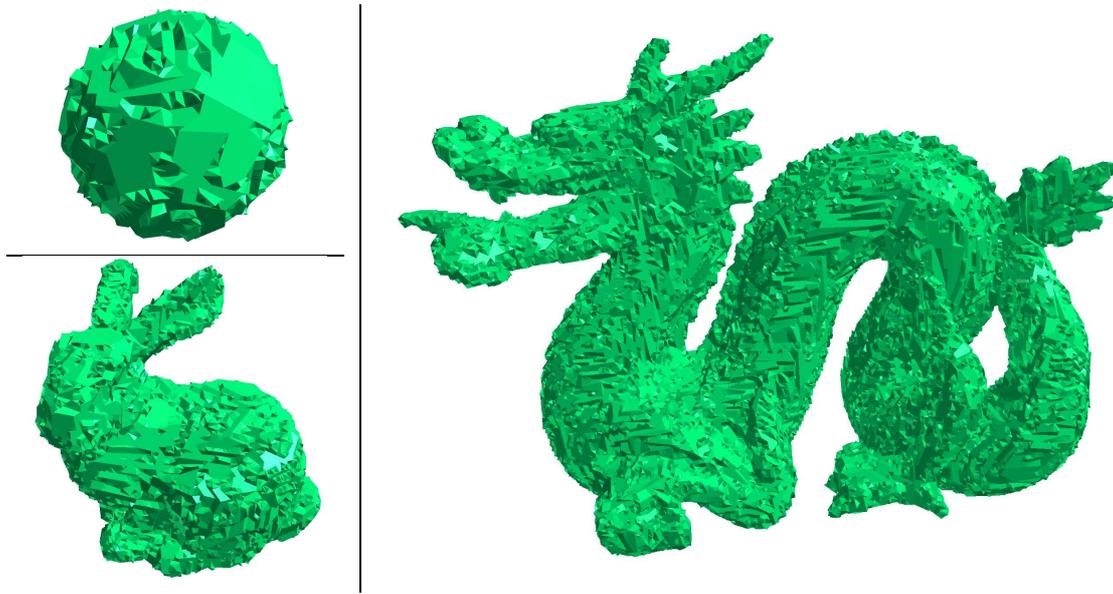


FIG. 3.21 – Exemples d’objets obtenus avec notre méthode de reconstruction.

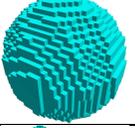
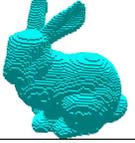
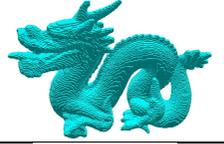
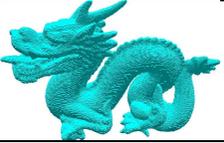
méthode de reconstruction sans joint (voir Section 3.2.1.2), dont notre méthode est fortement inspirée, génère des courbes polygonales de qualité similaire, là encore à cause de l’utilisation d’un algorithme de reconnaissance fortement contraint.

De plus, nous pouvons constater que les facettes reconstruites sont de formes et de tailles extrêmement variées et présentent dans la plupart des cas de très nombreuses concavités (voir Figure 3.22). L’inconvénient majeur de la présence de ces dernières est que les facettes devant être reconstruites à partir de ces concavités sont souvent de très petite taille. En effet, le surfel contenant le sommet intérieur de la concavité est contraint par deux contraintes d’arêtes identiques (les deux arêtes sont contenues dans le même plan). Enfin, les faces reconstruites peuvent dans certains cas contenir un trou, comme nous pouvons le voir sur le figure 3.22b, ce qui là encore génère des facettes de petite taille, et pose de surcroît un problème d’ordre topologique.

Ensuite, il semble que la méthode de propagation choisie ait un impact non négligeable sur le nombre de facettes générées par notre algorithme. En effet, nous pouvons voir sur les figures 3.23 et 3.24 deux exemples de reconstructions obtenues à l’aide des quatre méthodes de propagation testées (voir Section 3.3.2.1). En particulier, les nombres de facettes les plus faibles sont donnés par les méthodes 1, 2 et 3, et sont de plus relativement proches les uns des autres, la méthode 3 donnant cependant un nombre de facettes légèrement plus élevés. Par contre, la méthode 4, consistant à choisir systématiquement un des surfels les plus contraints, génère un nombre de facettes nettement plus important. Les méthodes 1 et 2 semblent donc être les méthodes plus appropriées parmi les quatre méthodes évaluées pour notre algorithme de reconstruction.

Enfin, nous avons comparé, pour quelques objets, le nombre de facettes obtenues avec la méthode de propagation 1 par rapport au nombre de triangles obtenus pour les mêmes objets avec la méthode des Marching Cubes. Ces résultats sont présentés dans le tableau 3.2. Nous pouvons constater que nous obtenons approximativement 70% de faces en moins que la méthode des Marching Cubes.

TAB. 3.2 – Résultats : comparaison entre la méthode classique des Marching Cubes (MC) et notre méthode de reconstruction analytique discrète (RAD).

Données fournies	Dimensions du volume discret	# points discrets de la surface	MC # sommets	MC # triangles	RAD # sommets	RAD # faces
	33^3	4824	5212	10420	3013	3472
	$80 \times 79 \times 62$	22056	23105	46204	14037	16456
	$206 \times 146 \times 94$	110938	114552	229096	73649	87182
	$95 \times 231 \times 251$	135902	134503	268658	89659	105469
	$310 \times 218 \times 140$	246992	252506	505012	163540	194989

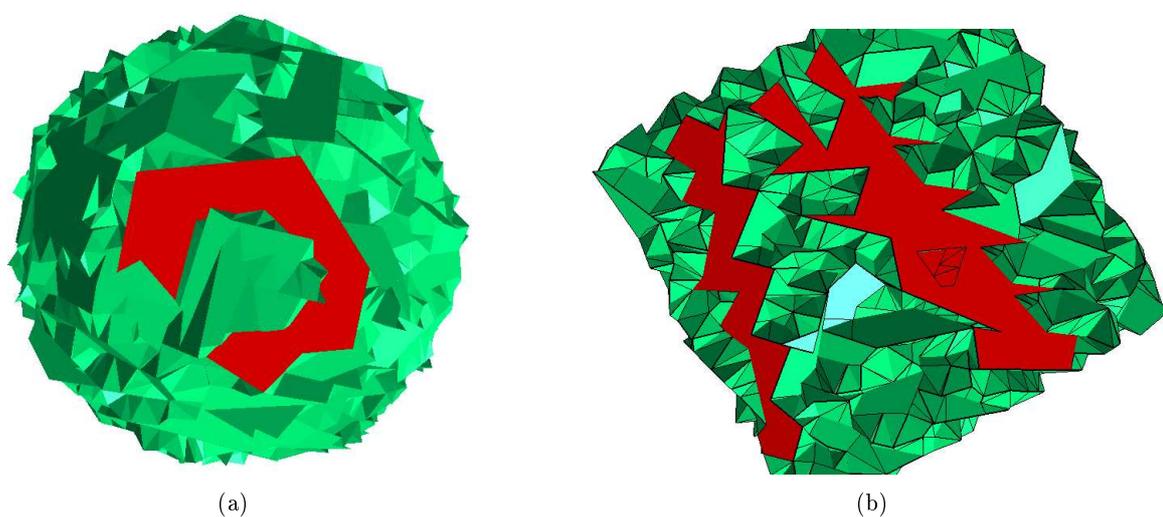


FIG. 3.22 – Exemple de facettes reconstruites concaves (en rouge sur les figures (a) et (b)). Sur la figure (b), nous pouvons voir le cas d'une facette contenant un trou.

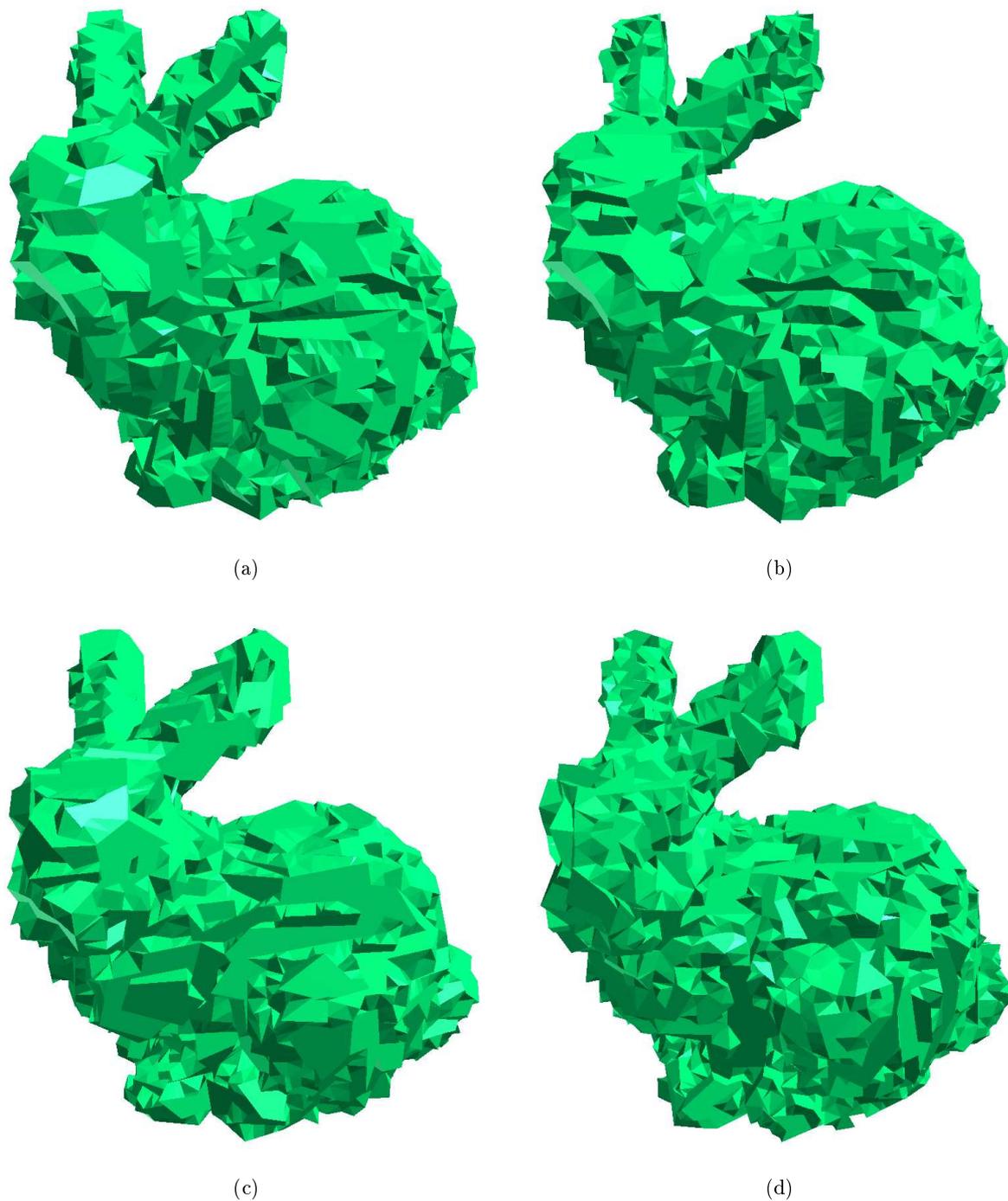


FIG. 3.23 – Exemple d'objet discret reconstruit à l'aide de quatre méthodes de propagation différentes : (a) Méthode 1 : 4358 faces, (b) Méthode 2 : 4494 faces, (c) Méthode 3 : 4603 faces, (d) Méthode 4 : 4850 faces.

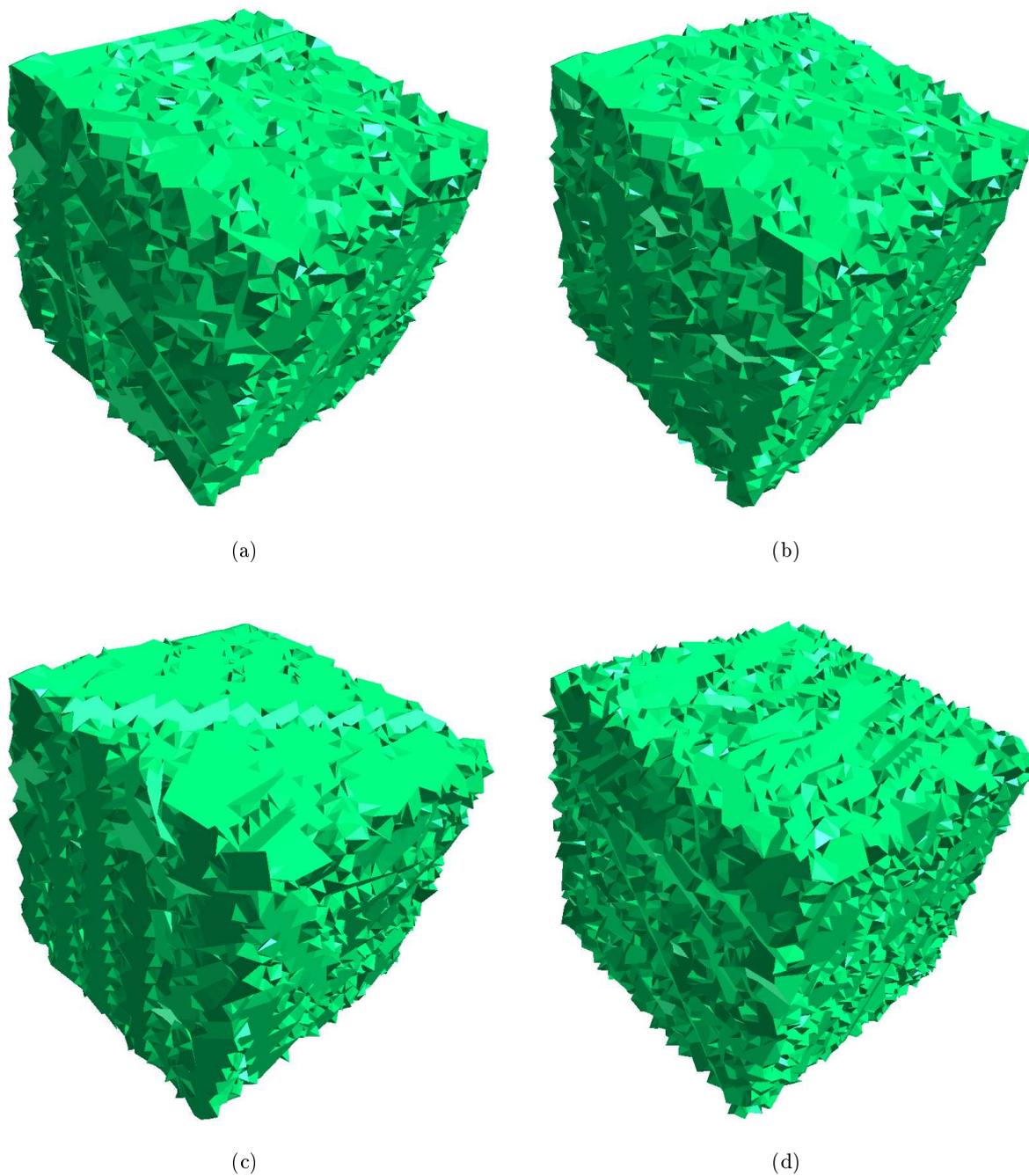


FIG. 3.24 – Exemple d'objet discret reconstruit à l'aide de quatre méthodes de propagation différentes : (a) Méthode 1 : 7661 faces, (b) Méthode 2 : 7646 faces, (c) Méthode 3 : 7950 faces, (d) Méthode 4 : 9176 faces.

3.3.6 Conclusion et perspectives

Dans cette section, nous avons présenté une nouvelle méthode de reconstruction du bord de volumes 2-connexe en dimension 3.

Cette méthode fournit un résultat inversible pour le modèle Standard et topologiquement cohérent avec l'objet discret de départ. Cette méthode correspond à l'extension de la méthode sans joints à la dimension 3. Elle s'effectue en un seul parcours, les facettes étant reconnues et reconstruites les unes après les autres grâce à l'utilisation d'un algorithme de reconnaissance contrainte, et d'un algorithme de polygonalisation de 2-courbes discrètes planaires.

L'algorithme de reconnaissance contrainte utilise l'algorithme de reconnaissance de plans discrets Standard basé sur le calcul de la préimage généralisée des voxels, qui a été présenté dans le chapitre 4.

Nous avons de plus proposé un nouvel algorithme de polygonalisation de courbes 2-connexes planaires de complexité linéaire.

Les surfaces générées avec cette méthode, bien qu'inversibles pour le modèle Standard et topologiquement correctes, sont visuellement peu satisfaisantes. Plusieurs améliorations peuvent cependant être apportées afin d'obtenir de meilleurs résultats. Tout d'abord, l'algorithme de reconnaissance pourrait être modifié de manière à reconnaître des morceaux de plans discrets plus réguliers. Des méthodes de segmentations en morceaux de plans discrets telles que celles proposées par I. SIVIGNON dans sa thèse [Siv04] pourraient par exemple être appliquées.

De plus, il serait intéressant d'essayer de détecter des voxels, ou des ensembles de voxels appartenant à plusieurs plans discrets, et ce à l'aide d'une étude des configurations locales de voxels.

Enfin, comme nous l'avons souligné, l'algorithme de polygonalisation de courbes discrète planaires pourrait être amélioré en appliquant notre méthode étendue à la polygonalisation des courbes irrégulières obtenues lors de cette deuxième phase de notre algorithme.

Une dernière perspective à ces travaux serait d'essayer d'appliquer notre méthode étendue à la dimension 3, et ce afin de réduire les contraintes de notre algorithme de reconnaissance de morceaux de plans discrets actuel.

L'algorithme de reconstruction de 2-courbes discrètes planaires et son application à la reconstruction de volumes discrets ont été publiés dans [DCA06].

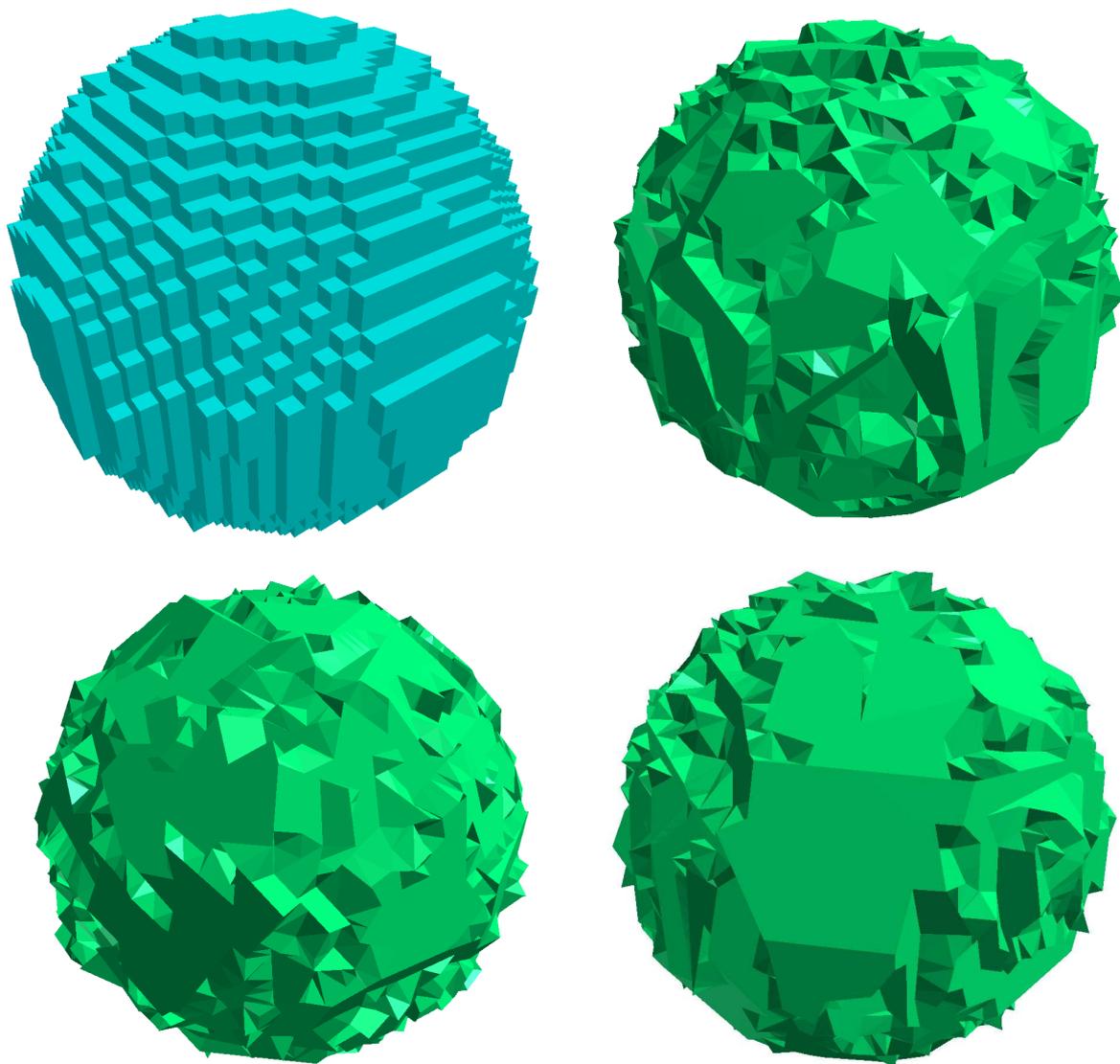


FIG. 3.25 – Exemple de reconstruction obtenue avec notre méthode : la surface de l'objet discret est constituée de 4824 points discrets, et l'objet reconstruit contient 3472 faces. La méthode de propagation utilisée est la méthode 2.

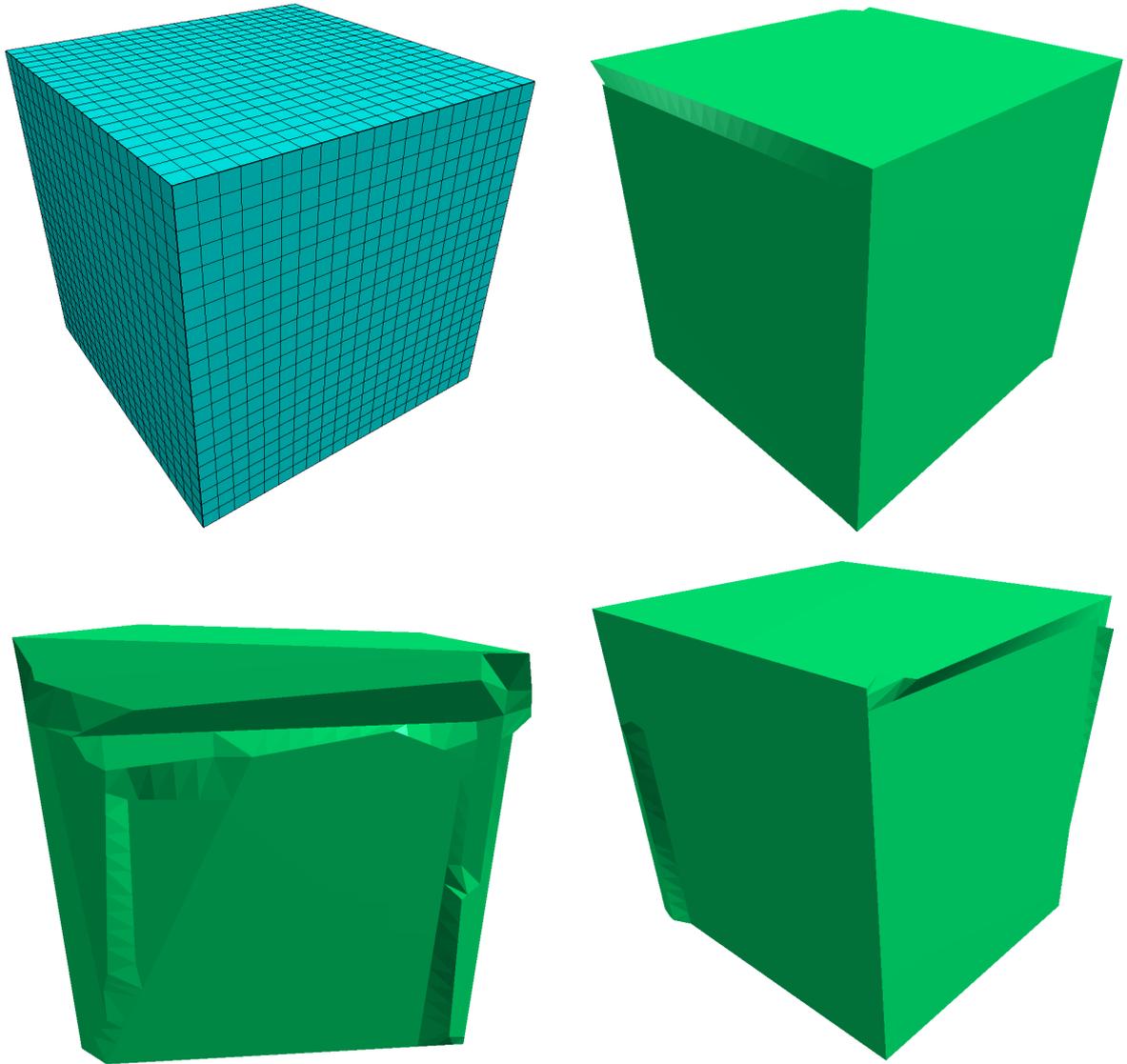


FIG. 3.26 – Exemple de reconstruction obtenue avec notre méthode : la surface de l'objet discret est constituée de 2648 points discrets, et l'objet reconstruit contient 336 faces. La méthode de propagation utilisée est la méthode 1.

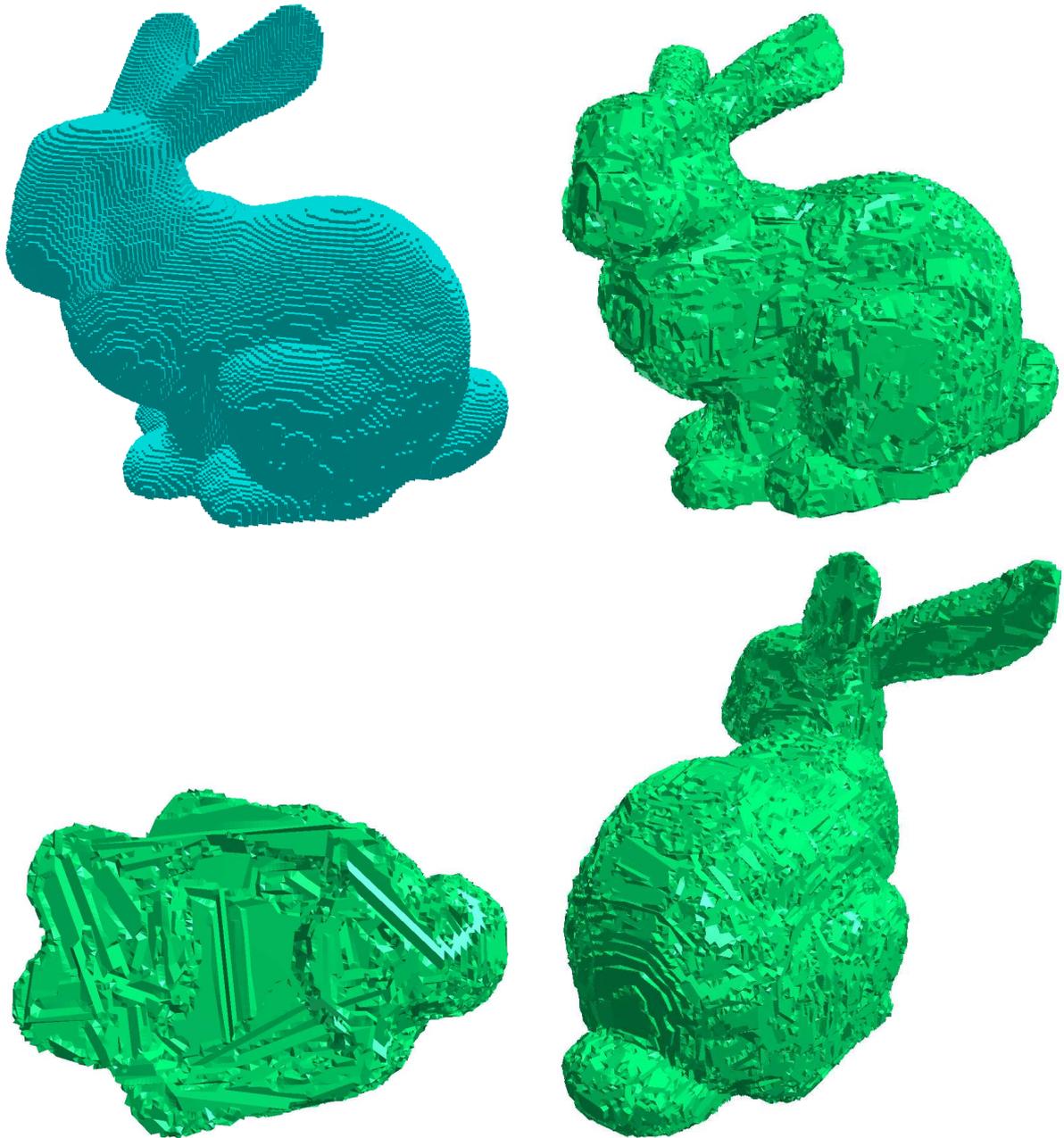


FIG. 3.27 – Exemple de reconstruction obtenue avec notre méthode : la surface de l'objet discret est constituée de 86516 points discrets, et l'objet reconstruit contient 63247 faces. La méthode de propagation utilisée est la méthode 1.

3.4 Conclusion du chapitre

Dans ce chapitre, nous avons tout d'abord proposé un nouvel algorithme de reconstruction analytique inversible de 1-courbes discrètes classiques ouvertes ou fermées, basé sur notre algorithme de reconnaissance de droites Standard. Cet algorithme améliore en particulier les résultats obtenus avec les méthodes avec et sans joint proposées dans [Bre03]. Nous avons de plus proposé une méthode de reconstruction de courbes non simples.

Nous avons de plus proposé une méthode de reconstruction analytique en dimension 3 permettant la reconstruction du bord d'un volume discret 2-connexe. La surface polygonale générée est en particulier inversible pour le modèle Standard et est topologiquement cohérente avec le volume discret d'origine. Cette méthode, tout comme la méthode de reconstruction proposée en dimension 2 est basée sur l'algorithme de reconnaissance de plans Standard présenté dans le chapitre 4. Diverses perspectives à ces travaux ont de plus été proposées.

Les différents algorithmes présentés ont été implantés et testés, et ce en particulier au sein du logiciel de modélisation d'objets géométriques discrets en dimensions 2 et 3 que nous présentons dans le chapitre suivant.

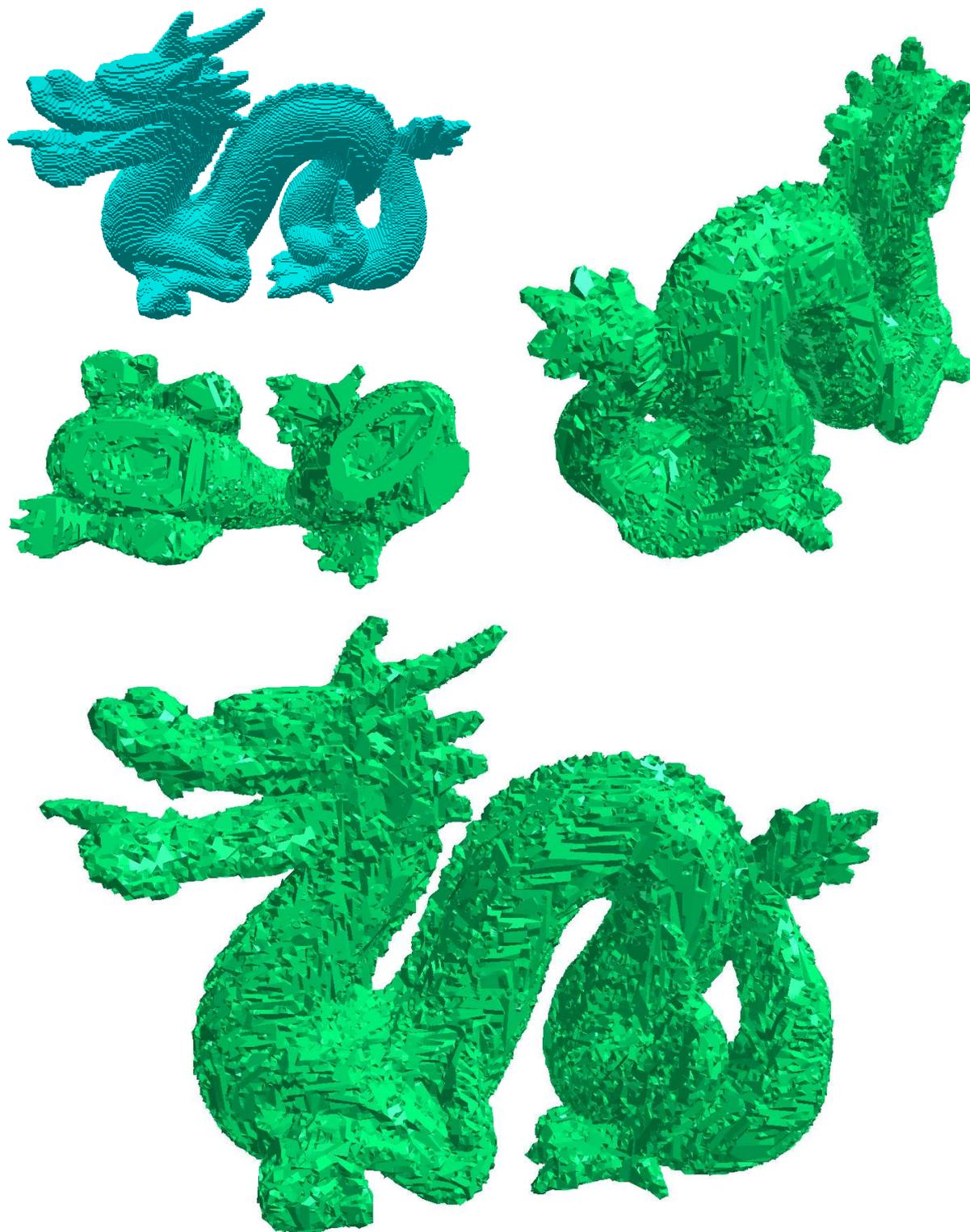


FIG. 3.28 – Exemple de reconstruction obtenue avec notre méthode : la surface de l'objet discret est constituée de 110940 points discrets, et l'objet reconstruit contient 87182 faces. La méthode de propagation utilisée est la méthode 2.

Chapitre 4

Un modeleur géométrique à base topologique d'objets discrets

Sommaire

4.1	Introduction	87
4.2	Le modèle topologique des cartes généralisées	89
4.2.1	Définition d'une carte généralisée	90
4.2.2	Cellules topologiques	90
4.2.3	Opérations de base	91
4.3	Une structure hiérarchique	92
4.3.1	Les niveaux	94
4.3.1.1	Niveau 3 : la représentation euclidienne	94
4.3.1.2	Niveau 2 : la représentation discrète analytique	95
4.3.1.3	Niveau 1 : les contours discrets	95
4.3.1.4	Niveau 0 : représentation discrète classique duale	96
4.3.1.5	Mise en œuvre	97
4.3.2	Contraintes de cohérence	98
4.4	Opérations dans la structure	99
4.4.1	Construction des niveaux et mise en place des liaisons	100
4.4.1.1	Discrétisation d'un objet euclidien	100
4.4.1.2	Reconstruction d'un objet discret	107
4.4.2	Modification et mise à jour des niveaux	108
4.4.2.1	Manipulation des représentations	109
4.4.2.2	Mise à jour de la structure	109
4.5	Conclusion et perspectives	115

4.1 Introduction

Les travaux présentés dans les chapitres précédents ont été effectués dans le cadre de la conception et du développement d'un logiciel de modélisation d'objets discrets, baptisé *SpaMod*⁹. Ce logiciel possède deux particularités qui le distinguent des logiciels de modélisation classiques.

⁹Acronyme de Spatial Modeler.

La première particularité est qu'il autorise la création, la manipulation et le traitement d'objets géométriques représentés soit sous forme discrète, soit sous forme continue (ou *euclidienne*). La deuxième particularité est que ces objets peuvent être considérés soit dans un espace de dimension 2, soit dans un espace de dimension 3.

Les opérations relatives aux objets euclidiens, à savoir la création et la manipulation de ces objets, est assurée par un logiciel de modélisation à base topologique d'objets géométriques euclidiens appelé *Moka*¹⁰. Ce logiciel, actuellement en cours de développement dans le laboratoire SIC¹¹ de Poitiers, autorise à l'heure actuelle la manipulation d'objets euclidiens sous formes polygonale (dimension 2) ou polyédrique (dimension 3). Un ensemble important d'opérations de création, telles que la création de sphères ou de maillages, ainsi que des opérations géométriques élémentaires, telles que la rotation, l'homothétie ou encore la translation y sont intégrées.

Notre logiciel se présente donc comme une extension du modeleur Moka. Avec ce logiciel, notre but est de fournir un large choix d'opérations tant discrètes que continues afin de pouvoir manipuler des objets géométriques fournis sous l'une ou l'autre de ces deux formes. Nous souhaitons de plus pouvoir travailler à un moment donné et de manière préférentielle soit sur la représentation discrète, soit sur la représentation euclidienne d'un même objet géométrique. La représentation qui n'a pas été modifiée peut alors, afin de conserver les différentes représentations cohérentes, être mise à jour en conséquence en appliquant soit un algorithme de discrétisation (voir Chapitre 1), soit un algorithme de reconstruction (voir Chapitre 3). Cependant, une mise à jour des objets dans leur intégralité peut s'avérer particulièrement coûteuse en terme de temps de calcul. Ainsi, nous souhaitons pouvoir effectuer des mises à jour locales de nos objets lorsque celles-ci sont possibles.

Afin de représenter et manipuler nos objets, nous avons besoin d'une structure de données regroupant les différentes représentations, et établissant une relation concrète entre elles. Dans une première version de SpaMod [ABL01], la structure proposée regroupait plusieurs représentations, à la fois discrètes et continue, du même objet géométrique. Cependant, seule la représentation continue pouvait être modifiée, et dans ce cas, l'ensemble des représentations de l'objet devait être entièrement recalculé.

Cette structure de données ne répondant pas à nos attentes, nous nous sommes alors intéressés à l'étude d'une nouvelle structure adaptée aux deux problèmes suivants :

- La structure doit autoriser la modification des différentes représentations.
- Les différentes représentations doivent être reliées¹² afin de pouvoir éventuellement répercuter localement aux autres représentations une modification de l'une d'entre elles.

Afin de concilier ces différentes contraintes, nous avons choisi d'intégrer les différentes représentations dans une même structure de données hiérarchique. Ce type de structure consiste en plusieurs *niveaux* ordonnés, chaque niveau étant physiquement relié aux niveaux le précédant et le suivant dans la structure. Plusieurs exemples d'utilisation d'une structure hiérarchique dans le cadre de la manipulation d'images sont décrits dans [GS06]. De même, un exemple d'utilisation d'une telle structure dans un cadre différent, la modélisation de complexes architecturaux, est décrit dans [Fra04]. Dans notre cas, chaque niveau de la structure correspond à une représentation particulière, euclidienne ou discrète, d'un même objet géométrique. De plus, toutes

¹⁰Site web de Moka : <http://www.sic.sp2mi.univ-poitiers.fr/moka/>.

¹¹Signal, Image, Communications - Université de Poitiers

¹²Nous entendons par "reliées" l'existence de relations entre les différentes représentations nous permettant de mettre en correspondance les divers éléments composant chaque représentation. Ceci est en effet possible puisque ces représentations sont construites à partir d'un même objet euclidien.

les représentations peuvent être modifiées, et recalculées à partir des autres représentations, en utilisant notamment tout ou partie des processus de discrétisation et de reconstruction. Enfin, notons qu'il est nécessaire de garantir une cohérence entre les différents niveaux puisque toutes les représentations sont basées sur un unique objet.

Afin de modéliser les représentations constituant les quatre niveaux de la structure, nous avons choisi d'utiliser le modèle topologique des *cartes généralisées*. Il s'agit d'un modèle de représentation par les bords, défini en toute dimension, qui nous permet de disposer pour chaque représentation d'informations à la fois topologiques et géométriques. Le choix de ce modèle est en grande partie lié au fait que la structure de données interne au modèleur Moka est elle-même basé sur ce modèle. Ainsi, dans un souci d'homogénéité entre les différentes représentations, nous avons choisi de conserver ce modèle de représentation.

Enfin, dans le but de résoudre le second problème posé, à savoir la possibilité de propager localement à toutes les représentations des modifications appliquées à l'une d'entre elles, nous établissons des liaisons entre elles. Ces liaisons sont situées au niveau topologique de chaque représentation.

Ce dernier chapitre est organisé comme suit. Tout d'abord, dans la section 4.2, nous faisons quelques rappels concernant le modèle topologique des cartes généralisées. Puis, dans la section 4.3, nous détaillons notre structure hiérarchique en précisant de plus de quelle manière elle a été mise en œuvre au sein de SpaMod. Enfin, la section 4.4 est quant à elle consacrée à la description de plusieurs opérations basées sur la structure décrite. Dans les deux dernières sections, nous nous limitons à l'étude du problème en dimension 2. Cependant, la structure et les opérations peuvent tout aussi bien être décrites en dimension 3, comme nous le verrons dans la conclusion de ce chapitre.

4.2 Le modèle topologique des cartes généralisées

Un modèle topologique permet de représenter des subdivisions de \mathbb{R}^n en cellules de dimension $i \in \llbracket 0, n \rrbracket$ (sommets, arêtes, faces et volumes pour les dimensions 0, 1, 2 et 3). Il permet de plus de représenter explicitement les relations d'incidence et d'adjacence entre les différentes cellules. Un modèle topologique peut alors être complété par diverses informations, telles que des informations géométriques. Ces informations sont appelées des *plongements*.

Afin de disposer d'informations topologiques au sein de notre structure, nous avons choisi de représenter nos objets géométriques, qu'ils soient sous forme discrète ou euclidienne, à l'aide du modèle topologique des *cartes généralisées* ou *G-cartes* [Lie89, Lie94]. Ce modèle, introduit par P. LIENHARDT, est un modèle combinatoire qui permet de représenter la topologie de quasi-variétés cellulaires, orientables ou non orientables, avec ou sans bords. De manière informelle, une quasi-variété correspond à un assemblage deux à deux de cellules de dimension i le long de cellules de dimension $i-1$, $i \in \llbracket 1, n \rrbracket$. Ce modèle fait partie des modèles dits de *représentation par les bords* ou *B-rep* [Lie91].

Dans ce qui suit, nous donnons plusieurs définitions et notations relatives aux cartes généralisées. Nous présentons de plus les trois opérations fondamentales permettant la construction d'une carte généralisée.

4.2.1 Définition d'une carte généralisée

Une carte généralisée de dimension n est composée d'un ensemble d'éléments abstraits appelés *brins*, et d'un ensemble d'opérations sur ces brins, notées α_i , $i \in \llbracket 0, n \rrbracket$.

Définition 23 (n -G-carte) Soit $n \in \mathbb{N}$. Une carte généralisée de dimension n est une algèbre $G = (B, \alpha_0, \dots, \alpha_n)$, où :

- B est un ensemble fini de brins ;
- Pour tout i tel que $0 \leq i \leq n$, α_i est une involution¹³ sur B ;
- Pour tout couple (i, j) tel que $0 \leq i < i + 2 \leq j \leq n$, $\alpha_i \circ \alpha_j$ est une involution.

La figure 4.1 illustre la décomposition topologique d'un objet en dimension 2 à l'aide d'une 2-G-carte. L'objet topologique de la figure 4.1a est décomposé en faces topologiques reliées par des involutions α_2 (voir Figure 4.1b). De la même manière, les faces sont décomposées en arêtes topologiques reliées par des involutions α_1 (voir Figure 4.1c). Enfin, les arêtes sont décomposées en deux brins qui sont reliés par des involutions α_0 (voir Figure 4.1d).

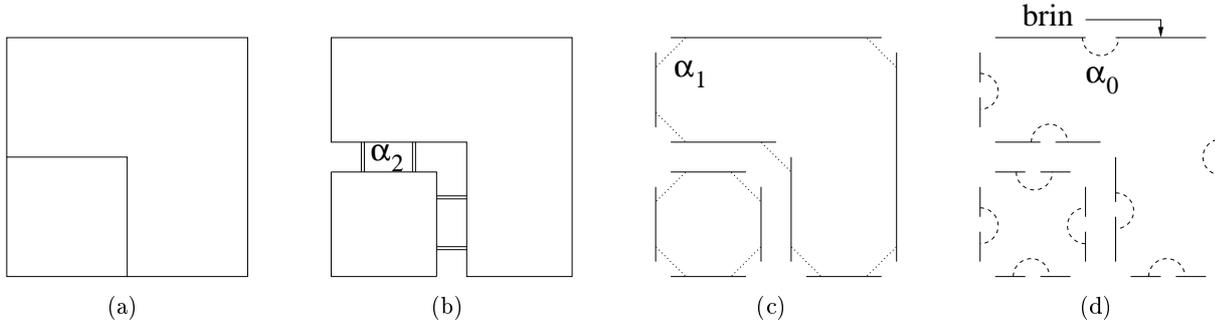


FIG. 4.1 – Exemple de décomposition topologique d'un objet en 2-G-carte : (a) Un objet euclidien en dimension 2, (b) Sa décomposition en faces topologiques, (c) Décomposition des faces en arêtes topologiques, (d) Décomposition des arêtes en brins.

Une n -G-carte peut soit couvrir une partie de l'espace dans lequel elle est définie, soit la totalité de celui-ci. On parle alors soit de n -G-carte avec bord, soit de n -G-carte sans bord (voir Figure 4.2).

Définition 24 Soit $G = (B, \alpha_0, \dots, \alpha_n)$ une n -G-carte. G est dite sans bord si et seulement si $\forall i \in [0, n]$ et $\forall b \in B$, $\alpha_i(b) \neq b$.

4.2.2 Cellules topologiques

A partir de la notion de brins, nous pouvons définir la notion d'orbite. De manière informelle, une orbite est un sous-ensemble de brins qui peut être obtenu en appliquant une composition donnée d'involutions sur un brin donné. Par exemple, si nous considérons un brin b , l'orbite $\langle \alpha_0, \alpha_1 \rangle (b)$ est l'ensemble des brins obtenu par n'importe quelle combinaison des involutions α_0 et α_1 appliquée au brin b (voir Figure 4.3c).

¹³Une involution f sur un ensemble fini E est une bijection de E dans E telle que $f = f^{-1}$.

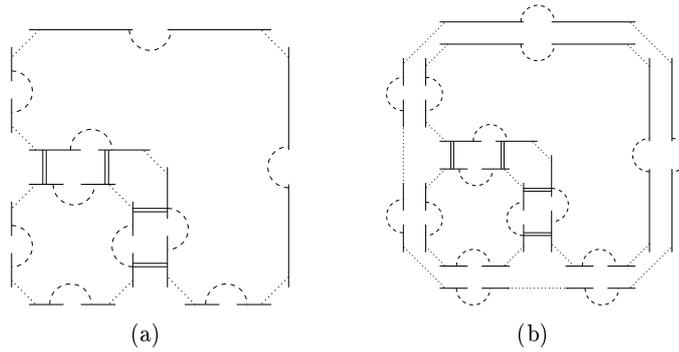
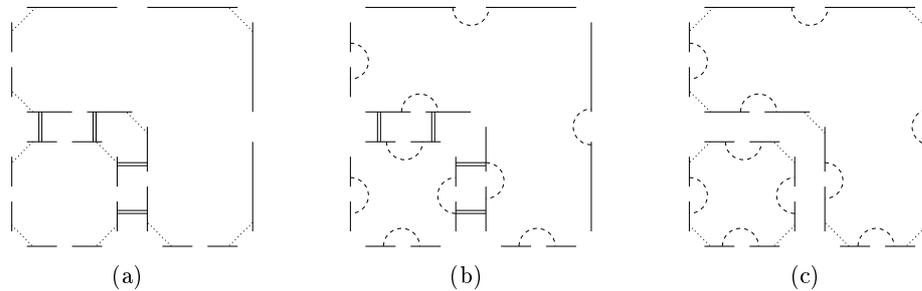


FIG. 4.2 – Exemple de 2-G-carte avec bord (a), et sans bord (b).

Dans une n -G-carte, chaque cellule topologique de dimension $i \in \llbracket 0, n \rrbracket$, appelée aussi i -cellule, correspond à une orbite particulière. Cette orbite, notée $\langle \alpha_0, \dots, \alpha_{i-1}, \alpha_{i+1}, \dots, \alpha_n \rangle$ est l'ensemble de brins qui peut être atteint en partant d'un brin donné, et en utilisant n'importe quelle combinaison d'involutions, exception faite de l'involution α_i .

Définition 25 (i -cellule) Soient $G = (B, \alpha_0, \dots, \alpha_n)$ une n -G-carte, b un brin de B et $i \in \llbracket 0, n \rrbracket$. La i -cellule incidente à b est l'orbite $\langle \alpha_0, \dots, \alpha_{i-1}, \alpha_{i+1}, \dots, \alpha_n \rangle$ (b).

Par exemple, dans une 2-G-carte, les orbites $\langle \alpha_1, \alpha_2 \rangle$, $\langle \alpha_0, \alpha_2 \rangle$ et $\langle \alpha_0, \alpha_1 \rangle$ correspondent respectivement aux sommets, arêtes et faces topologiques (voir Figure 4.3).


 FIG. 4.3 – Exemples de i -cellules d'un objet topologique (en gras sur la figure) : (a) Sommets topologiques, (b) Arêtes topologiques, (c) Faces topologiques.

Notons que, pour $i \in \llbracket 0, n \rrbracket$, l'ensemble des i -cellules d'une n -G-carte G constitue une partition de l'ensemble des brins de G .

4.2.3 Opérations de base

Une n -G-carte peut être construite à l'aide de trois opérations de base.

La première d'entre elles est l'ajout de brin, chaque brin ajouté étant invariant¹⁴ par toutes les involutions.

La deuxième est la couture par l'involution α_i , ou i -couture, de deux orbites $\langle \alpha_0, \dots, \alpha_{i-2}, \alpha_{i+2}, \dots, \alpha_n \rangle$. Par exemple, une 3-G-carte peut être construite par assemblage

¹⁴Soit b un brin de B , α_i , $i \in \llbracket 0, n \rrbracket$, un involution. b est invariant par α_i si $\alpha_i(b) = b$.

de deux volumes topologiques (orbites $\langle \alpha_0, \alpha_1, \alpha_2 \rangle$) en reliant deux faces (une face de chaque volume) par des involutions α_3 . Ceci n'est possible qu'en assemblant deux faces topologiques possédant la même structure, c'est-à-dire en cousant par α_3 deux orbites $\langle \alpha_0, \alpha_1 \rangle$ qui sont isomorphes (voir Figure 4.4).

Définition 26 (i-couture) Soit $G = (B, \alpha_0, \dots, \alpha_n)$ une n - G -carte, b et b' deux brins de B tels que b et b' sont invariants par α_i . La couture par α_i de b et b' peut être réalisée si et seulement si $\langle \alpha_0, \dots, \alpha_{i-2}, \alpha_{i+2}, \dots, \alpha_n \rangle (b)$ est isomorphe à $\langle \alpha_0, \dots, \alpha_{i-2}, \alpha_{i+2}, \dots, \alpha_n \rangle (b')$ par une application ϕ . Le résultat de cette opération est donc une n - G -carte $G' = (B, \alpha_0, \dots, \alpha_{i-1}, \alpha'_i, \alpha_{i+1}, \dots, \alpha_n)$ telle que :

- $\forall b'' \in \langle \alpha_0, \dots, \alpha_{i-2}, \alpha_{i+2}, \dots, \alpha_n \rangle (b)$ et $\forall b''' \in \langle \alpha_0, \dots, \alpha_{i-2}, \alpha_{i+2}, \dots, \alpha_n \rangle (b')$,
 $b''\alpha'_i = b''\phi$ et $b'''\alpha'_i = b'''\phi^{-1}$,
- $\forall b'' \notin \langle \alpha_0, \dots, \alpha_{i-2}, \alpha_{i+2}, \dots, \alpha_n \rangle (b) \cup \langle \alpha_0, \dots, \alpha_{i-2}, \alpha_{i+2}, \dots, \alpha_n \rangle (b')$, $b''\alpha'_i = b''\alpha_i$.

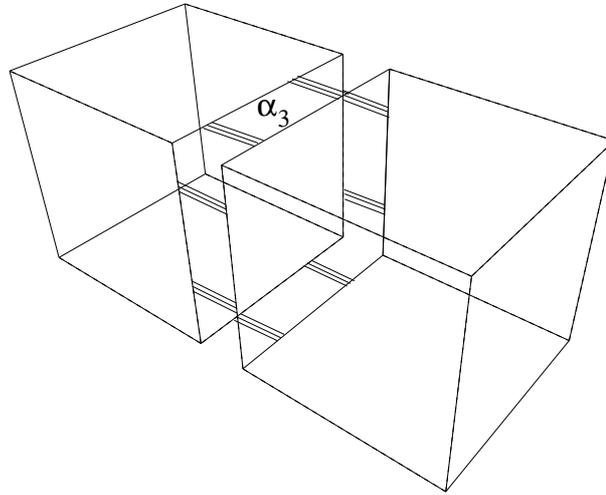


FIG. 4.4 – Exemple de 3-couture possible entre deux volumes topologiques. Les deux faces topologiques cousues sont isomorphes.

La dernière opération est l'opération de découverture par une involution α_i , $i \in \llbracket 0, n \rrbracket$, ou *i-découverte*. Celle-ci est effectuée en supprimant une liaison α_i existant entre deux brins. Aucune précondition sur les brins à découper n'est nécessaire.

Remarque 8 Dans la suite de ce mémoire, et dans le but de simplifier les figures, nous représenterons une 2- G -carte comme sur la figure 4.5.

4.3 Une structure hiérarchique

Dans cette section, nous décrivons la structure de données que nous avons conçue afin de pouvoir modéliser et modifier des objets géométriques représentés soit sous forme euclidienne, soit sous forme discrète.

Il s'agit d'une structure hiérarchique constituée de quatre niveaux différents (voir Figure 4.6), chaque niveau correspondant à une représentation particulière, euclidienne polygonale ou discrète, d'un même objet géométrique. Plus précisément, le niveau le plus élevé dans la structure

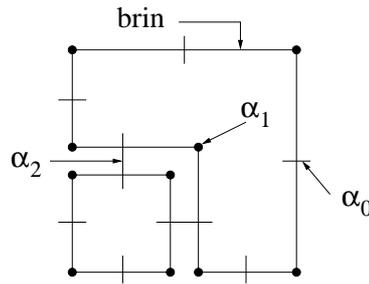


FIG. 4.5 – Représentation simplifiée d'une 2-G-carte.

correspond à la représentation euclidienne de l'objet, et chacun des trois autres niveaux correspond à une représentation discrète de ce même objet. Parmi les représentations discrètes, le niveau le plus bas est la représentation des objets sous forme de pixels (ou de voxels en dimension 3). Les deux niveaux intermédiaires, quant à eux, correspondent à deux représentations discrètes des frontières, ou *contours*, délimitant les différentes régions¹⁵ composant l'objet euclidien : une représentation implicite de ces contours, dite *analytique*, et une représentation explicite. Nous verrons par la suite que ces deux représentations possèdent des propriétés intéressantes qui leur sont propres.

Dans chaque niveau, l'objet géométrique correspondant est topologiquement modélisé par une 2-G-carte (voir Chapitre 4.2). De plus, divers types d'informations, principalement géométriques, sont éventuellement associées à la 2-G-carte de chaque niveau sous la forme de plongements, afin notamment de pouvoir visualiser les différentes représentations.

Dans le but de permettre la propagation locale d'éventuelles modifications appliquées à un niveau, des liaisons sont instaurées entre chaque 2-G-carte constituant un niveau et les 2-G-cartes la suivant et la précédant à l'intérieur de la structure. Concrètement, ces liaisons correspondent à des involutions supplémentaires établies entre certains brins constituant les différentes 2-G-cartes. Nous nommerons ces liaisons des liaisons γ . Ces liaisons sont aussi établies de manière à rendre "accessibles" aux autres niveaux les données géométriques stockées dans un niveau donné, et ce de manière à limiter la redondance d'information à l'intérieur de la structure.

Enfin, nous avons opté dans ce travail pour l'ajout de deux niveaux en sus des deux niveaux extrêmes. Il s'agissait d'obtenir une évolution "progressive" entre les différentes représentations. Nous entendons par progressif le fait de passer d'une représentation à une autre la suivant ou la précédant dans la structure sans engendrer de fortes modifications géométriques et topologiques de l'objet représenté. Notre structure a cependant été conçue de manière à pouvoir aisément ajouter ou supprimer un niveau donné, et ce en fonction des besoins.

Dans la suite de cette section, nous décrivons tout d'abord chaque niveau de notre structure, c'est à dire les différentes représentations des objets, tant au niveau topologique que géométrique. Puis nous donnons les différentes contraintes de cohérence devant être vérifiées au sein de la structure. Enfin, nous expliquons de quelle manière cette structure a été mise en œuvre à l'intérieur de SpaMod.

¹⁵Une région d'un objet géométrique est un ensemble de faces connexes possédant le même label.

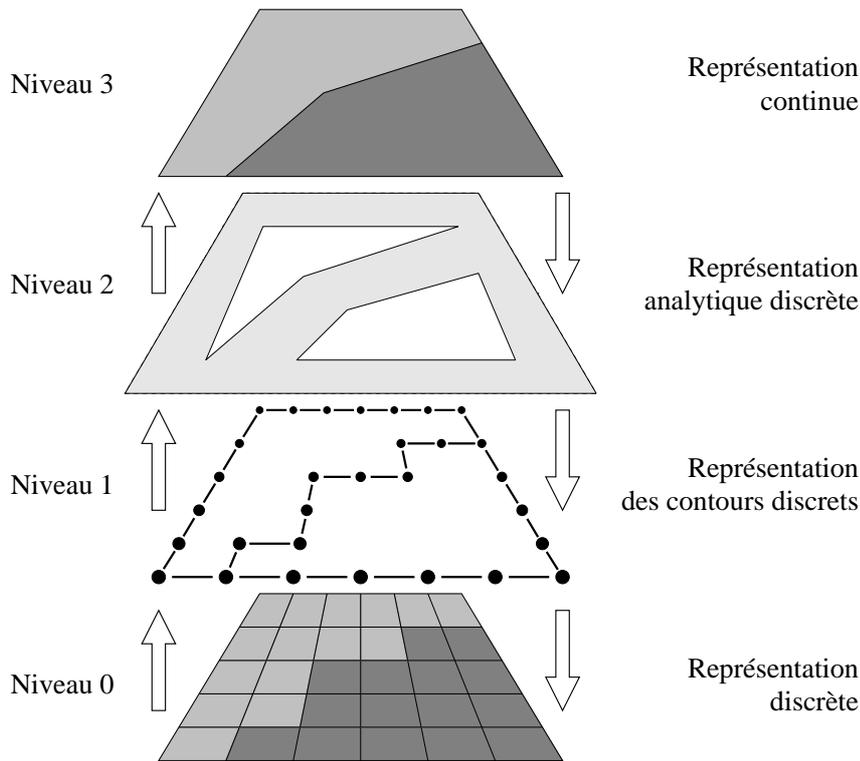


FIG. 4.6 – Illustration de la structure hiérarchique en dimension 2. Chaque niveau correspond à une représentation spécifique, discrète ou continue, d'un même objet géométrique. De plus, des liaisons bidirectionnelles sont instaurées entre deux niveaux consécutifs dans la structure.

4.3.1 Les niveaux

Chaque niveau de notre structure est une 2-G-carte sans bord (voir Définitions 23 et 24), à laquelle sont éventuellement associées des informations géométriques sous la forme de plongements. Toutes les 2-G-cartes correspondent à une subdivision de l'espace en faces topologiques. Notons que nous avons choisi de travailler avec des 2-G-cartes sans bord afin d'éviter le traitement de cas particuliers inhérents à l'existence d'un bord.

Dans ce qui suit, nous décrivons chacun des quatre niveaux en précisant tout d'abord de quelle représentation il s'agit, puis en détaillant les différents plongements associés à la 2-G-carte correspondante.

4.3.1.1 Niveau 3 : la représentation euclidienne

Ce niveau correspond à la représentation euclidienne polygonale des objets. Chaque région est décrite par un polygone euclidien auquel peut par exemple être associée une couleur (voir Figure 4.7a). L'objet de ce niveau peut être soit importé, soit obtenu par construction, à l'aide des outils présents dans le modéleur Moka.

Afin de pouvoir modéliser l'objet géométrique correspondant à ce niveau, deux types d'informations sont nécessaires : les coordonnées des sommets de l'objet géométrique et le cas échéant la couleur associée à chacune des faces le composant. Ainsi, des points à coordonnées réelles sont

associés à chaque sommet topologique de la 2-G-carte, et une couleur est éventuellement associée à chaque face topologique (voir Figure 4.7b).

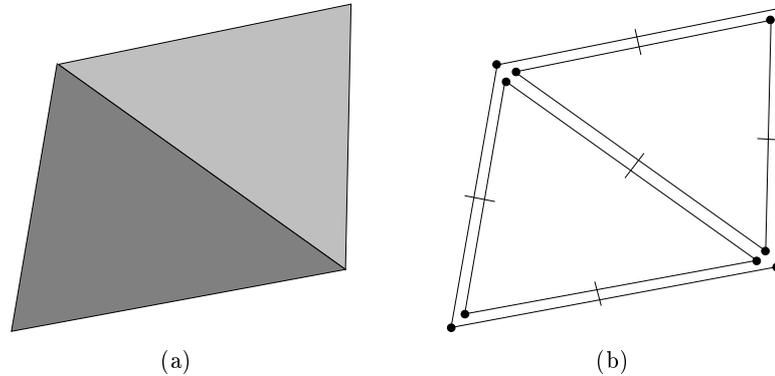


FIG. 4.7 – Le niveau euclidien. (a) Visualisation : objet composé de deux faces colorées adjacentes. (b) 2-G-carte : une couleur et des coordonnées à coefficients réels sont respectivement associées aux faces et sommets topologiques.

4.3.1.2 Niveau 2 : la représentation discrète analytique

L'objet de ce niveau est une description analytique dans le modèle discret Standard des contours d'un objet euclidien (voir Figure 4.8a). Plus précisément, chaque élément du contour (segment de droite) est décrit par un polygone analytique discret calculé suivant le modèle analytique discret Standard. L'intérêt de cette représentation est qu'il s'agit d'une représentation implicite de la discrétisation des contours. Elle ne dépend pas du nombre de points discrets la composant, contrairement à la représentation discrète classique (niveau 0). Ainsi, l'espace mémoire nécessaire au stockage de cette représentation est amoindri.

Afin de modéliser ce niveau, nous devons connaître les inéquations correspondant à la description analytique discrète Standard des segments des contours. Ces inéquations sont calculées pour chaque sommet et chaque arête de l'objet euclidien, et sont ensuite stockées dans les sommets et arêtes topologiques de la 2-G-carte du niveau (voir Figure 4.8b). Plus précisément, deux inéquations, correspondant à la description analytique Standard de la droite porteuse d'une arête, sont associées à chaque arête topologique. De même, quatre inéquations, correspondant à la description analytique de la discrétisation Standard d'un point euclidien, sont associées à chaque sommet topologique.

4.3.1.3 Niveau 1 : les contours discrets

Ce niveau correspond aux frontières entre les régions discrètes contenues dans une image (voir Figure 4.9a). Nous entendons par région tout sous-ensemble de pixels 1-connexes possédant une même couleur. Le modèle que nous utilisons pour modéliser ces contours est le modèle *interpixels* [KKM90a, Kov89]. Avec ce modèle, chaque point discret du contour est modélisé par un *pointel*, et les relations adjacence entre deux pointels sont modélisées par un *lignel*. L'avantage de cette représentation est de nous permettre de disposer d'informations topologiques de haut niveau concernant la représentation discrète, comme les voisinages entre régions.

Afin de modéliser cet objet, il est nécessaire de connaître les coordonnées des pointels le composant. Or, ces coordonnées sont déjà stockées au niveau 0, puisque les mêmes points discrets

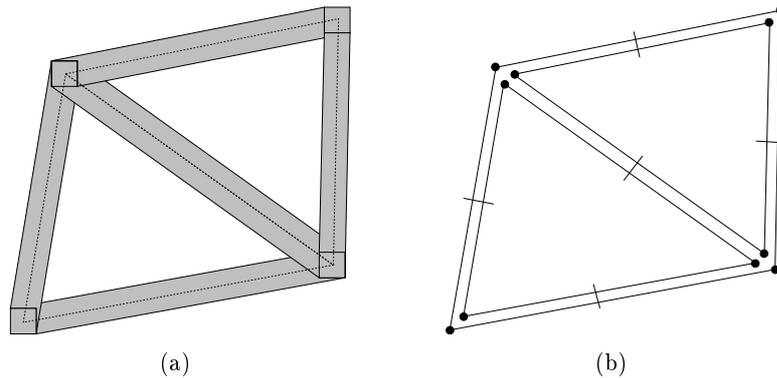


FIG. 4.8 – Le niveau discret analytique. (a) Visualisation : un objet euclidien (en pointillés) et la représentation analytique discrète Standard de ses contours (zones grisées). (b) 2-G-carte : quatre (resp. deux) inéquations sont associées à chaque sommet (resp. arête) topologique.

y sont aussi présents. Il est alors possible d'accéder à ces coordonnées en exploitant les liaisons existant entre le niveau 1 et le niveau 0. Ainsi, afin de ne pas surcharger la structure avec des informations redondantes, aucun plongement n'est associé à la 2-G-carte de ce niveau (voir Figure 4.9b). Ceci est possible car les informations géométriques nécessaires à la modélisation de ce niveau, à savoir les coordonnées des sommets de pixels et voxels sont situées au niveau 0 et sont accessibles à partir du niveau 1.

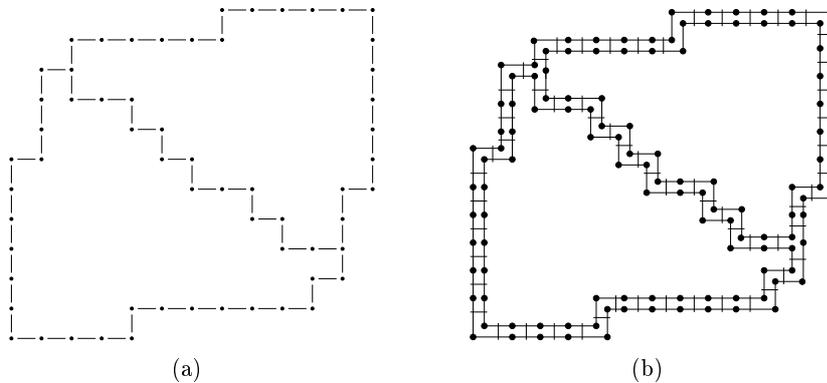


FIG. 4.9 – Les contours discrets : (a) Visualisation : chaque point discret est représenté par un pointel, et deux pointels successifs sont reliés par un lignel. (b) 2-G-carte : aucun plongement n'est associé.

4.3.1.4 Niveau 0 : représentation discrète classique duale

Ce niveau correspond à la représentation discrète classique des objets, c'est à dire sous forme de pixels (voir Figure 4.10a). Nous nous plaçons ici dans l'espace discret dual, c'est à dire l'espace dans lequel les points discrets se trouvent non pas au centre des pixels mais à leurs sommets. Les objets de ce niveau peuvent être des images importées ou créées grâce aux outils disponibles dans le modéleur.

La visualisation de ce niveau se fait en affichant les pixels composant l'objet discret. Un

pixel étant uniquement déterminé par une position dans l'espace et un label (par exemple, une couleur), deux types de plongements sont associés à la 2-G-carte de ce niveau (voir Figure 4.10b). Le premier est une couleur que nous associons à chaque face topologique. Les autres plongements sont des coordonnées entières que nous associons à chaque sommet topologique.

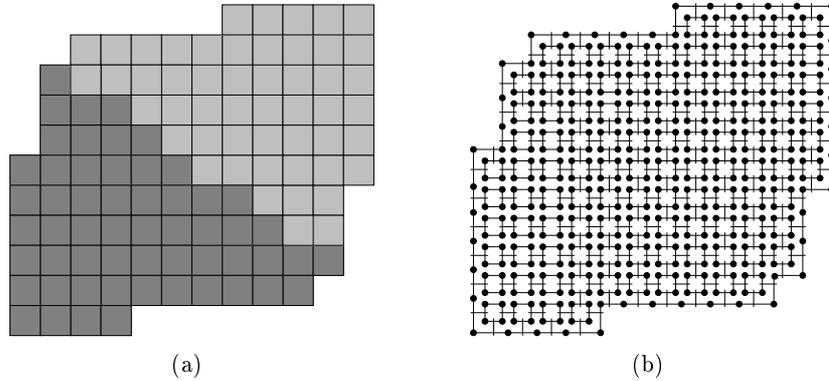


FIG. 4.10 – Le niveau discret : (a) Visualisation : une image en niveaux de gris. (b) 2-G-carte : un point de coordonnées entières et une couleur sont respectivement associés à chaque sommet et face topologiques.

4.3.1.5 Mise en œuvre

La structure complète, à savoir les différents niveaux, ainsi que les liaisons entre les niveaux ont été mis en œuvre grâce au noyau de cartes généralisées présent dans le modèleur Moka. Cependant, l'espace mémoire nécessaire au stockage du niveau 0 sous la forme d'une G-carte étant très important, et notamment en dimension 3, ce niveau a été implémenté de façon plus économique¹⁶.

En effet, l'espace mémoire nécessaire au stockage d'une image constituée de 512^2 pixels sous la forme d'une G-carte est de l'ordre de 100 Mo, alors que pour une image en dimension 3 constituée de 512^3 voxels, l'espace nécessaire est proche de 300 Go.

Ainsi, dans le but de minimiser l'espace mémoire nécessaire, la 2-G-carte du niveau 0 est simulée à l'aide de deux matrices :

- La première matrice simule les sommets topologiques du niveau 0. Plus précisément, chaque case de la matrice simule un sommet topologique du niveau (voir Figure 4.11), ce qui suffit à simuler l'intégralité de la 2-G-carte, puisque l'ensemble des sommets topologiques forme une partition de l'ensemble de ses brins. De plus, afin de diminuer d'autant plus l'espace mémoire utilisé, seul un brin sur deux est simulé. Plus précisément, pour tout couple de brins cousus par une liaison α_2 , seul un brin est simulé. Ainsi, chaque case de la matrice contient un tableau de quatre cases, chaque case étant dédiée à un brin du même sommet. Les liaisons α_0 et α_1 entre les brins sont déduites des positions respectives des cases de la matrice par rapport à leurs voisines (α_0), ainsi que des positions des cases des tableaux (α_1).

Les plongements correspondant aux coordonnées des points discrets sont directement déduits des indices des cases de la matrice. Leur stockage n'est donc pas nécessaire.

¹⁶Ce travail a été effectué en collaboration avec A. FOUSSE, Maître de conférences, Laboratoire SIC.

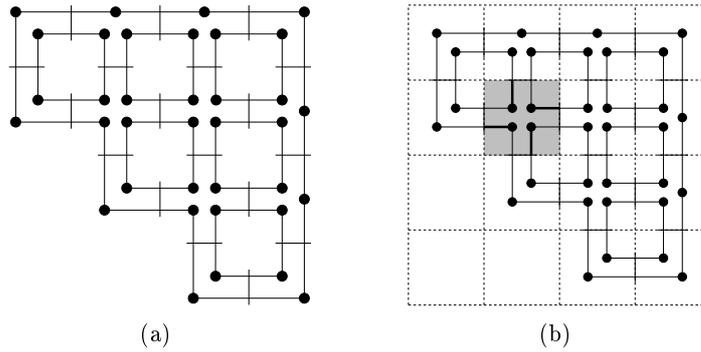


FIG. 4.11 – Exemple de 2-G-carte simulée : (a) 2-G-carte. (b) Chaque case de la matrice (traits pointillés) simule quatre brins (en gras sur la figure) d'un sommet topologique.

- La deuxième matrice contient uniquement les couleurs associées aux faces topologiques de la 2-G-carte.

Ces améliorations permettent de réduire de manière conséquente la taille de l'espace mémoire nécessaire au stockage du niveau 0. Par exemple, une image de dimension 2 constituée de 512^2 pixels sera stockée dans 1 Mo, alors qu'une image de dimension 3 composée de 512^2 voxels ne nécessitera plus que 300 Mo.

4.3.2 Contraintes de cohérence

Cette structure hiérarchique a été conçue afin de pouvoir manipuler un même objet géométrique représenté sous différentes formes. Les différents niveaux de la structure doivent donc vérifier plusieurs contraintes afin de rester cohérents entre eux. Ces contraintes sont les suivantes :

- Le niveau 1 correspond aux frontières entre les régions du niveau 0 (voir Figure 4.12).

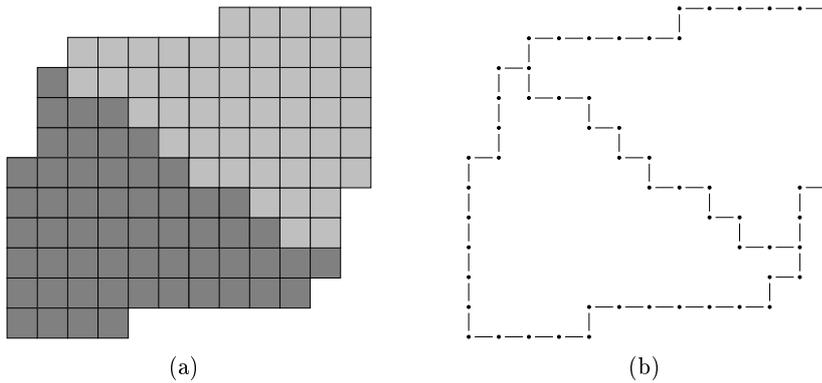


FIG. 4.12 – Illustration de la première contrainte de cohérence : (a) Niveau 0. (b) Niveau 1.

- Le niveau 2 correspond à la description analytique discrète Standard des contours du niveau 3 (voir Figure 4.13).

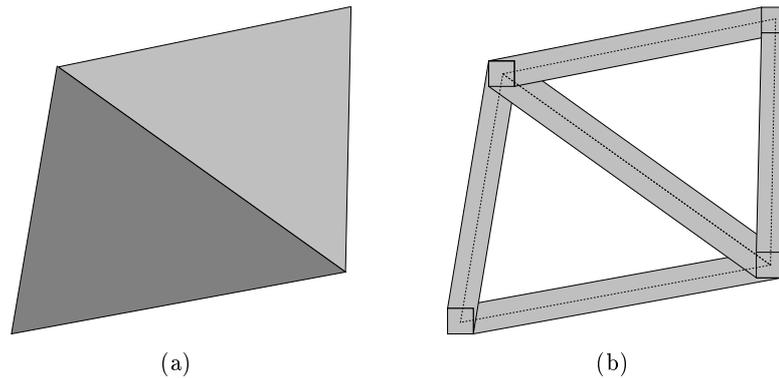


FIG. 4.13 – Illustration de la deuxième contrainte de cohérence : (a) Niveau 3. (b) Niveau 2.

- La discrétisation Standard des contours de l'objet euclidien doit contenir les points discrets du niveau 1. Cette propriété est équivalente à la propriété suivante : la description analytique du niveau 2 contient tous les points discrets du niveau 1 (voir Figure 4.14). Nous verrons dans le chapitre suivant que la propriété réciproque n'est pas nécessairement vérifiée.

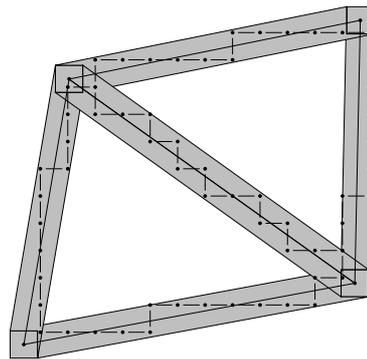


FIG. 4.14 – Illustration de la troisième contrainte de cohérence : tous les points discrets du niveau 1 sont contenus dans l'objet décrit analytiquement au niveau 2.

Remarque 9 *Nous nous sommes limités à la présentation de notre structure en dimension 2. La définition de cette structure peut cependant être aisément étendue aux dimensions supérieures, et en particulier à la dimension 3. En effet, notre structure est uniquement basée sur le modèle des cartes généralisées qui, nous l'avons vu, est défini en toutes dimensions.*

4.4 Opérations dans la structure

Les différents niveaux de notre structure ayant été définis dans le chapitre précédent, nous nous intéressons à présent aux opérations nécessaires à la construction et la modification de ces niveaux.

En effet, dans un premier temps, les différents niveaux de la structure doivent être construits. Pour cela, nous avons supposé que nous disposions d'un objet géométrique sous une certaine forme, et avons cherché à en déduire les autres représentations. Nous avons donc développé un

ensemble d'opérations de construction nous permettant d'obtenir l'ensemble des niveaux de la structure à partir d'un niveau donné. Ces opérations permettent de plus l'établissement des liaisons entre les différents niveaux. Dans ce travail, nous nous sommes principalement penchés sur la construction des niveaux à partir d'un objet euclidien (niveau 3), ou à partir d'un objet discret (niveau 0). En effet, nous verrons dans la section suivante que la construction d'un niveau peut se faire soit à partir du niveau le suivant, soit à partir du niveau le précédant à l'intérieur de la structure. Ainsi, la construction de la structure à partir de l'un des deux niveaux intermédiaires peut être effectuée en exploitant certaines étapes des processus de construction appliqués aux niveaux extrémités.

Dans un deuxième temps, la structure doit pouvoir être modifiée tout en conservant la cohérence des quatre niveaux (voir Section 4.3.2). Nous nous sommes donc intéressés à la manière dont les différentes représentations peuvent être mises à jour suite à la modification de l'une d'entre elles. Nous nous sommes notamment penchés sur le problème de la mise à jour locale des niveaux, c'est à dire une mise à jour effectuée suite à une modification n'affectant qu'une partie de la représentation. L'intérêt d'une mise à jour locale est de conserver les détails présents dans la partie non modifiée de la représentation d'origine, alors que ces détails peuvent être altérés lors d'une reconstruction complète de la structure. Nous verrons de quelle manière cette mise à jour peut être effectuée en exploitant les liaisons topologiques existant entre les niveaux.

Dans cette section, nous présentons tout d'abord les opérations de construction de la structure, que ce soit à partir d'un objet euclidien, ou à partir d'un objet discret. Nous décrivons les étapes nécessaires à la construction des différents niveaux de la structure, tant au niveau topologique, que géométrique. Nous expliquons en particulier de quelle manière les liaisons sont mises en place entre les niveaux. Nous abordons ensuite le problème de la mise à jour de la structure en cas de modification d'un niveau. Dans la suite, nous décrivons les opérations sans prendre en compte la modification du niveau 0 que nous avons mise en œuvre. Nous considérerons donc que le niveau 0 de la structure est une 2-G-carte (voir Section 4.3.1.1).

4.4.1 Construction des niveaux et mise en place des liaisons

Nous nous intéressons ici à la construction des différents niveaux de la structure, que ce soit à partir d'un objet euclidien ou à partir d'un objet discret. En particulier, nous décrivons la manière dont chaque niveau est construit à partir d'un niveau voisin. Nous détaillons principalement les opérations topologiques nécessaires à la construction du niveau ainsi que la manière dont les liaisons sont mises en place durant la construction. La construction topologique des niveaux est basée sur l'utilisation d'opérations appliquées aux cartes généralisées des différents niveaux. Ces opérations, appelées *contraction* et *suppression* de cellules ont été spécifiées par G. DAMIAND et P. LIENHARDT dans [DL03]. Nous ne les détaillerons pas ici, mais nous contenterons de les décrire de manière intuitive.

Dans la suite, nous décrivons en premier lieu la construction de la structure à partir d'un objet euclidien, puis à partir d'un objet discret.

4.4.1.1 Discrétisation d'un objet euclidien

La construction des niveaux de notre structure à partir d'un objet euclidien s'effectue en trois étapes : le calcul de la représentation discrète analytique, la discrétisation Standard des contours, et enfin, le calcul de la représentation discrète sous forme de pixels de l'objet euclidien.

Dans toute cette section, nous illustrons nos propos en nous appuyant notamment sur l'objet euclidien présent sur la figure 4.15.

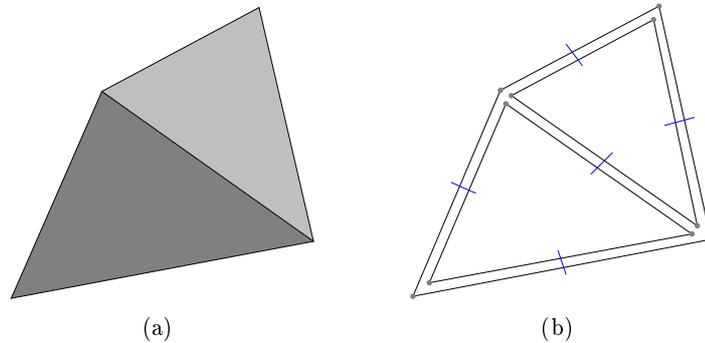


FIG. 4.15 – Exemple d'objet euclidien : (a) Représentation géométrique. (b) Représentation topologique.

La représentation discrète analytique des contours

Le niveau 2 de la structure est obtenu en déterminant un ensemble d'inéquations décrivant la discrétisation Standard des contours de l'objet euclidien (voir Figure 4.16), puis en les associant à la 2-G-carte du niveau. L'ensemble de ces inéquations, que nous appellerons *description discrète analytique*, est calculé à partir des coordonnées des sommets de l'objet euclidien.

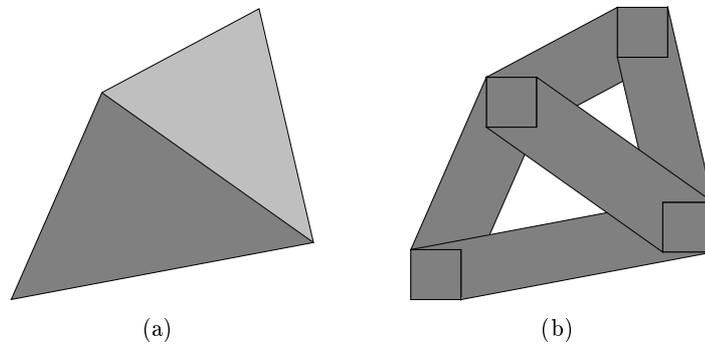


FIG. 4.16 – (a) Objet euclidien. (b) Représentation discrète analytique correspondante.

Les 2-G-cartes présentes aux niveaux 2 et 3 sont en règle générale topologiquement identiques (voir Figure 4.17). Seuls les plongements associés diffèrent. La construction du niveau 2 se limite donc essentiellement à recopier la 2-G-carte du niveau 3, et à stocker par la suite les différents plongements. De plus, des liaisons sont établies entre les brins correspondants dans chacune des 2-G-cartes, comme nous pouvons le voir sur la figure 4.17. Tous les brins des deux 2-G-cartes sont alors reliés.

Cependant, nous avons remarqué que dans certains cas de figure, la 2-G-carte du niveau 2 peut éventuellement être simplifiée, et ce dans le but de diminuer le nombre de calculs. En effet, nous pouvons voir sur l'exemple de la figure 4.18a que les huit sommets de l'objet euclidien (en gris clair sur la figure) situés dans le pixel coloré en gris foncé possèdent la même discrétisation Standard

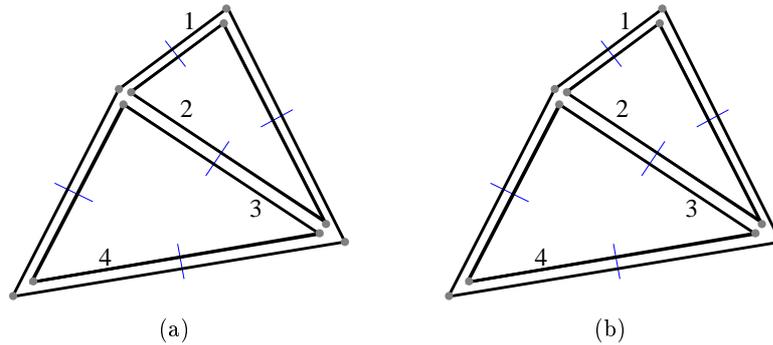


FIG. 4.17 – 2-G-cartes et liaisons entre le niveau 2 et le niveau 3. Les brins en gras sont reliés. Par exemple, les brins numérotés 1 (resp. 2, 3 et 4) dans les deux représentations sont reliés. Sur cet exemple, il y a exactement 20 liaisons entre les deux niveaux.

(ici, le pixel gris foncé). Ainsi, il est possible d'obtenir une description discrète analytique de tout l'objet (voir Figure 4.18b) sans nécessairement calculer une description discrète analytique des arêtes contenues dans le pixel. Nous voyons alors que si nous simplifions l'objet de la figure 4.18a en l'objet de la figure 4.18c, la discrétisation Standard du nouvel objet obtenu est identique à celle de l'objet initial.

Nous avons donc choisi de simplifier la 2-G-carte du niveau 2 (voir Figure 4.19a) de la manière suivante : dès lors que plus de deux sommets géométriques consécutifs de l'objet euclidien possèdent la même discrétisation Standard, tous les sommets topologiques correspondants, exceptés les deux extrémités, sont supprimés. Pour ce faire, pour chaque sommet intermédiaire (voir Figure 4.19b), nous appliquons une opération de suppression de sommet topologique [DL03], et ce jusqu'à obtenir une arête unique entre les deux sommets extrémités, comme nous pouvons le voir sur la figure 4.19c.

Dans l'exemple de la figure 4.19, la suppression du sommet se fait en décousant les quatre brins du sommet par α_0 , puis en cousant les brins décousus n'appartenant pas au sommet par α_1 , comme nous pouvons le voir sur la figure 4.20.

Notons que dans le cas où la G-carte du niveau 2 a été simplifiée, certains brins de la G-carte du niveau 3 ne sont pas reliés (voir Figure 4.21).

Enfin, la 2-G-carte du niveau étant construite, les différents plongements sont associés. Ainsi, pour chaque sommet et arête topologique du niveau 2, les inéquations correspondantes sont calculées à partir des coordonnées stockées dans les sommets topologiques du niveau 3. Pour chaque sommet, celles-ci sont obtenues en utilisant les liaisons existant entre un brin du sommet au niveau 2 et le brin correspondant au niveau 3.

La discrétisation Standard des contours

La construction du niveau 1 est réalisée en effectuant une discrétisation incrémentale suivant le modèle Standard des contours de l'objet euclidien (voir Figure 4.22).

La construction de la 2-G-carte du niveau 1 se fait à partir de celle du niveau 2. Tout d'abord, la G-carte du niveau 2 est recopiée, et des liaisons sont mises en place entre les brins correspondants des deux 2-G-cartes (voir Figure 4.23).

Puis, la discrétisation incrémentale Standard de chaque arête géométrique est effectuée en partant d'un des deux sommets extrémités de l'arête. Les coordonnées des sommets géométriques

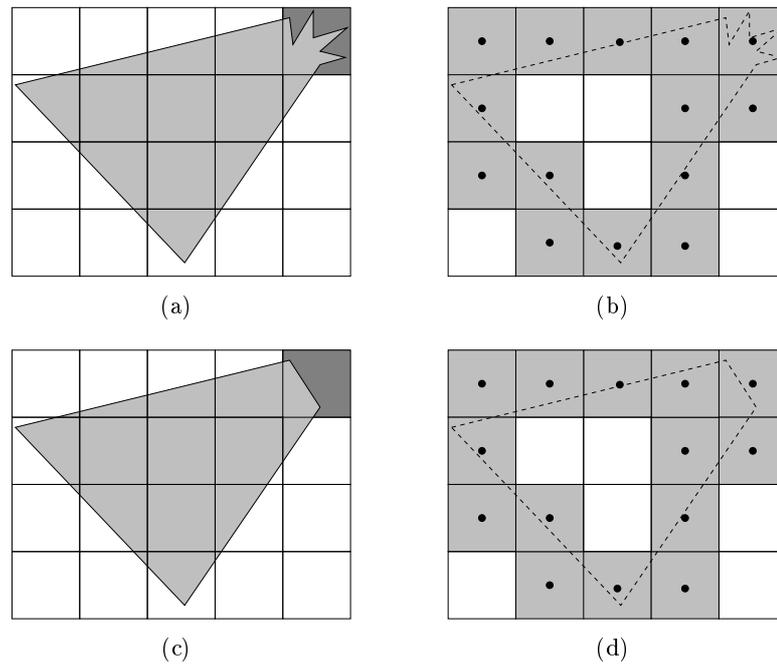


FIG. 4.18 – Simplification du niveau 2 : (a) Objet euclidien, (b) Discrétisation Standard de l'objet (a), (c) Simplification de l'objet (a). (d) Discrétisation de l'objet (c).

nécessaires au calcul des points discrets, présentes au niveau 3, sont obtenues en utilisant les liaisons existant entre les niveaux 1, 2 et 3. Dès qu'un nouveau point discret est calculé, un sommet topologique est inséré sur l'arête séparant le sommet auquel est associé le point discret précédemment calculé et le sommet extrémité de l'arête topologique (voir Figure 4.24b).

L'insertion d'un sommet topologique se fait de la manière suivante : tout d'abord, les quatre brins de l'arête dans laquelle doit être inséré le sommet sont décousus par α_0 . Puis, quatre nouveaux brins sont créés et cousus par des liaisons α_0 et α_2 . Enfin, ces brins sont cousus aux brins de l'arête à l'aide d'une liaisons α_1 .

Nous pouvons voir sur la figure 4.24c un exemple de 2-G-carte niveau 1 obtenu. Suite à l'insertion des sommets topologiques, les liaisons existant entre les 2-G-cartes des niveaux 1 et 2

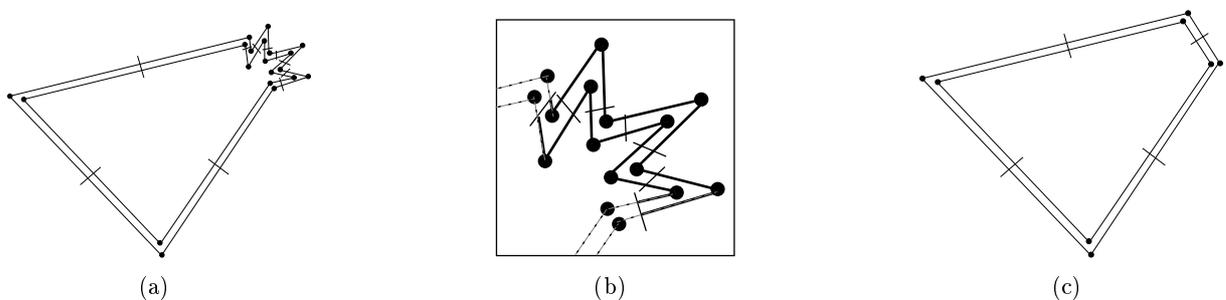


FIG. 4.19 – Simplification du niveau discret analytique. (a) 2-G-carte du niveau 3. (b) Les sommets en gras sont supprimés. (c) 2-G-carte du niveau 2 obtenue.

FIG. 4.20 – Suppression d'un sommet topologique.

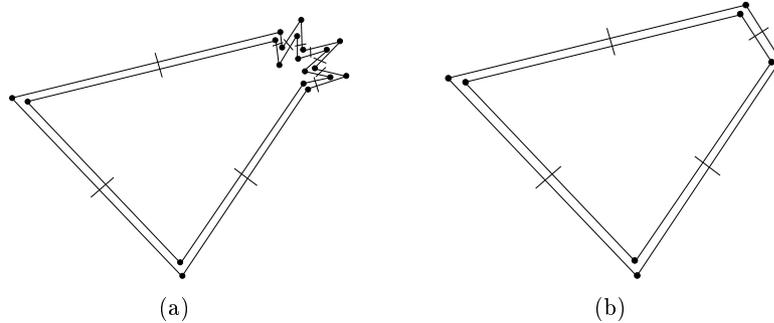


FIG. 4.21 – Liaisons entre le niveau 2 et le niveau 3 (cas d'une simplification). Les brins en gras sont reliés, alors que les brins en pointillés ne le sont pas.

sont conservées. Les autres brins du niveau 1 ne sont pas reliés.

Cependant, les objets obtenus suite à la discrétisation Standard des contours de l'objet euclidien peuvent parfois poser problème. En effet, comme nous pouvons le voir sur la figure 4.25, le fait que certains points de la discrétisation soient identiques génère des particularités géométriques et topologiques que nous appellerons des *faces dégénérées*. Ces faces peuvent être soit entièrement dégénérées (voir Figure 4.25d), soit partiellement (voir Figures 4.25a, 4.25b et 4.25c), faisant alors apparaître des particularités que nous nommerons des *étranglements*.

Deux options se sont donc posées à nous quant à la conservation de ces faces et de ces étranglements tant du point de vue géométrique que topologique de la représentation. Le fait de supprimer ces particularités peut éventuellement entraîner une perte d'information. Par exemple, comme nous pouvons le voir sur la figure 4.26, si nous supprimons l'étranglement, nous obtenons deux faces distinctes. Nous avons alors perdu l'information relative à l'origine de ces deux faces. Cependant, le fait de conserver ces informations peut tout autant poser des problèmes d'interprétation et d'exploitation de l'information. Par exemple, nous pouvons voir sur la figure 4.26

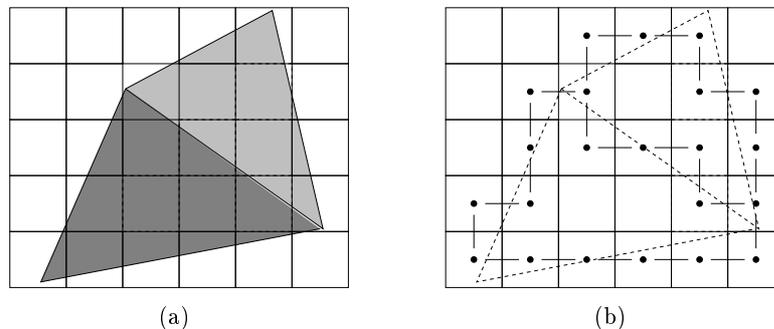


FIG. 4.22 – Discrétisation Standard des contours : (a) Objet euclidien. (b) Discrétisation obtenue représentée à l'aide de pointels et lignels.

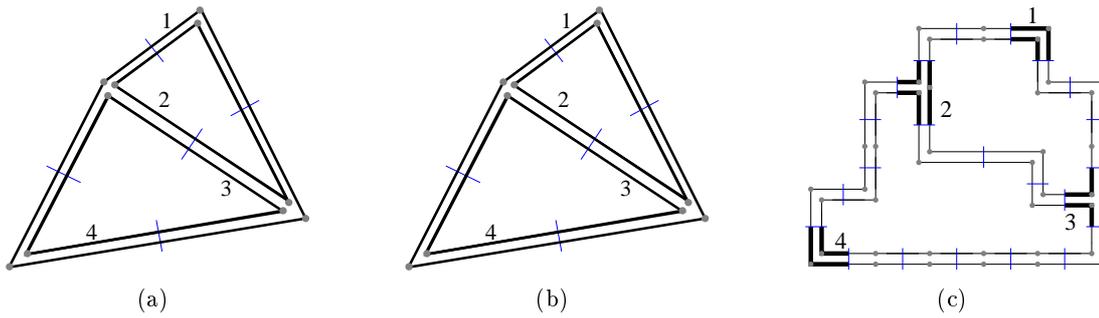


FIG. 4.23 – 2-G-cartes et liaisons entre les niveaux 1 et 2.

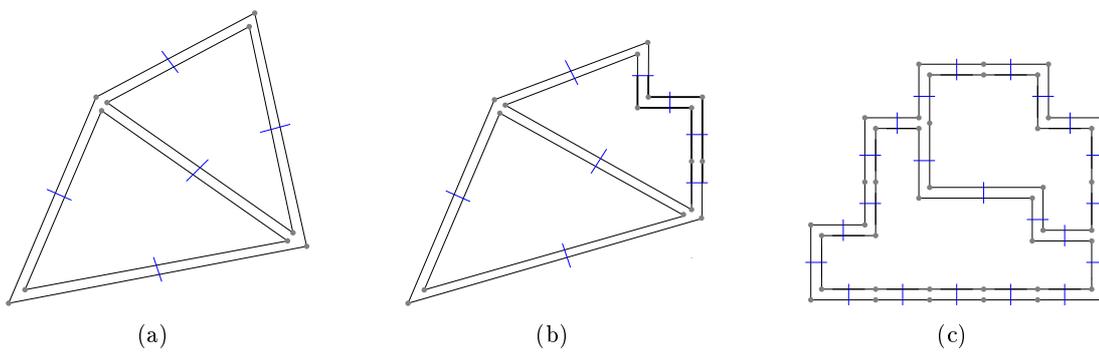


FIG. 4.24 – Construction de la G-carte. Niveau 1 en cours de construction après la discrétisation Standard d'une arête.

un exemple d'étranglement topologiquement difficilement exploitable. Nous avons alors choisi de supprimer ces particularités du niveau 1.

Pour ce faire, les sommets topologiques correspondant à des points discrets identiques sont déterminés. Puis, les arêtes comprises entre ces sommets sont contractées [DL03].

Suite à la disparition des étranglements, certaines liaisons préexistantes ont parfois de même disparues. Nous avons alors choisi, afin de conserver ces liaisons, de les déplacer préalablement. Celles-ci sont déplacées le long de l'étranglement, jusqu'à atteindre une arête ne devant pas être contractée. Dans le cas d'une face entièrement dégénérée, les liaisons existant entre les brins de cette face et le niveau 2 ne sont pas conservées. Cela nous permet de conserver une information sur le provenance des sommets supprimés. Une illustration de cette opération est montrée sur la figure 4.27.

Enfin, des plongements correspondant aux coordonnées entières des points discrets sont momentanément associés à chaque sommet topologique de la G-carte.

Des contours à l'objet discret

Le niveau 0 est obtenu en déterminant quels sont les pixels se trouvant à l'intérieur des contours du niveau 1. Ceux-ci peuvent par exemple être obtenus en appliquant un algorithme de remplissage à une matrice représentant le niveau 1. Les points discrets se trouvent alors placés sur les sommets des pixels (voir Figure 4.28).

La construction de la 2-G-carte de ce niveau est effectuée en deux étapes. Tout d'abord, la

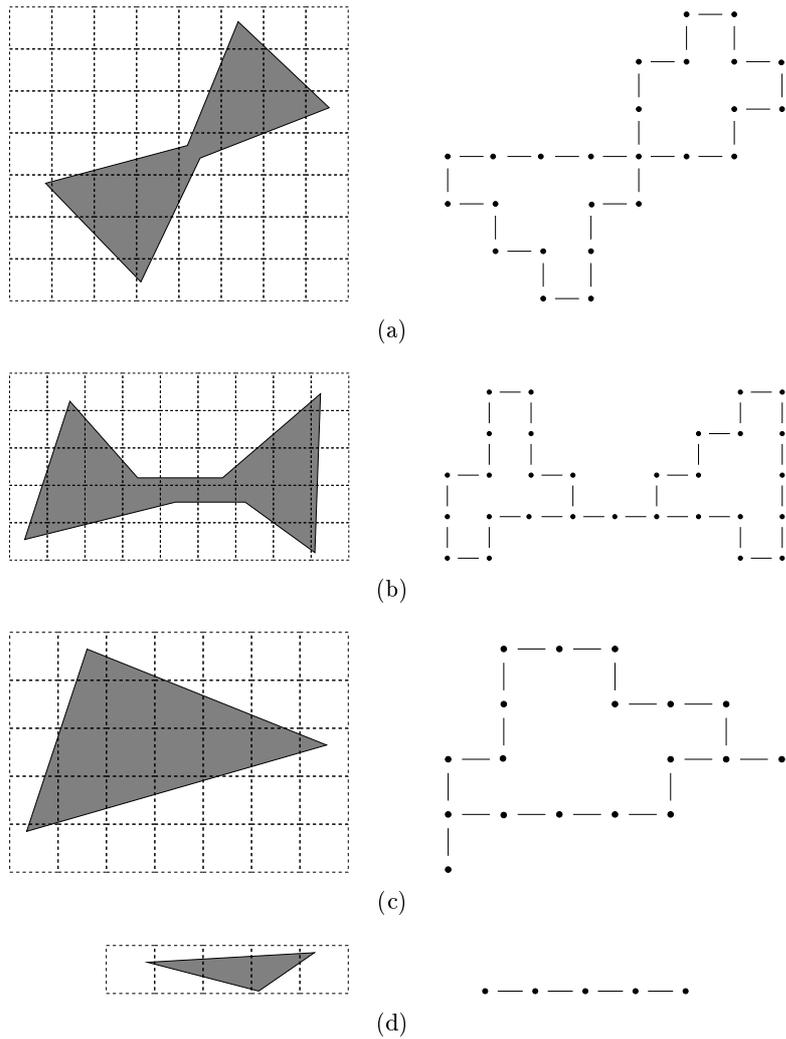


FIG. 4.25 – Exemples de faces dégénérées : (a), (b) et (c) Divers étranglements possibles. (d) Face entièrement dégénérée.

2-G-carte du niveau 1 est recopiée et les liaisons mises en place entre les brins correspondants des deux 2-G-cartes.

Puis, pour chaque pixel déterminé, une face topologique composée de quatre arêtes et cousue par des liaisons α_2 à la 2-G-carte du niveau. Les plongements correspondants, à savoir la couleur du pixel, ainsi que les coordonnées des sommets sont ensuite associés. Notons que la couleur du pixel doit correspondre à la couleur associée à la face dont le pixel appartient à la discrétisation. De plus les coordonnées des sommets sont récupérées en accédant aux coordonnées stockées dans le niveau 1 via les liaisons γ entre les deux niveaux. Ces dernières sont alors supprimées du niveau 1, puisque déjà stockées au niveau 0 et accessibles à partir de n'importe quel niveau via les liaisons γ .

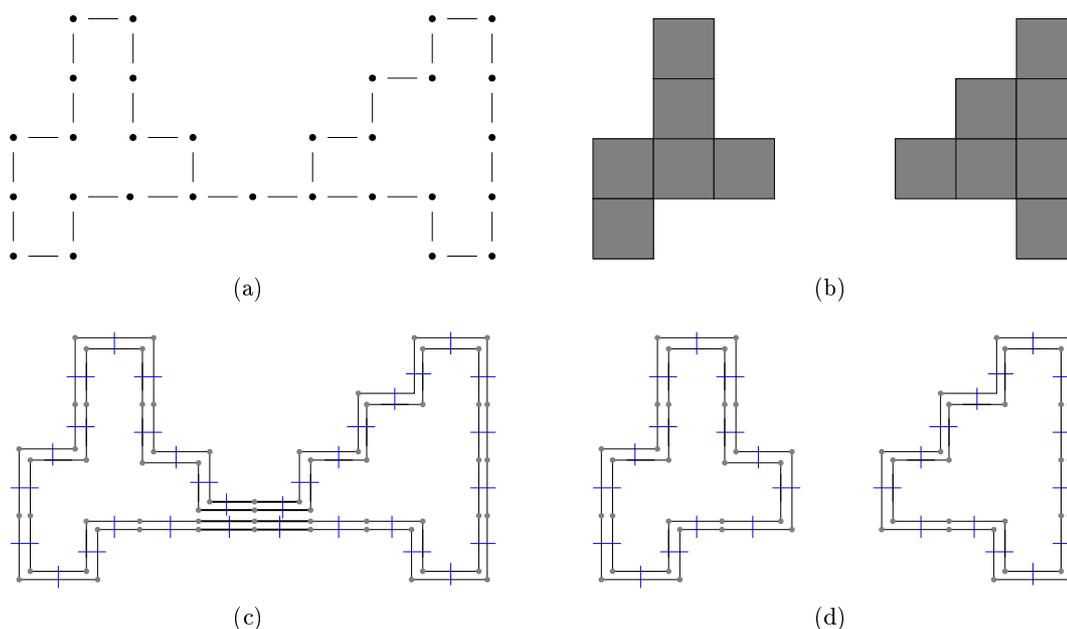


FIG. 4.26 – Exemple de déconnexion de faces.

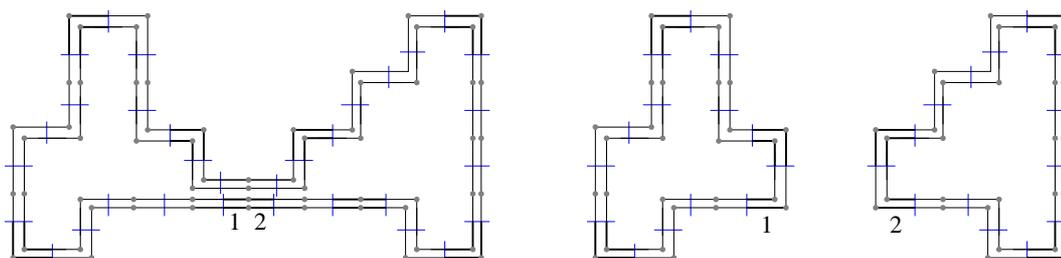


FIG. 4.27 – Exemple de déplacement des liaisons avant la suppression de brins reliés.

4.4.1.2 Reconstruction d'un objet discret

Nous nous intéressons à présent à la construction de notre structure à partir d'un objet discret. Nous détaillons les deux étapes nécessaires, la première pour construire le niveau 1, et la deuxième pour construire les niveaux 2 et 3 simultanément. Le niveau 1 est obtenu en extrayant les contours de la représentation discrète du niveau 0. Les niveaux 2 et 3 sont obtenus en effectuant une reconstruction (voir Partie 2) des contours du niveau 1, cette reconstruction devant être inversible pour le modèle Standard.

Extraction des contours de l'objet discret

Le niveau 1 est obtenu en extrayant les contours du niveau 0 se trouvant entre deux ensembles de pixels 1-connexes possédant la même couleur.

La 2-G-carte du niveau 1 peut être obtenue de la manière suivante : tout d'abord, la 2-G-carte du niveau 0 est recopiée et des liaisons γ mises en place entre les brins des deux 2-G-cartes. Puis, les faces topologiques adjacentes auxquelles est associée la même couleur sont fusionnées.

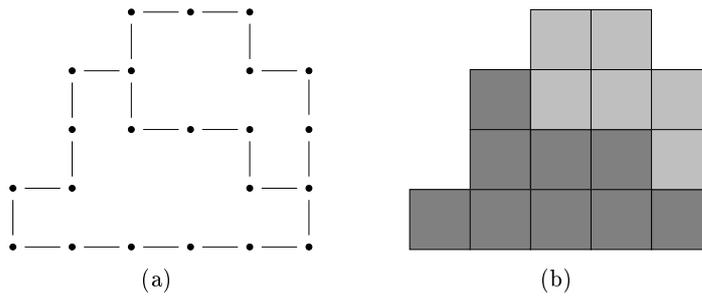


FIG. 4.28 – Exemples de niveaux 0 et 1.

La fusion de deux faces topologiques est obtenue en appliquant une opération de suppression sur l'arête commune aux deux faces. Suite à la fusion des diverses faces, certaines liaisons γ sont alors supprimées.

La reconstruction d'un objet continu

Les niveaux 2 et 3 se construisent lors de la même opération. En effet, lorsque l'objet initial est un objet discret, les niveaux 2 et 3 sont topologiquement identiques. Il s'agit d'utiliser une méthode de reconstruction, telles que celles que nous avons décrites dans la partie 2, sur les contours discrets du niveau 1, et ce de façon à obtenir une représentation polygonale de ces contours.

La construction de la G-carte du niveau 2 peut être fait de la manière suivante : tout d'abord, la G-carte du niveau 1 est recopiée, et des liaisons γ sont mises en place entre les brins correspondants des deux 2-G-cartes. Puis, lors du processus de reconnaissance des segments discrets, les sommets topologiques correspondants aux points discrets reconnus sont supprimés. Ceci s'effectue de la même manière que dans le cas de la simplification du niveau 2 (voir Section 4.4.1.1).

Les liaisons entre les deux niveaux sont donc les liaisons qui ont été conservées lors du processus de suppression des sommets topologiques, c'est à dire les liaisons avec les brins extrémités des segments reconnus. Les autres brins du niveau 1 ne sont pas reliés.

Le niveau 3 est obtenu en recopiant la 2-G-carte du niveau 2 et en établissant des liaisons γ entre les 2-G-cartes des deux niveaux.

Enfin, des plongements sont associés aux 2-G-cartes des niveaux 2 et 3. Plus précisément, les coordonnées des sommets de l'objet euclidien calculé sont associées aux sommets topologiques de la 2-G-carte du niveau 3. De plus, les couleurs des faces de l'objet euclidien, déduites des couleurs des pixels correspondants du niveau 0 via les liaisons γ sont associées aux faces topologiques de la 2-G-carte du niveau 3. Enfin, les inéquations constituant la description discrète analytique du niveau 2 sont calculées et associées aux arêtes et sommets topologiques de la 2-G-carte du niveau 2.

4.4.2 Modification et mise à jour des niveaux

Dans cette section, nous abordons le problème de la mise à jour de la structure en cas de modification d'un niveau.

Nous présentons tout d'abord des diverses opérations de modification que nous souhaitons pouvoir appliquer aux niveaux de notre structure. Puis, nous discutons de la propagation des modifications d'un niveau aux autres niveaux de la structure. Nous verrons en particulier de

quelle manière les liaisons entre les niveaux peuvent être exploitées de façon à mettre à jour localement les différents niveaux.

4.4.2.1 Manipulation des représentations

Avec notre logiciel de modélisation, nous souhaitons pouvoir créer et modifier des objets euclidiens ou discrets.

Diverses opérations géométriques et topologiques permettant la manipulation d'objets euclidiens sont déjà intégrées à notre modeleur par le biais du modeleur Moka. Parmi ces opérations, nous trouvons des opérations de création d'objets (polygones, maillages, sphères, tores, ...), des opérations topologiques de bas niveau (couture, décousure, suppression de brins) et de haut niveau (fusion, contraction, insertion de cellules), ou encore des opérations géométriques simples (rotation, translation, homothétie).

Au niveau discret, les opérations fondamentales permettant la manipulation des images est l'ajout ou la suppression d'un pixel (ce qui revient à un changement de couleur du pixel). Cette opération n'est à l'heure actuelle pas implémentée, mais il s'agit de l'une des premières opérations qu'il faudra implémenter dans le futur.

Enfin, nous souhaitons à plus long terme développer des opérations permettant la création et la modification des niveaux 1 et 2 de la structure.

4.4.2.2 Mise à jour de la structure

Lorsque l'une des quatre représentations est modifiées, il est nécessaire, afin de maintenir une cohérence entre les représentations, de mettre à jour les autres niveaux. Pour cela, deux possibilités s'offrent à nous : reconstruire la totalité des trois autres niveaux en utilisant les opérations décrites dans la section 4.4.1, ou reconstruire localement les différentes représentations.

Pour ce faire, notre idée est d'utiliser les liaisons γ . En effet, ces liaisons peuvent être exploitées à plusieurs fins. Tout d'abord, nous avons vu que les informations géométriques sont stockées une seule fois au sein de la pyramide et ce pour éviter la redondance d'information. Ceci est possible car les liaisons γ permettent l'accès aux diverses informations contenus dans les niveaux, et ce à partir de n'importe quel niveau. De plus, ces liaisons nous permettent de connaître la "provenance" d'un segment topologique à un niveau donné. Ainsi, si un niveau de la pyramide est modifié, il est possible de connaître les brins des autres niveaux concernés par la modification. Les liaisons seront donc utilisées dans le cadre des mises à jour de la structure.

Il y a cependant des situations pour lesquelles une mise à jour locale ne sera pas possible, comme par exemple suite à la translation de l'objet discret. La reconstruction locale des niveaux n'est cependant pas toujours faisable. En effet, l'application de certaines opérations à une représentation impliquent nécessairement une reconstruction complète de la structure. C'est par exemple le cas de la rotation euclidienne. De la même manière, certaines modifications d'un objet discret peuvent amener à la reconstruction totale des niveaux.

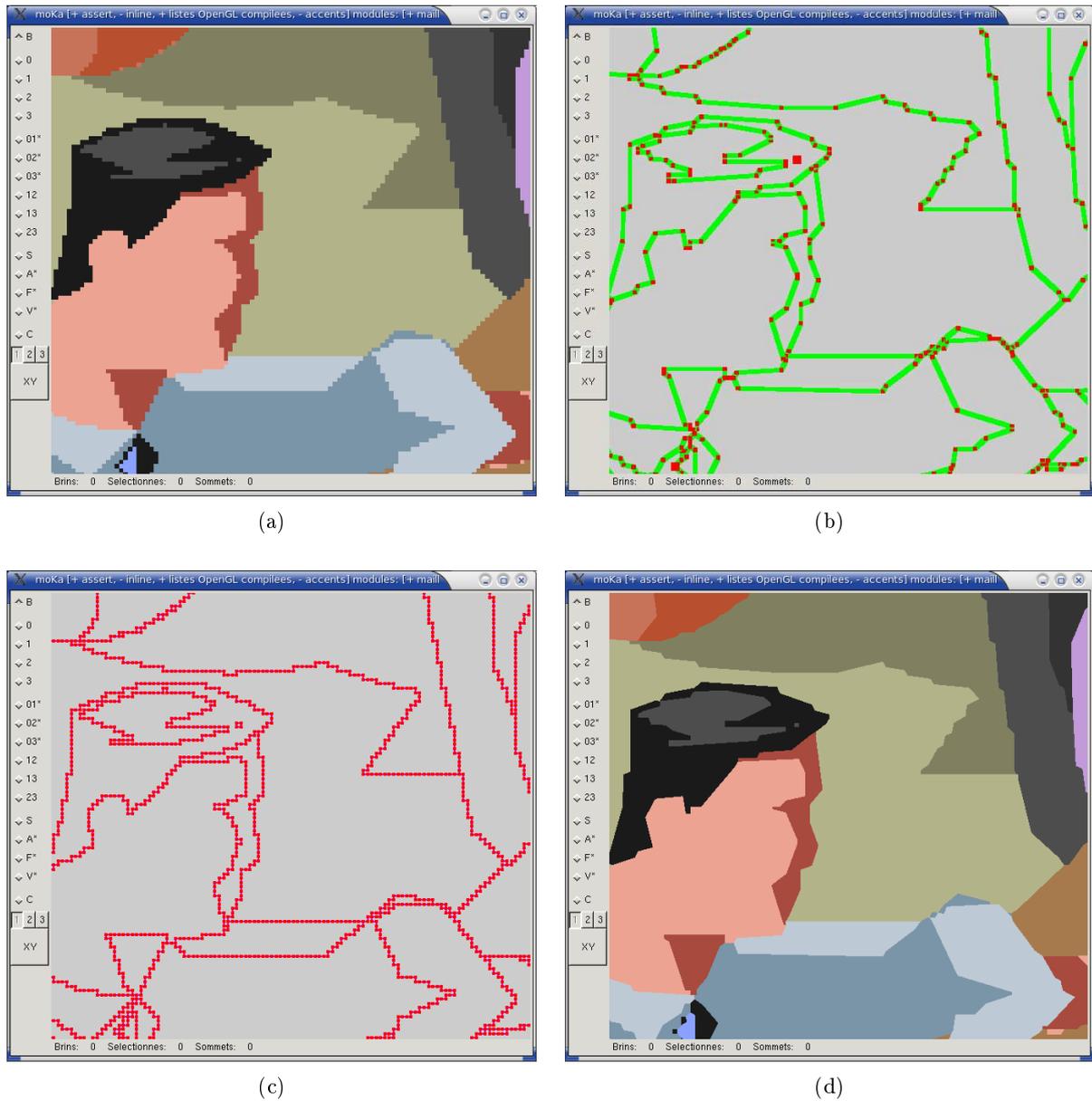
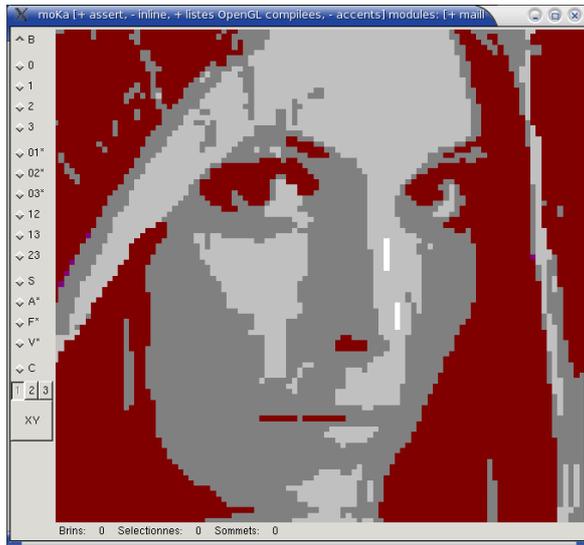


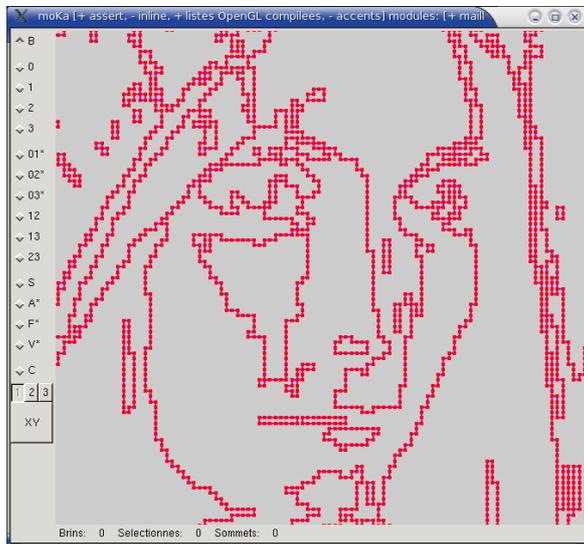
FIG. 4.29 – Une image couleur segmentée : (a) Représentation discrète, (b) Représentation analytique discrète, (c) Représentation discrète des contours, (d) Représentation euclidienne.



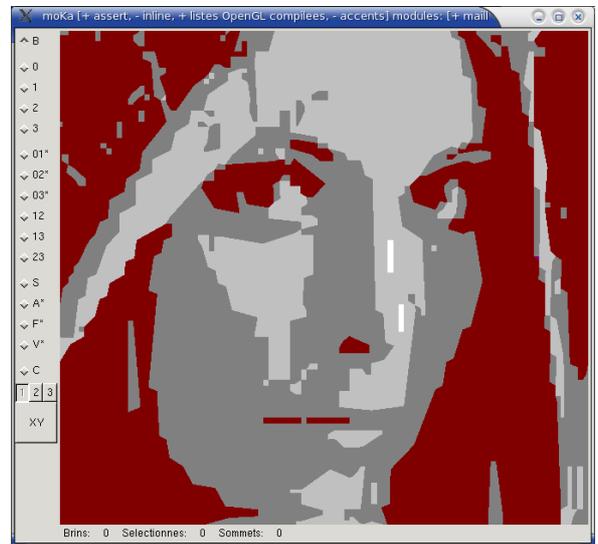
(a)



(b)



(c)



(d)

FIG. 4.30 – Une image segmentée en niveaux de gris

4.5 Conclusion et perspectives

Dans ce chapitre, nous avons tout d'abord rapidement présenté le modèle topologique des cartes généralisées à l'aide duquel les différents objets manipulés dans notre modèleur sont représentés, ce qui nous permet de disposer d'informations topologiques.

Nous avons ensuite décrit la structure hiérarchique que nous avons conçue afin de pouvoir stocker et manipuler les différentes représentations de nos objets. Cette structure permet notamment d'établir un ensemble de liaisons entre les différentes représentations de nos objets, tant au niveau topologique que géométrique.

Enfin, nous avons décrit les opérations fondamentale de construction de la structure, tout d'abord à partir d'un objet euclidien, puis à partir d'une image. Nous avons ensuite abordé le problème de la mise à jour des différents niveaux suite à une modification de l'un d'entre eux en donnant quelques idées sur le problème posé et sur les solutions envisageables.

Le développement de la partie SpaMod du logiciel de modélisation, à savoir la mise en œuvre de la structure hiérarchique et des opérations de construction, ainsi que l'intégration de ces travaux au sein du logiciel de modélisation d'objets euclidiens Moka, ont été entièrement effectuées durant cette thèse. Quelques captures d'écrans sont montrées sur les figures 4.29, 4.30 et 4.31.

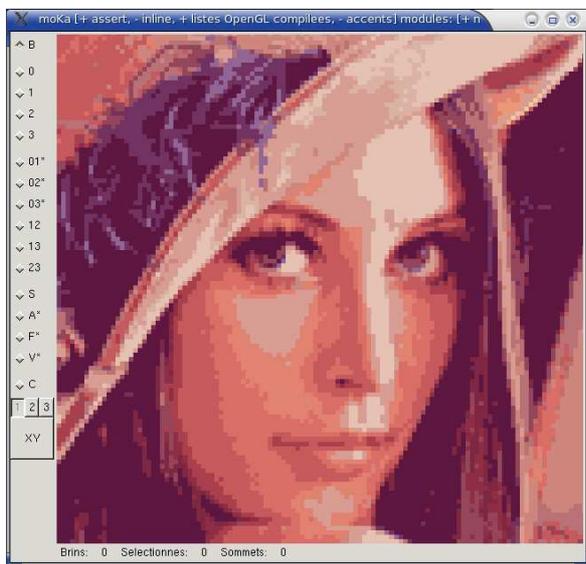
Ce logiciel de modélisation est actuellement toujours en cours de développement, et de nombreuses fonctionnalités peuvent encore lui être associées, tant pour le traitement et la manipulation d'images en dimension 2 que pour le traitement de volumes en dimension 3.

Tout d'abord, en dimension 2, les opérations de mises à jour des différentes représentations sont encore en cours d'étude.

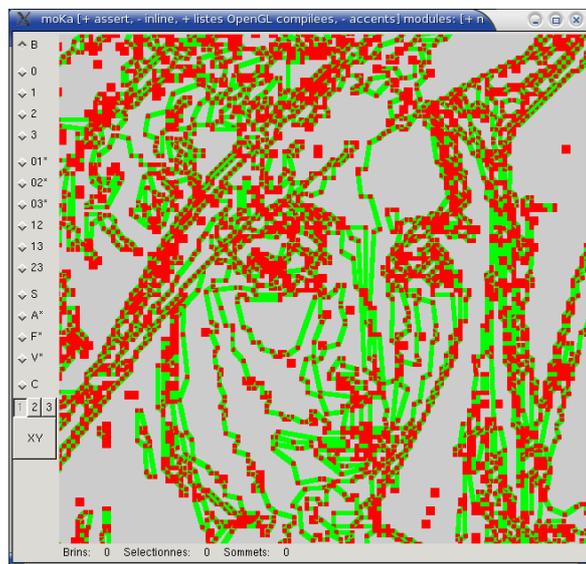
De même, certaines opérations annexes peuvent éventuellement être ajoutées. Quelques opérations telles que des opérations l'homothétie discrète [LSA06], ainsi qu'un algorithme de segmentation par ligne de partage des eaux ont déjà été intégrées.

Enfin, l'extension à la dimension 3 de ce logiciel est elle aussi en cours développement. La structure ayant été conçue de manière générique, elle est utilisable en toutes dimensions.

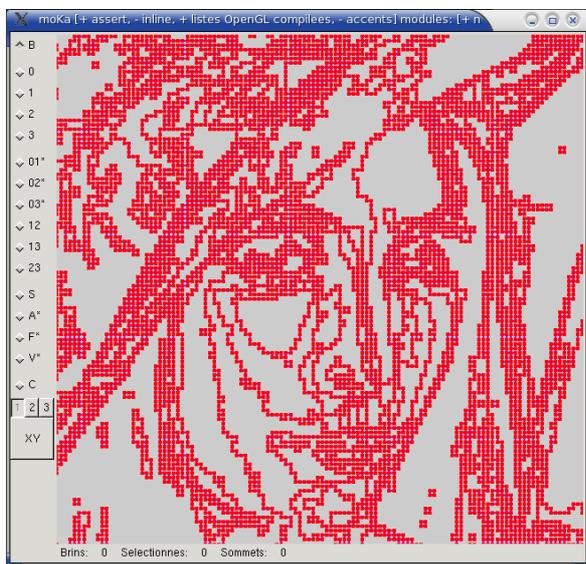
Les travaux présentés dans ce chapitre ont fait l'objet de plusieurs publications [DDGAL05, DA05, DA06b], dont un article de revue.



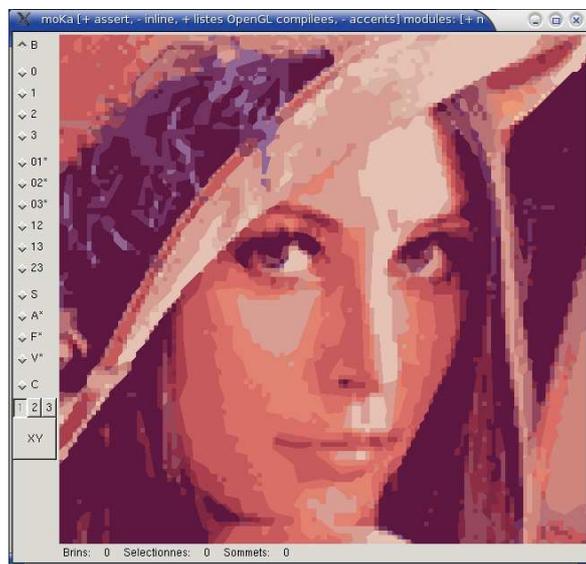
(a)



(b)



(c)



(d)

FIG. 4.31 – Une partie de l'image *Lena*

Conclusion et perspectives

Dans cette section de conclusion, nous passons en revue les différents points que nous avons abordés dans ce travail de thèse, à savoir, la reconnaissance de primitives discrètes, la reconstruction analytique d'objets discrets et enfin la conception d'un logiciel de modélisation d'objets discrets.

Pour chacun de ces points, nous récapitulons nos différentes contributions et donnons quelques perspectives à ces travaux.

Reconnaissance de primitives discrètes

Dans cette thèse, nous avons tout d'abord défini la notion de *Supercouverture universelle* d'un hyperplan euclidien comme étant l'ensemble des pavés d'un espace discret quelconque coupés par l'hyperplan (voir Chapitre 1).

Nous avons ensuite décrit, dans le chapitre 2 de ce mémoire, la *préimage généralisée* d'un ensemble de polytopes. Cette préimage est un polytope d'un espace de paramètres que nous avons défini dans le même chapitre. En particulier, chaque point de cette préimage est associé à un hyperplan coupant les polytopes donnés.

A partir de cette préimage, nous avons proposé un nouvel *algorithme de reconnaissance* d'hyperplans discrets U-Supercouverture. Cet algorithme détermine si un ensemble de pavés appartient à un hyperplan discret U-Supercouverture en calculant la préimage généralisée des pavés. Il fournit de plus l'ensemble des hyperplans solutions. Cet algorithme est incrémental, indépendant de la position et de la connexité des pavés.

Enfin, nous avons vu que cet algorithme pouvait par exemple être très aisément appliqué à la reconnaissance d'hyperplans discrets dans les espaces classiques, tels que les hyperplans Supercouverture, Standard ou Naïfs, ainsi qu'à la reconnaissance de droites dans une grille irrégulière isothétique.

Reconstruction analytique inversible d'objets discrets

Une deuxième partie de notre travail a consisté en l'étude de méthodes analytiques de reconstruction d'objets discrets. Nous voulions en particulier que les reconstructions obtenues à l'aide de ces méthodes soient inversibles pour le modèle Standard (c'est à dire que la discrétisation Standard de l'objet obtenu soit identique à l'objet discret d'origine) et topologiquement cohérentes avec l'objet discret de départ.

Nous avons proposé dans le chapitre 3 de ce mémoire deux méthodes de reconstruction répondant à nos critères en dimensions 2 et 3.

Reconstruction en dimension 2

Nous avons proposé un algorithme de reconstruction inversible de 1-courbes discrètes classiques, ouvertes ou fermées, simples ou non simples. Cette algorithme est en particulier basé sur l'algorithme de reconnaissance d'hyperplans discrets présenté dans ce mémoire et appliqué à la reconnaissance de droites Standard.

Les résultats obtenus avec cet algorithme sont visuellement de très bonne qualité et sont notamment meilleurs, en terme de nombre de segments reconstruits, que les résultats obtenus avec les deux méthodes proposées par R. BRETON dans sa thèse [Bre03].

Une première perspective à ce travail est l'amélioration de la méthode de reconstruction en recherchant par exemple des points particuliers de la courbe (tels que les points de rebroussements définis dans [Bre03]) afin de guider la reconnaissance des segments discrets pour obtenir des résultats visuellement plus proches de l'objet discret d'origine (par exemple essayer de reconstruire un "carré discret" en un carré euclidien).

Une deuxième perspective à ces travaux est l'extension de cet algorithme à la reconstruction de courbes discrètes dans un espace discret quelconque, tel que les grilles irrégulières isothétiques. De part la définition de la préimage généralisée, cette extension est quasi-immédiate.

Reconstruction en dimension 3

Nous avons présenté un nouvel algorithme de reconstruction analytique de volumes discrets 2-connexes fournissant une surface polygonale répondant à nos critères. Cet algorithme est, tout comme l'algorithme de reconstruction de courbes en dimension 2, basé sur notre algorithme de reconnaissance d'hyperplans discrets appliqué cette fois-ci à la reconnaissance de plans Standard. De plus, chaque facette de la surface polygonale générée est obtenue en deux étapes : une étape de reconnaissance contrainte (chaque facette devant être générée peut éventuellement être adjacente à une autre facette déjà générée), et une étape de polygonalisation. Cette dernière étape est effectuée en appliquant un algorithme de polygonalisation de 2-courbes discrètes planaires ouvertes ou fermées que nous avons proposé dans ce même chapitre. Cet algorithme est en particulier basé sur une méthode de reconstruction de 1-courbes irrégulières.

Les surface générées avec cette méthode, bien que répondant à nos critères, sont visuellement peu satisfaisantes. Cela est principalement du au caractère fortement contraint de l'étape de reconnaissance. De même, le nombre de facettes générées est nettement inférieur au nombre de facettes générées par la méthode des Marching Cubes, mais reste cependant proche des résultats obtenus avec des méthodes de simplifications de Marching Cubes répondant aussi à nos critères. Plusieurs améliorations peuvent cependant être apportées à notre méthode de manière à obtenir de meilleurs résultats.

Une première perspective à ces travaux est donc l'amélioration d'un ensemble d'étapes de notre algorithme. En effet, les facettes générées ont des formes très variées et présentent de multiples concavités induisant la génération de petites facettes. Une première amélioration serait d'essayer d'obtenir des facettes de forme plus régulière, et ce de manière à obtenir un nombre inférieur de petites facettes.

Une deuxième amélioration serait d'utiliser notre algorithme de reconstruction de courbes 1-connexes étendu au cas des grilles irrégulières pour reconstruire le contour des facettes de la surface. En effet, l'algorithme utilisé actuellement génère un nombre important de segments, car fortement contraint.

A plus long terme, une dernière perspective serait d'essayer d'étendre notre méthode de reconstruction de courbes discrètes en dimension 2 à la reconstruction de volumes discrets en dimension 3, c'est à dire, essayer de moins contraindre l'étape de reconnaissance de plans discrets.

Le logiciel de modélisation SpaMod

Dans le dernier chapitre de ce mémoire, nous avons décrit l'architecture de notre logiciel de modélisation géométrique à base topologique d'objets discrets en dimensions 2 et 3. Nous avons tout d'abord décrit la structure hiérarchique que nous avons conçue afin de permettre la manipulation d'objets géométriques représentés aussi bien sous forme discrète que continue. Nous avons de plus décrit les opérations de construction de cette structure et donné quelques idées concernant le manière de mettre à jour la structure dans le cas de la modification d'un des niveaux.

Ce logiciel de modélisation est actuellement encore en cours de développement, et de nombreuses fonctionnalités peuvent encore lui être associées, tant pour le traitement et la manipulation d'images en dimension 2 que pour le traitement de volumes en dimension 3.

Tout d'abord, en dimension 2, les opérations de mises à jour des différentes représentations suite à la modification de l'une d'entre elles, discrète ou euclidienne, sont encore en cours d'étude.

De même, certaines opérations annexes peuvent éventuellement être ajoutées. Certaines ont déjà été ajoutées telles que des opérations d'homothétie discrète [LSA06], ainsi qu'un algorithme de segmentation par ligne de partage des eaux.

Enfin, l'extension à la dimension 3 de ce logiciel est elle aussi en cours développement.

Bibliographie

- [AAS97] E. Andres, R. Acharya, and C. Sibata. Discrete analytical hyperplanes. *Graphical Models and Image Processing*, 59(5) :302–309, 1997.
- [ABC⁺04] C. Andújar, P. Brunet, A. Chica, J. Rossignac, I. Navazo, and A. Vinacua. Computing maximal tiles and application to impostor-based simplification. *Computer Graphics Forum*, 23(3) :401–410, 2004.
- [ABL01] E. Andres, R. Breton, and P. Lienhardt. Spamod : design of a spatial modeling tool. *Digital and Image Geometry*, (2243) :91–107, 2001.
- [ADRS⁺96] E. Andres, I. Debled-Rennesson, C. Sibata, R. Acharya, and K. Shin. Linear contour segmentation and its application to the computation of stereotactic radiosurgery dose distribution. In *SPIE Medical Imaging, section Image Processing.*, volume SPIE 2710, 1996.
- [AK85] T. A. Anderson and C. E. Kim. Representation of digital line segments and their preimages. *CVGIP : Image Understanding*, 30(3) :279–288, 1985.
- [And92] E. Andres. *Cercles Discrets et Rotations Discrètes*. Thèse de doctorat, Université Louis Pasteur, Strasbourg, France, 1992.
- [And96] E. Andres. The quasi-shear rotation. In *Discrete Geometry for Computer Imagery, 6th International Workshop.*, volume 1176 of *Lecture Notes in Computer Science*, pages 307–314, Lyon, France, 1996.
- [And00] E. Andres. *Modélisation analytique discrète d’objets géométriques*. Thèse d’habilitation, Université de Poitiers, France, 2000.
- [And03] E. Andres. Discrete linear objects in dimension n : the standard model. *Graphical Models*, 65 :92–111, 2003.
- [ANF97a] E. Andres, P. Nehlig, and J. Françon. Supercover of straight lines, planes and triangles. In *Discrete Geometry for Computer Imagery*, volume 1347 of *Lecture Notes in Computer Science*, pages 243–257, Montpellier, France, 1997.
- [ANF97b] E. Andres, P. Nehlig, and J. Françon. Tunnel-free supercover 3D polygons and polyhedra. *Computer Graphics Forum*, 16(3) :3–14, August 1997. Proceedings of Eurographics ’97. ISSN 1067-7055.
- [BCK07] V. Brimkov, D. Cœurjolly, and R. Klette. Digital Planarity - A Review. *Discrete Applied Mathematics*, 155(4) :468–495, feb 2007.
- [BD05] V. E. Brimkov and S. S. Dantchev. Complexity analysis for digital hyperplane recognition in arbitrary fixed dimension. In *Discrete Geometry for Computer Imagery*, number 3429 in *Lecture Notes in Computer Science*, pages 287–298, Poitiers, France, 2005.

- [BF94] P. Borianne and J. Françon. Reversible polyhedrization of discrete volumes. In *Discrete Geometry for Computer Imagery*, pages 157–168, Grenoble, France, 1994.
- [BM03] J. Burguet and R. Malgouyres. Strong thinning and polyhedric approximation of the surface of a voxel object. *Discrete Applied Mathematics*, 125(1) :93–114, 2003.
- [Bre65] J. Bresenham. Algorithm for computer control of a digital plotter. *IBM Systems Journal*, 4(1) :25–30, 1965.
- [Bre03] R. Breton. *Reconstruction inversible d’objets discrets 2D*. Thèse de doctorat, Université de Poitiers, France, 2003.
- [BSDA03] R. Breton, I. Sivignon, F. Dupont, and E. Andres. Towards an invertible Euclidean reconstruction of a discrete object. In *Discrete Geometry for Computer Imagery*, number 2886 in *Lecture Notes in Computer Science*, pages 246–256, Naples, Italy, 2003.
- [Buz02] L. Buzer. An incremental linear time algorithm for digital line and plane recognition using a linear incremental feasibility problem. In *Discrete Geometry for Computer Imagery*, volume 2301 of *Lecture Notes in Computer Science*, pages 372–381, Bordeaux, France, 2002.
- [Buz03] L. Buzer. A linear incremental algorithm for Naive and Standard digital lines and planes recognition. *Graphical models*, 65(1–3) :61–76, 2003.
- [CB06] D. Cœurjolly and V. Brimkov. Computational aspects of digital plane and hyperplane recognition. In *International Workshop on Combinatorial Image Analysis*, volume 4040 of *Lecture Notes in Computer Science*, pages 291–306, Berlin, Germany, 2006.
- [CDJS06] D. Cœurjolly, F. Dupont, L. Jospin, and I. Sivignon. Optimization schemes for the reversible discrete volume polyhedrization using marching cubes simplification. In *Discrete Geometry for Computer Imagery*, volume 4245 of *Lecture Notes in Computer Science*, pages 413–424, Szeged, Hungary, 2006.
- [CGS04] D. Cœurjolly, A. Guillaume, and I. Sivignon. Reversible discrete volume polyhedrization using marching cubes simplification. In *Proceedings of the SPIE*, volume 5300 of *Vision Geometry XII*, pages 1–11, 2004.
- [Che95] E. V. Chernyaev. *Marching cubes 33 : Construction of topologically correct isosurfaces*, 1995.
- [Cœu02] D. Cœurjolly. *Algorithmique et géométrie discrète pour la caractérisation des courbes et des surfaces*. Thèse de doctorat, Université Lumière Lyon 2, Lyon, France, 2002.
- [Cœu05] D. Cœurjolly. Supercover model and digital straight line recognition on irregular isothetic grids. In *Discrete Geometry for Computer Imagery*, volume 3429 of *Lecture Notes in Computer Science*, pages 311–322, Poitiers, France, 2005.
- [COK95] D. Cohen-Or and A. Kaufman. Fundamentals of surface voxelization. *Graphical Models and Image Processing*, 57(6) :453–461, 1995.
- [CSD⁺05] D. Coeurjolly, I. Sivignon, F. Dupont, F. Feschet, and J.-M. Chassery. On digital plane preimage structure. *DAMATH : Discrete Applied Mathematics and Combinatorial Operations Research and Computer Science*, 151, 2005.
- [CZ06] D. Cœurjolly and L. Zerarga. Supercover model, digital straight line recognition and curve reconstruction on the irregular isothetic grids. *Computers and Graphics*, 30(1) :46–53, 2006.

-
- [DA05] M. Dexet and E. Andres. Architecture d'un modeleur géométrique à base topologique d'objets discrets. In *Journées de l'Association Française d'Informatique Graphique*, Strasbourg, France, 2005.
- [DA06a] M. Dexet and E. Andres. A generalized preimage for the Standard and Supercover digital hyperplane recognition. In *Discrete Geometry for Computer Imagery*, volume 4245 of *Lecture Notes in Computer Science*, pages 639–650, Szeged, Hungary, 2006.
- [DA06b] M. Dexet and E. Andres. Hierarchical topological structure for the design of a discrete modeling tool. In *WSCG Full Papers Proceedings*, pages 1–8, Plzen, Czech Republic, 2006.
- [DA06c] M. Dexet and E. Andres. Linear discrete line recognition and reconstruction based on a generalized preimage. In *International Workshop on Combinatorial Image Analysis*, volume 4040 of *Lecture Notes in Computer Science*, pages 174–188, Berlin, Germany, 2006.
- [Dam01] G. Damiand. *Définition et étude d'un modèle topologique minimal de représentation d'images 2d et 3d*. Thèse de doctorat, Université Montpellier II, Montpellier, France, 2001.
- [DCA06] M. Dexet, D. Coeurjolly, and E. Andres. Invertible polygonalization of 3d planar digital curves and application to volume data reconstruction. In *International Symposium on Visual Computing*, Lecture Notes on Computer Science, Lake Tahoe, Nevada (USA), 2006.
- [DDGAL05] G. Damiand, M. Dexet-Guiard, E. Andres, and P. Lienhardt. Removal and contraction operations to define combinatorial pyramids : application to the design of a spatial modeler. *Image and Vision Computing*, 23(2) :259–269, 2005.
- [DDSD03] X. Décoret, F. Durand, F. Sillion, and J. Dorsey. Billboard clouds for extreme model simplification. In *Proceedings of the ACM Siggraph*. ACM Press, 2003.
- [DH72] R. O. Duda and P. E. Hart. Use of the Hough transformation to detect lines and curves in pictures. *Commun. ACM*, 15(1) :11–15, 1972.
- [DL03] G. Damiand and P. Lienhardt. Removal and contraction for N-dimensional generalized maps. In *Discrete Geometry for Computer Imagery*, volume 2886 of *Lecture Notes in Computer Science*, pages 408–419, Naples, Italy, november 2003.
- [DR95] I. Debled-Renesson. *Étude et reconnaissance des droites et plans discrets*. Thèse de doctorat, Université Louis Pasteur, Strasbourg, France, 1995.
- [DRR95] I. Debled-Renesson and J.-P. Reveillès. A linear algorithm for segmentation of digital curves. *International Journal of Pattern Recognition and Artificial Intelligence*, 9(6) :635–662, 1995.
- [DS84] L. Dorst and A. W. M. Smeulders. Discrete representation of straight lines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(4) :450–463, 1984.
- [FB74] R. A. Finkel and J. L. Bentley. Quad trees : A data structure for retrieval on composite keys. *Acta Informatica*, 4(1) :1, 1974.
- [FJT06] C. Fiorio, D. Jamet, and J.-L. Toutant. Discrete circles : an arithmetical approach with non-constant thickness. In *SPIE Electronic Imaging - Vision Geometry XIV*, San Jose, United States, 2006.
- [FP99] J. Françon and L. Papier. Polyhedrization of the boundary of a voxel object. In *Discrete Geometry for Computer Imagery*, volume 1568 of *Lecture Notes in Computer Science*, pages 425–434, Lyon, France, 1999.
-

- [Fra04] D. Fradin. *Modélisation et simulation d'éclairage à base topologique : application aux environnements architecturaux complexes*. Thèse de doctorat, Université de Poitiers, Poitiers, France, 2004.
- [FST96] J. Françon, J.-M. Schramm, and M. Tajine. Recognizing arithmetic straight lines and planes. In *Discrete Geometry for Computer Imagery*, volume 1176 of *Lecture Notes in Computer Science*, pages 141–150, Lyon, France, 1996.
- [FT06] C. Fiorio and J.-L. Toutant. Discrete arithmetic hyperspheres and separatingness. In *Discrete Geometry for Computer Imagery*, volume 4245 of *Lecture Notes in Computer Science*, pages 425–436, Szeged, Hungary, 2006.
- [FT07] C. Fiorio and J.-L. Toutant. Arithmetic characterization of polynomial-based discrete curves. In *SPIE Electronic Imaging - Vision Geometry XV*, San Jose, United States, 2007.
- [GDRZ05] Y. Gerard, I. Debled-Rennesson, and P. Zimmermann. An elementary digital plane recognition algorithm. *DAMATH : Discrete Applied Mathematics and Combinatorial Operations Research and Computer Science*, 151, 2005.
- [GS06] C. Grasset-Simon. *Définition et étude des pyramides généralisées nD : application pour la segmentation multi-échelle d'images 3D*. Thèse de doctorat, Université de Poitiers, Poitiers, France, 2006.
- [Gui06] N. Guiard. *Construction de modèles géologiques 3D par co-raffinement de surfaces*. Thèse de doctorat, Ecole Nationale Supérieure des Mines de Paris, Paris, France, 2006.
- [HHVW96] T. He, L. Hong, A. Varshney, and S. W. Wang. Controlled topology simplification. *IEEE Transactions on Visualization and Computer Graphics*, 2(2), 1996. ISSN 1077-2626.
- [Hou62] P. V. C. Hough. Methods and means to recognize complex patterns. *U.S. Patent 3 069 654*, 1962.
- [HWC⁺05] C.-C. Ho, F.-C. Wu, B.-Y. Chen, Y.-Y. Chuang, and M. Ouhyoung. Cubical marching squares : Adaptive feature preserving surface extraction from volume data. *Computer Graphics Forum*, 24(3) :537–545, 2005. Special issue : Proceedings of EUROGRAPHICS 2005.
- [KIE96] Y. Kenmochi, A. Imiya, and N. F. Ezquerra. Polyhedra generation from lattice points. In *Discrete Geometry for Computer Imagery*, volume 1176 of *Lecture Notes in Computer Science*, pages 127–138, Lyon, France, 1996.
- [Kim83] C. E. Kim. Three-dimensional line segments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(2) :231–234, 1983.
- [Kim84] C. E. Kim. Three-dimensional digital planes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(5) :639–645, 1984.
- [KKM90a] E. Khalimsky, R. Kopperman, and P.R. Meyer. Boundaries in digital planes. *Journal of Applied Mathematics and Stochastic Analysis*, 3 :27–55, 1990.
- [KKM90b] E. Khalimsky, R. Kopperman, and P.R. Meyer. Computer graphics and connected topologies as finite ordered sets. *Topology and its Applications*, 36 :1–17, 1990.
- [KL02] C.-S. Kim and S. U. Lee. Compact encoding of 3-D voxel surfaces based on pattern code representation. *IEEE Transactions on Image Processing*, 11(8) :932–943, 2002.

-
- [Kov89] V. A. Kovalevsky. Finite topology as applied to image analysis. *Computer Vision, Graphics and Image Processing*, 46 :141–161, 1989.
- [KR82] C. E. Kim and A. Rosenfeld. Convex digital solids. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 4(6) :612–618, November 1982.
- [KR04a] R. Klette and A. Rosenfeld. *Digital Geometry : Geometric Methods for Digital Picture Analysis*. Computer Graphics and Geometric Modeling. Morgan Kaufmann, 2004.
- [KR04b] R. Klette and A. Rosenfeld. Digital straightness – a review. *Discrete Applied Mathematics*, 139(1–3) :197–230, 2004.
- [KS91] C. E. Kim and I. Stojmenović. On the recognition of digital planes in three-dimensional space. *Pattern Recognition Letters*, 12(11) :665–669, 1991.
- [KS01] R. Klette and H. J. Sun. Digital planar segment based polyhedrization for surface area estimation. In *International Workshop on Visual Form*, volume 2059 of *Lecture Notes in Computer Science*, pages 356–366, 2001.
- [Lac96] J.-O. Lachaud. Topologically defined iso-surfaces. In *Discrete Geometry for Computer Imagery*, volume 1176 of *Lecture Notes in Computer Science*, pages 245–256, Lyon, France, 1996.
- [LC87] W. E. Lorensen and H. E. Cline. Marching cubes : a high resolution 3D surface construction algorithm. In *SIGGRAPH*, volume 21 of *Computer Graphics (ACM)*, pages 163–169, Anaheim, USA, 1987.
- [LI03] T. K. Linh and A. Imiya. Nonlinear optimization for polygonalisation. In *Discrete Geometry for Computer Imagery*, volume 2886 of *Lecture Notes in Computer Science*, pages 444–453, Naples, Italy, 2003.
- [Lie89] P. Lienhardt. Subdivision of N-dimensional spaces and N-dimensional generalized maps. In *ACM Symposium on Computational Geometry*, pages 228–236, Saarbrücken, Germany, 1989.
- [Lie91] P. Lienhardt. Topological models for boundary representation : a comparison with N-dimensional generalized maps. *Computer Aided Design*, 23(1) :59–82, 1991.
- [Lie94] P. Lienhardt. N-dimensional generalized combinatorial maps and cellular quasi-manifolds. *International Journal of Computational Geometry and Applications*, 4(3) :275–324, 1994.
- [Lin05] J. Lindblad. Surface area estimation of digitized 3D objects using weighted local configurations. *Image and Vision Computing*, 23(2) :111–122, 2005.
- [LL01] T. Y. Lee and C. H. Lin. Growing-cube isosurface extraction algorithm for medical volume data. *Comput Med Imaging Graph.*, 25(5) :405–415, 2001.
- [LLVT03] T. Lewiner, H. Lopes, A. W. Vieira, and G. Tavares. Efficient implementation of marching cubes’cases with topological guarantees. *Journal of Graphics Tools : JGT*, 8(2) :1–15, 2003.
- [LM00] J.-O. Lachaud and A. Montanvert. Continuous analogs of digital boundaries : A topological approach to iso-surfaces. *Graphical Models and Image Processing*, 62 :129–164, 2000.
- [LN02] J. Lindblad and I. Nyström. Surface area estimation of digitized 3D objects using local computations. In *Discrete Geometry for Computer Imagery*, volume 2301 of *Lecture Notes in Computer Science*, pages 267–278, Bordeaux, France, 2002.
-

- [LSA06] G. Largeteau-Skapin and E. Andres. Two discrete-continuous operations based on the scaling transform. In *Discrete Geometry for Computer Imagery*, volume 4245 of *Lecture Notes in Computer Science*, pages 41–52, Szeged, Hungary, 2006.
- [Mai85] H. Maitre. Un panorama de la transformation de Hough – a review on Hough transform. *Traitement du Signal*, 2(4) :305–317, 1985.
- [McI85] M. D. McIlroy. A note on discrete representation of lines. *AT&T Technical Journal*, 64(2) :481–490, 1985.
- [MSS94] C. Montani, R. Scateni, and R. Scopigno. Discretized marching cubes. In *Proceedings of the Conference on Visualization*, pages 281–287, Los Alamitos, CA, USA, 1994.
- [MSS00] C. Montani, R. Scateni, and R. Scopigno. Decreasing isosurface complexity via discrete fitting. *Computer Aided Geometric Design*, 17(3) :207–232, 2000.
- [Nie03] G. M. Nielson. On marching cubes. *IEEE Transactions on Visualization and Computer Graphics*, 9(3) :283–297, 2003.
- [Nie04] G. M. Nielson. Dual marching cubes. In *IEEE Visualization*, pages 489–496. IEEE Computer Society, 2004.
- [Pap99] L. Papier. *Polyédrisation et visualisation d’objets discrets tridimensionnels*. Thèse de doctorat, Université Louis Pasteur, Strasbourg, France, 1999.
- [PZvBG00] H. Pfister, M. Zwicker, J. van Baar, and M. H. Gross. Surfels : surface elements as rendering primitives. In *SIGGRAPH*, pages 335–342, 2000.
- [Rev91] J.-P. Reveillès. *Géométrie discrète, calcul en nombres entiers et algorithmique*. Thèse d’état, Université Louis Pasteur, Strasbourg, France, 1991.
- [Rev95] J.-P. Reveillès. Combinatorial pieces in digital lines and planes. *SPIE Vision Geometry IV*, 2573, 1995.
- [SB01] I.-M. Sintorn and G. Borgefors. Weighted distance transforms in rectangular grids. In *ICIAP*, pages 322–326. IEEE Computer Society, 2001.
- [SB02] I.-M. Sintorn and G. Borgefors. Weighted distance transforms for images using elongated voxel grids. In *Discrete Geometry for Computer Imagery*, volume 2301 of *Lecture Notes in Computer Science*, pages 244–254, Bordeaux, France, 2002.
- [SBDA05] I. Sivignon, R. Breton, F. Dupond, and E. Andres. Discrete analytical curve reconstruction without patches. *Image and Vision Computing*, 23(2) :191–202, 2005.
- [SC06] I. Sivignon and D. Coeurjolly. Minimal decomposition of a digital surface into digital plane segments is NP-hard. In *Discrete Geometry for Computer Imagery*, volume 4245 of *Lecture Notes in Computer Science*, pages 674–685, Szeged, Hungary, 2006.
- [SDC03] I. Sivignon, F. Dupont, and J.-M. Chassery. Decomposition of a three-dimensional discrete object surface into discrete plane pieces. *Algorithmica*, 38(1) :25–43, 2003.
- [SDC05] I. Sivignon, F. Dupont, and J.-M. Chassery. Reversible polygonalization of a 3D planar discrete curve : Application on discrete surfaces. In *Discrete Geometry for Computer Imagery*, volume 3429 of *Lecture Notes in Computer Science*, pages 347–358, Poitiers, France, 2005.
- [SFYC96] R. Shekhar, E. Fayyadm, R. Yagel, and J. Cornhill. Octree-based decimation of marching cubes surfaces. *Proceedings of IEEE Visualization’96*, pages 335–342, 1996.

- [Siv04] I. Sivignon. *De la caractérisation des primitives à la reconstruction polyédrique de surfaces en géométrie discrète*. Thèse de doctorat, Institut National Polytechnique de Grenoble, Grenoble, France, 2004.
- [ST91] I. Stojmenović and R. Tošić. Digitization schemes and the recognition of digital straight lines, hyperplanes, and flats in arbitrary dimensions. In *Vision Geometry*, volume 119 of *Contemporary Mathematics Series*, pages 197–212. American Mathematical Society, 1991.
- [TWR99] M. Tajine, D. Wagner, and C. Ronse. Hausdorff discretization and its comparison to other discretization schemes. In *Discrete Geometry for Computer Imagery*, volume 1568 of *Lecture Notes in Computer Science*, pages 399–412, Noisy-le-Grand, France, 1999.
- [VC99] J. Vittone and J.-M. Chassery. (n, m) -cubes and farey nets for Naive planes understanding. In *Discrete Geometry for Computer Imagery*, volume 1568 of *Lecture Notes in Computer Science*, pages 76–90, Noisy-le-Grand, France, 1999.
- [VC00] J. Vittone and J.-M. Chassery. Recognition of digital Naive planes and polyhedrization. In *Discrete Geometry for Computer Imagery*, volume 1953 of *Lecture Notes in Computer Science*, pages 296–307, Uppsala, Sweden, 2000.
- [Vee99] P. Veelaert. Geometric constructions in the digital plane. *Journal of Mathematical Imaging and Vision*, 11(2) :99–118, October 1999.
- [Vee05] P. Veelaert. Uncertain geometry in computer vision. In *Discrete Geometry for Computer Imagery*, volume 3429 of *Lecture Notes in Computer Science*, pages 359–370, Poitiers, France, 2005.
- [Via96] A. Vialard. *Chemins euclidiens : un modèle de représentation des contours discrets*. Thèse de doctorat, Université Bordeaux 1, Bordeaux, France, 1996.
- [Vit99] J. Vittone. *Caractérisation et reconnaissance de droites et de plans en géométrie discrète*. Thèse de doctorat, Université Joseph Fourier, Grenoble, France, 1999.
- [YK02] L. Yu and R. Klette. An approximative calculation of relative convex hulls for surface area estimation of 3d digital objects. In *International Conference on Pattern Recognition*, volume 1, pages 131–134, 2002.

Table des figures

1		1
2	Quelques propriétés du monde discret : (a) Deux droites discrètes séquentes sans intersection, (b) Deux droites discrètes parallèles avec intersection, (c) Deux droites discrètes différentes contenant les deux mêmes pixels.	3
1.1	De gauche à droite : projections symétriques en dimension 2 des hypercubes de dimensions respectives 2 (un carré), 3 (un cube), 4 (un <i>tesseract</i>) et 5. Notons que sur certains exemples, telle que la deuxième figure, plusieurs sommets de l'objet coïncident.	6
1.2	Espaces discrets classiques en dimensions 2 et 3 : (a) et (b) Deux pavages possibles de l'espace par des pixels ; chaque point discret est matérialisé par un point noir. (c) Pavage de l'espace par des voxels.	7
1.3	Exemples de pavages réguliers et irréguliers en dimension 2 : (a) Pavage régulier par des hexagones. (b) Pavage régulier par des pavés concaves. (c) Pavage irrégulier. (d) Grille irrégulière isothétique.	8
1.4	Exemples de voisinages en dimension 2 : (a) Un pixel gris et ses quatre 1-voisins (en noir). (b) Le même pixel avec ses huit 0-voisins.	9
1.5	Exemples de voisinages en dimension 3 : (a) Un voxel (au centre) et ses six 2-voisins. (b) Le même voxel et ses dix-huit 1-voisins. (c) Le même voxel et ses vingt-six 0-voisins.	9
1.6	Exemples de k -courbes : (a) Une 1-courbe en dimension 2. (b) Une 0-courbe en dimension 2. (c) Une 2-courbe en dimension 3.	10
1.7	Exemples de 1-courbes irrégulières : (a) 1-courbe fermée dans un espace irrégulier. (b) 1-courbe ouverte dans une grille irrégulière isothétique.	11
1.8	Segments de droites discrètes de paramètres $C = (1, -3)$ et d'épaisseur variable : (a) Une droite mince. (b) Une droite Naïve. (c) Une droite Standard. (d) Une droite épaisse.	12
1.9	Droite Naïve de paramètres $(1, 3, -7)$. Le pixel coloré en noir est le pixel de centre $(0, 0)$	13
1.10	Exemple de discrétisation Naïve et Naïve fermée d'une droite euclidienne en dimension 2. (a) Pavage de l'espace par des losanges réguliers (boules B_{d_1} de rayon $\frac{1}{2}$). (b) Droite Naïve et Naïve fermée obtenue.	14
1.11	Droite Supercouverture de paramètres $(0, 3, -7)$. Le pixel coloré en noir est le pixel de centre $(0, 0)$. Nous pouvons remarquer la présence d'une 2-bulle (pixels gris foncés).	14
1.12	Discrétisation Supercouverture d'une droite euclidienne dans une grille irrégulière isothétique (figure inspirée de [Cœu05]).	15

1.13	Hyperplans Standard en dimensions 2 et 3 : (a) Droite Standard de paramètres $(0, 3, -7)$. Le pixel coloré en noir est le pixel de centre $(0, 0)$. Notons que le pixel blanc n'appartient pas à la droite Standard. (b) Plan Standard de paramètres $(0, 3, -1, 2)$	16
1.14	Exemple de droite Standard en dimension 3 et ses trois projections orthogonales.	17
1.15	Exemple de segment Standard et sa description analytique (matérialisée par le polygone en traits continus gras). Les pixels en gris correspondent à la discrétisation Standard de S (trait pointillé).	17
2.1	Extrusions positive et négative d'un point p (demi-droites) et leur objet dual : un demi-espace, (a) Extrusion positive de p , (b) Extrusion négative de p	25
2.2	Extrusions positive et négative d'un polytope de dimension 2 : (a) Un 2-polytope P_c , (b) Extrusion positive de P_c , (c) Extrusion négative de P_c	26
2.3	Dual d'un polygone convexe P_c : (a) Dual de l'extrusion positive de P_c , (b) Dual de l'extrusion négative de P_c , (c) Dual de P_c	29
2.4	Exemples de duaux d'hypervoxels en dimensions 2 et 3. La numérotation illustre la correspondance entre les sommets d'un pixel (resp. voxel) et les droites (resp. plans) constituant le bord de son dual : (a) Dual du pixel de centre $(0, 0)$, (b) Dual du voxel de centre $(0, 0, 0)$	31
2.5	Quelques exemples de duaux de pixels. Le pixel coloré en gris clair, et commun à toutes les illustrations, est le pixel de centre $(1, -1)$	32
2.6	Exemples de duaux de voxels.	34
2.7	Exemples de duaux de voxels.	35
2.8	Exemples de duaux de voxels.	36
2.9	Exemple de préimage généralisée en dimension 3 : (a) Un ensemble \mathcal{V} de dix-sept voxels, (b) Préimage généralisée correspondante $\mathbb{G}_P(\mathcal{V})$. Son barycentre B est matérialisé par un point noir, (c) et (d) Résultat de l'intersection entre \mathcal{V} et le plan dual de B . Nous pouvons remarquer que le plan correspondant à $Dual(B)$ coupe bien l'ensemble des voxels de \mathcal{V}	37
2.10	Préimage généralisée de pixels : deux polygones infinis.	38
2.11	Préimage généralisée de pixels : un polygone infini.	38
2.12	Préimage généralisée de pixels : deux polygones finis.	39
2.13	Exemples de préimages généralisées en dimension 3 : (a) Préimage généralisée de trois voxels alignés sur l'axe X_3 , (b) Préimage généralisée de trois voxels alignés sur l'axe X_2 , (c) Préimage généralisée de quatre voxels formant un carré sur le plan (O, X_1, X_2) , (d) Préimage généralisée de quatre voxels formant un carré sur le plan (O, X_2, X_3)	40
2.14	Exemple de calcul d'une préimage généralisée de pixels.	42
3.1	Exemple d'objet reconstruit à partir d'images : (a) <i>Stanford Bunny</i> , (b) Reconstruction obtenue à partir de plusieurs clichés de l'objet original.	44
3.2	Reconstruction d'une 1-courbe discrète : (a) Résultat de l'étape de reconnaissance de segments discrets. Les pixels gris foncés appartiennent à deux segments discrets différents, (b) Une reconstruction possible (chaque segment discret reconnu est remplacé par un segment euclidien), (c) Exemple de reconstruction inversible pour le modèle Standard.	45
3.3	Exemples de reconstructions inversibles possibles de 1-courbes discrètes simples et non simples : (a) Courbe simple fermée, (b) Courbe non simple.	46

3.4	Exemple de reconstruction avec joints : (a) Ensemble des droites solutions (en gris foncé sur la figure) pour le segment discret reconnu, (b) Choix d'une droite solution, (c) Résultat du processus de reconnaissance, (d) Courbe polygonale obtenue après l'ajout de joints.	48
3.5	Exemple de reconstruction sans joints : (a) Ensemble des droites solutions contenant le point fixé (en gris foncé sur la figure), (b) Choix d'une droite solution et d'un nouveau point fixé, (c) Résultat.	50
3.6	Principe de notre méthode de reconstruction. (a) Reconnaissance du premier segment. Le pixel de départ est en gris foncé. La flèche noire indique la direction de reconnaissance. (b) Reconnaissance des segments suivants. (c) Étape de polygonalisation. La flèche noire indique la direction de polygonalisation.	51
3.7	Illustration du processus de reconnaissance. (a) Contrainte de la reconnaissance du premier segment (I sur la figure). Pixel de départ (P). (b) Contrainte de la reconnaissance du deuxième segment. (c) Ensemble de droites euclidiennes résultant du processus de reconnaissance de courbe discrète.	52
3.8	Résultat du processus de polygonalisation (en pointillés sur la figure). Le sens de reconstruction est indiqué par la flèche noire.	56
3.9	Résultat du processus de reconstruction : (a) Image originale, (b) Objet obtenu, (c) La courbe polygonale obtenue coupe bien tous les pixels de la courbe discrète.	56
3.10	Exemple de courbe discrète reconstruite avec la méthode étendue : (a) Image d'origine : la frontière entre les régions noire et blanche est constituée de 178 points discrets, (b) Image résultante : la frontière entre les régions noire et blanche est constituée de 38 segments.	58
3.11	Exemple de courbe discrète reconstruite avec la méthode étendue : (a) Image d'origine : la frontière entre les régions noire et blanche est constituée de 212 points discrets, (b) Image résultante : la frontière entre les régions noire et blanche est constituée de 38 segments.	60
3.12	Exemple de courbe discrète reconstruite avec la méthode étendue : (a) Image d'origine : la frontière entre les régions noire et blanche est constituée de 302 points discrets, (b) Image résultante : la frontière entre les régions noire et blanche est constituée de 64 segments.	61
3.13	Exemple de courbe discrète reconstruite avec la méthode étendue : (a) Image d'origine : la frontière entre les régions noire et blanche est constituée de 305 points discrets, (b) Image résultante : la frontière entre les régions noire et blanche est constituée de 39 segments.	62
3.14	Données traitées : (a) Objet discret d'origine V . (b) et (c) Bord B_V de V composé de points discrets. (d), (e) et (f) Objets discrets équivalents à B_V . L'objet euclidien obtenu doit couper chaque voxel de (e).	64
3.15	Illustration de la méthode de reconstruction : (a) Surfels reconnus, (b) Extraction du bord de la courbe discrète (en gras), (c) Une courbe discrète 2-connecte planaire C et le plan euclidien choisi P , (d) Résultat de l'intersection entre P et C	65
3.16	Surfel de départ et ordre de reconnaissance. (a) Surfel de départ choisi pour la première reconnaissance de plan. (b) Ordre d'ajout des surfels pendant le processus de reconnaissance.	66
3.17	Exemple de polygonalisation de courbe : (a) Une courbe discrète, (b) Un plan coupant tous les voxels, (c) Intersection entre le plan et la courbe, (d) Un résultat possible.	69

3.18	Illustration de l'algorithme d'intersection : (a) Opération d'intersection entre le plan et les arêtes, (b) Création de nouvelles arêtes, (c) Création de nouvelles faces.	71
3.19	Illustration de l'algorithme de polygonalisation de 1-courbe irrégulière : (a) Premier cône calculé, (b) Réduction du cône de visibilité en fonction du pavé suivant, (c) Exemple de cône de visibilité vide, (d) Choix d'un nouveau point et calcul d'un nouveau cône.	72
3.20	Traitement particulier. (a) Un surfel contenant trois voxels contraints. (b) Ajout de deux facettes adjacentes.	72
3.21	Exemples d'objets obtenus avec notre méthode de reconstruction.	73
3.22	Exemple de facettes reconstruites concaves (en rouge sur les figures (a) et (b)). Sur la figure (b), nous pouvons voir le cas d'une facette contenant un trou.	75
3.23	Exemple d'objet discret reconstruit à l'aide de quatre méthodes de propagation différentes : (a) Méthode 1 : 4358 faces, (b) Méthode 2 : 4494 faces, (c) Méthode 3 : 4603 faces, (d) Méthode 4 : 4850 faces.	76
3.24	Exemple d'objet discret reconstruit à l'aide de quatre méthodes de propagation différentes : (a) Méthode 1 : 7661 faces, (b) Méthode 2 : 7646 faces, (c) Méthode 3 : 7950 faces, (d) Méthode 4 : 9176 faces.	77
3.25	Exemple de reconstruction obtenue avec notre méthode : la surface de l'objet discret est constituée de 4824 points discrets, et l'objet reconstruit contient 3472 faces. La méthode de propagation utilisée est la méthode 2.	79
3.26	Exemple de reconstruction obtenue avec notre méthode : la surface de l'objet discret est constituée de 2648 points discrets, et l'objet reconstruit contient 336 faces. La méthode de propagation utilisée est la méthode 1.	81
3.27	Exemple de reconstruction obtenue avec notre méthode : la surface de l'objet discret est constituée de 86516 points discrets, et l'objet reconstruit contient 63247 faces. La méthode de propagation utilisée est la méthode 1.	83
3.28	Exemple de reconstruction obtenue avec notre méthode : la surface de l'objet discret est constituée de 110940 points discrets, et l'objet reconstruit contient 87182 faces. La méthode de propagation utilisée est la méthode 2.	86
4.1	Exemple de décomposition topologique d'un objet en 2-G-carte : (a) Un objet euclidien en dimension 2, (b) Sa décomposition en faces topologiques, (c) Décomposition des faces en arêtes topologiques, (d) Décomposition des arêtes en brins.	90
4.2	Exemple de 2-G-carte avec bord (a), et sans bord (b).	91
4.3	Exemples de i -cellules d'un objet topologique (en gras sur la figure) : (a) Sommets topologiques, (b) Arêtes topologiques, (c) Faces topologiques.	91
4.4	Exemple de 3-couture possible entre deux volumes topologiques. Les deux faces topologiques cousues sont isomorphes.	92
4.5	Représentation simplifiée d'une 2-G-carte.	93
4.6	Illustration de la structure hiérarchique en dimension 2. Chaque niveau correspond à une représentation spécifique, discrète ou continue, d'un même objet géométrique. De plus, des liaisons bidirectionnelles sont instaurées entre deux niveaux consécutifs dans la structure.	94
4.7	Le niveau euclidien. (a) Visualisation : objet composé de deux faces colorées adjacentes. (b) 2-G-carte : une couleur et des coordonnées à coefficients réels sont respectivement associées aux faces et sommets topologiques.	95

4.8	Le niveau discret analytique. (a) Visualisation : un objet euclidien (en pointillés) et la représentation analytique discrète Standard de ses contours (zones grisées). (b) 2-G-carte : quatre (resp. deux) inéquations sont associées à chaque sommet (resp. arête) topologique.	96
4.9	Les contours discrets : (a) Visualisation : chaque point discret est représenté par un pointel, et deux pointels successifs sont reliés par un lignel. (b) 2-G-carte : aucun plongement n'est associé.	96
4.10	Le niveau discret : (a) Visualisation : une image en niveaux de gris. (b) 2-G-carte : un point de coordonnées entières et une couleur sont respectivement associés à chaque sommet et face topologiques.	97
4.11	Exemple de 2-G-carte simulée : (a) 2-G-carte. (b) Chaque case de la matrice (traits pointillés) simule quatre brins (en gras sur la figure) d'un sommet topologique.	98
4.12	Illustration de la première contrainte de cohérence : (a) Niveau 0. (b) Niveau 1.	98
4.13	Illustration de la deuxième contrainte de cohérence : (a) Niveau 3. (b) Niveau 2.	99
4.14	Illustration de la troisième contrainte de cohérence : tous les points discrets du niveau 1 sont contenus dans l'objet décrit analytiquement au niveau 2.	99
4.15	Exemple d'objet euclidien : (a) Représentation géométrique. (b) Représentation topologique.	101
4.16	(a) Objet euclidien. (b) Représentation discrète analytique correspondante.	101
4.17	2-G-cartes et liaisons entre le niveau 2 et le niveau 3. Les brins en gras sont reliés. Par exemple, les brins numérotés 1 (resp. 2, 3 et 4) dans les deux représentations sont reliés. Sur cet exemple, il y a exactement 20 liaisons entre les deux niveaux.	102
4.18	Simplification du niveau 2 : (a) Objet euclidien, (b) Discrétisation Standard de l'objet (a), (c) Simplification de l'objet (a). (d) Discrétisation de l'objet (c).	103
4.19	Simplification du niveau discret analytique. (a) 2-G-carte du niveau 3. (b) Les sommets en gras sont supprimés. (c) 2-G-carte du niveau 2 obtenue.	103
4.20	Suppression d'un sommet topologique.	104
4.21	Liaisons entre le niveau 2 et le niveau 3 (cas d'une simplification). Les brins en gras sont reliés, alors que les brins en pointillés ne le sont pas.	104
4.22	Discrétisation Standard des contours : (a) Objet euclidien. (b) Discrétisation obtenue représentée à l'aide de pointels et lignels.	104
4.23	2-G-cartes et liaisons entre les niveaux 1 et 2.	105
4.24	Construction de la G-carte. Niveau 1 en cours de construction après la discrétisation Standard d'une arête.	105
4.25	Exemples de faces dégénérées : (a), (b) et (c) Divers étranglements possibles. (d) Face entièrement dégénérée.	106
4.26	Exemple de déconnexion de faces.	107
4.27	Exemple de déplacement des liaisons avant la suppression de brins reliés.	107
4.28	Exemples de niveaux 0 et 1.	108
4.29	Une image couleur segmentée : (a) Représentation discrète, (b) Représentation analytique discrète, (c) Représentation discrète des contours, (d) Représentation euclidienne.	111
4.30	Une image segmentée en niveaux de gris	113
4.31	Une partie de l'image <i>Lena</i>	116

Architecture d'un modeleur géométrique à base topologique d'objets discrets et méthodes de reconstruction en dimensions 2 et 3

Résumé :

Dans cette thèse nous nous intéressons à la question suivante : comment obtenir un objet continu à partir d'un objet discret ? Dans un premier temps, nous nous intéressons au problème de la reconnaissance de primitives discrètes, en proposant une notion de préimage généralisée. Cette préimage est un objet géométrique qui fournit l'ensemble des hyperplans discrets contenant les points discrets considérés. Nous en déduisons un nouvel algorithme incrémental de reconnaissance d'hyperplans discrets et de reconstruction inversibles en dimensions 2 et 3. Dans un deuxième temps, nous avons abordé la modélisation d'objets discrets sous la forme d'un outil de modélisation géométrique à base topologique permettant la manipulation d'objets tant sous forme continue que discrète. Les différentes représentations d'un même objet géométrique coexistent à l'intérieur d'une unique structure hiérarchique. Chaque niveau de la structure ainsi que les opérations internes sont décrites.

Mots-clés : géométrie discrète, reconnaissance, reconstruction inversible, modélisation géométrique et topologique, cartes généralisées.

Design of a topology based geometrical discrete modeler and reconstruction methods in 2D and 3D

Abstract:

Digital geometry provides tools to manipulate digital data such as digital images. Especially, relations between discrete and Euclidean objects are of interest. For instance, we try to answer the following questions : How to obtain digital objects from Euclidean ones, and in the same way, how to obtain Euclidean objects from digital ones? In this thesis, we study these two problems. First, we are interested in the digital hyperplane recognition problem which consists in determining if a digital point set belongs to a same digital hyperplane. In order to solve this problem, we propose the definition of a generalized preimage, defined in a parameter space, which provides the set of digital hyperplanes that contain all given digital points. We deduce a new incremental digital hyperplane recognition algorithm. We propose two digital objects invertible reconstruction methods in dimensions 2 and 3. In the second part of this work, we present the kernel of a modeling software that handles geometric objects represented in continuous and digital forms. Different representations of a same object coexist in an unique hierarchical structure. Each level of this structure as well as construction operations are detailed.

Keywords: Discrete geometry, recognition, invertible reconstruction, geometrical and topological modeling, G-maps.

