HAL open science

Normes de chanfrein et axe médian dans le volume discret

Eric Remy

To cite this version:

Eric Remy. Normes de chanfrein et axe médian dans le volume discret. Informatique [cs]. Université de la Méditerranée, 2001. Français. NNT : . tel-01502985

HAL Id: tel-01502985

https://amu.hal.science/tel-01502985v1

Submitted on 6 Apr 2017

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

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

Université de la Méditerranée
Aix-Marseille II

Faculté des Sciences de Luminy
163, Avenue de Luminy 13288 Marseille cedex 9

THÈSE DE DOCTORAT
en informatique

présentée par Eric Remy

en vue d'obtenir le grade de docteur de l'Université de la Méditerranée

Normes de chanfrein et axe médian dans le volume discret

soutenue le 17 décembre 2001 devant le jury composé de

Serge Miguet	Professeur, rapporteur
Annick Montanvert	Professeur, rapporteur
Christophe Fiorio	Maître de Conférences, examinateur
Jean Sequeira	Professeur, examinateur
Edouard Thiel	Maître de Conférences, directeur de thèse

N° attribué par la bibliothèque

Année

À tous ceux qui ont rendu ce travail possible.

Remerciements

Je tiens à remercier toutes les personnes qui ont rendu ce travail possible :

Je remercie Serge Miguet, pour avoir accepté d'être rapporteur de ce travail et pour les remarques pertinentes qu'il m'a faite lors de la finalisation du manuscrit.

Je remercie tout particulièrement Annick Montanvert, qui a accepté d'être rapporteur de cette thèse alors que son emploi du temps était particulièrement chargé, notamment par une soutenance « à la suédoise ». ;)

Je remercie Jean Sequeira pour m'avoir accueilli dans son équipe de recherche, pour son oreille attentive et pour le soin qu'il a toujours pris à préserver les thésards lorsque les problèmes surviennent.

Je remercie Christophe Fiorio et Jean-Claude Bajard qui ont tous deux accepté d'être examinateurs lors de la soutenance.

Je remercie Édouard Thiel, pour sa disponibilité, sa patience, sa rigueur scientifique dans la direction de ma thèse, ainsi que pour son amitié qui m'ont accompagné tout au long de ces trois ans.

Je remercie les membres du département d'informatique de Luminy et du Laboratoire d'Informatique de Marseille qui m'ont accueilli dans ces structures, avec une pensée particulière pour Solange Panattoni et Sylvie Calabrèse pour leur gentillesse et leur patience infinie avec moi.

Je remercie mes camarades de route : Nicolas (Nik) Denand, Christophe (ChrisRocket) Delpias, Laurent (ArchAngel) Astart, Marianne (Flocon) Michel et Hélène (Nell) Tran, avec qui j'ai partagé tous les hauts et les inévitables bas qui parsèment le parcours du thésard. Pour avoir toujours contribué à remettre du vent dans ma voile quand il en manquait, mon amitié leur est définitivement acquise. Je n'oublie pas non plus mes amis Jean-Luc Mari, Mounir Alaoui, Sébastien Mavromatis et Sébastien Richard.

Enfin, je tiens à remercier ma famille ainsi que mes amis de toujours (Danh, Eric, Agnès, Hélène, les Olivier(s), Anne et tous les autres), sans eux rien de tout cela n'aurait été envisageable, et ils ont de ce fait tous contribué à cet édifice.

À tous ceux que j'ai oublié, j'adresse mes excuses ; ils ne sont oubliés que sur le papier.

Eric Remy
le 13 décembre 2001

Table des matières

Table des matières **7**

Table des figures **11**

1. Introduction générale **13**

2. Définitions **15**
2.1 Introduction 15
2.2 Géométrie euclidienne 15
2.2.1 Combinaison linéaire 15
2.2.2 Enveloppe convexe 15
2.3 Géométrie discrète 16
2.3.1 Espace discret 16
2.3.2 Images et volumes numériques 16
2.3.3 Voisinage 17
2.3.4 Distances et normes discrètes 17
2.3.5 Exemples 18

3. Distances de chanfrein **19**
3.1 Définition des distances de chanfrein 19
3.1.1 Masque de chanfrein 19
3.1.2 Générateur du masque de chanfrein 20
3.1.3 Distances de chanfrein 21
3.1.4 Transformation de distance 22
3.1.5 Boules de chanfrein 25
3.1.6 Points visibles 26
3.1.7 Symétries 28
3.2 Étude des distances de chanfrein 28
3.2.1 La fonction d_C est une distance discrète 28
3.2.2 Un exemple surprenant 29
3.2.3 Régularité et semi-régularité 30
3.2.4 La semi-régularité n'est pas l'homogénéité 31
3.3 Conclusion 32

4. Normes de chanfrein **35**
4.1 Introduction 35
4.2 Norme de chanfrein dans l'espace continu 35
4.2.1 Masque rationnel équivalent 36

4.2.2 Cône continu 37
4.2.3 Déterminant d'un cône 37
4.2.4 Propriétés de $n_C^{\mathbb{R}}$ et enveloppe convexe des points de $\mathcal{M}'_C$ 37
4.2.5 Formule directe de $n_C^{\mathbb{R}}$ 41
4.3 Stratégie de construction d'une norme de chanfrein continue 43
4.3.1 Boule rationnelle équivalente 43
4.3.2 Générateur complété du masque de chanfrein 44
4.3.3 Convexité de la boule rationnelle équivalente 45
4.4 Retour au domaine discret 48
4.4.1 Cône régulier 48
4.4.2 Triangulation de Farey 50
4.4.3 Cônes non-réguliers et surcharge de $\mathcal{M}_C^g$ 52
4.5 Conclusion 53

5. Optimisation des normes de chanfrein 55
5.1 Introduction 55
5.2 Optimisation 55
5.3 Différentes méthodes et différents critères d'erreur 56
5.3.1 La méthode de Borgefors 56
5.3.2 Méthode de Verwer 57
5.4 Notre méthode 58
5.4.1 Comment énumérer ? 59
5.4.2 Surcharge du masque 59
5.4.3 Calcul de l'erreur 60
5.5 Application aux masques réguliers 61
5.6 Application aux masques non-réguliers 66
5.7 Conclusion 67

6. Calcul de l'axe médian 77
6.1 Introduction 77
6.2 Boules maximales et axe médian 77
6.3 Méthodes existantes pour le calcul de l'axe médian 78
6.3.1 Méthode des maxima locaux 78
6.3.2 Boules équivalentes 79
6.3.3 Tables de correspondances 80
6.4 Calcul de la Look-Up Table 81
6.4.1 Calcul de la table Lut 82
6.4.2 Calcul du voisinage de test $\mathcal{M}_{Lut}$ 84
6.5 Application 85
6.5.1 Application aux distances de chanfrein 85
6.5.2 Application à la distance euclidienne 88
6.6 Conclusion 91

7. Application **93**
7.1 Développement logiciel 93
7.2 Images d'exemple 96

8. Conclusion et perspectives **101**

Bibliographie **103**

Table des figures

2.1 Exemple d'ensemble non-convexe 15
2.2 Les différentes connexités 17
3.1 Les régions $\frac{1}{8}\mathbb{Z}^2$ et $\frac{1}{48}\mathbb{Z}^3$. 20
3.2 Diagramme de représentation de $\frac{1}{48}\mathbb{Z}^3$ en projection 20
3.3 Masque de chanfrein $\mathcal{M}_C$ et générateur $\mathcal{M}_C^g$ 21
3.4 Chemins de chanfrein 21
3.5 Algorithme CalculeDT 23
3.6 Demi-masque de chanfrein $\mathcal{M}_C^{1/2}$ 24
3.7 Carte de distance 24
3.8 Boule directe B_d et boule inverse B_d^{-1} 25
3.9 Points visibles 27
3.10 Points visibles de $\frac{1}{48}\mathbb{Z}^3$ en projection. 27
3.11 Symétriques des points visibles de $\mathcal{M}_C^g$ 28
3.12 Cas de $d_{3,1}$: $\mathcal{M}_C^g$ et chemin 29
3.13 Cas de $d_{3,1}$: carte de distance et boule 30
3.14 Contre-exemple : la régularité n'implique pas l'homogénéité 32
4.1 Masque $\mathcal{M}'_C$ à deux vecteurs. 38
4.2 Masque $\mathcal{M}'_C$ à quatre vecteurs. 39
4.3 Masque $\mathcal{M}'_C$ à six vecteurs. 41
4.4 Boules rationnelles équivalentes 44
4.5 Sous-triangulation $\triangle$ des boules rationnelles équivalentes 45
4.6 Tétraèdre (P,Q,R,S) direct. 46
4.7 Contraintes de convexité locale 47
4.8 Pavage des cônes 49
4.9 Exemples de déterminants 50
4.10 Triangulations compatibles 52
4.11 Triangulation régulière et non-Farey 52
5.1 Algorithme CalculeCT^g 60
5.2 Contraintes pour les masques {a,b,c} (triangulation naturelle) 61
5.3 Contraintes pour les masques {a,b,c} (triangulation non-naturelle) 62
5.4 Masques optimaux abc (triangulation naturelle) 63
5.5 Contraintes pour les masques {a,b,c,j} 64
5.6 Masques optimaux abcj 65
5.7 Contraintes pour les masques {a,b,c,e} 66
5.8 Masques optimaux {a,b,c,e} 67
5.9 Contraintes pour les masques {a,b,c,d,e} 68
5.10 Impossibilité de raffiner la triangulation pour {a,b,c,d,e} 68
5.11 Masques optimaux {a,b,c,d,e} 69

5.12 Contraintes pour les masques {a,b,c,d,e,f} 70
5.13 Masques optimaux {a,b,c,d,e,f} 71
5.14 Contraintes pour les masques {a,b,c,g,h,i,j,k,l,m} 72
5.15 Points de surcharge pour les masques {a,b,c,g,h,i,j,k,l,m} 73
5.16 Masques optimaux {a,b,c,g,h,i,j,k,l,m} 74
5.17 Surcharges pour les masques optimaux {a,b,c,g,h,i,j,k,l,m} 75
6.1 Axe médian et boules maximales. 78
6.2 Maximum local. 79
6.3 Erreur lors de l'extraction de l'axe médian 81
6.4 Test de recouvrement de deux boules dans $\frac{1}{8}\mathbb{Z}^2$. 82
6.5 Test de recouvrement translaté dans CT^g. 82
6.6 Algorithme CalculeLutCol 83
6.7 Algorithme CalculeEtVérifieLut 83
6.8 Algorithme EstAM^g 86
6.9 Algorithme CalculeDT^g 86
6.10 Différence entre (a) CT^g et (b) DT^g calculées sur $B_{5,7,\mathsf{c}=11}(27) \cap \frac{1}{8}\mathbb{Z}^2$. . . . 86
6.11 $\mathcal{M}_C^g$, $\mathcal{M}_{Lut}^g$ et Lut pour la distance 2D $d_{5,7,\mathsf{c}=11}$ 87
6.12 $\mathcal{M}_C^g$ et $\mathcal{M}_{Lut}^g$ pour la distance 2D $d_{14,20,\mathsf{c}=31,\mathsf{d}=44}$ 87
6.13 Exemples de Lut en 3D (cas où $\mathcal{M}_C^g = \mathcal{M}_{Lut}^g$) 88
6.14 $\mathcal{M}_{Lut}^g$ et Lut pour $d_{11,16,19,\mathsf{j}=45}$ 89
6.15 d_E^2 ne respecte pas l'inégalité triangulaire 90
6.16 Algorithme CalculeCT^g pour d_E^2 90
6.17 Algorithme CalculeDT^g pour d_E^2 général 91
6.18 Algorithme CalculeDT^g pour d_E^2 spécialisé 2D 92
7.1 Capture d'écran : GtkDv 95
7.2 Exemple : «4 boules» 97
7.3 Exemple : vertèbre 98
7.4 Exemple : cœur 99

Chapitre

1

Introduction générale

Avec le développement de l'informatique et des capteurs numériques, un nouveau champ d'étude est apparu au début des années 1960, afin de permettre un traitement numérique à la fois efficace et pertinent des images digitales. Différentes approches sont nées d'une même idée : remettre à plat toute la géométrie continue (euclidienne), et reconstruire, sur des bases saines, étape par étape, une *géométrie discrète* adaptée aux maillages réguliers des capteurs et des écrans.

Dans le cadre de l'analyse d'image, de nombreux travaux ont été menés sur la reconnaissance de formes, et plus particulièrement en 2D où les applications sont nombreuses (du guidage des missiles à la reconnaissance des caractères). Plus récemment avec le perfectionnement des appareils d'imagerie médicale (IRM, scanner, TEP, etc), est apparu le besoin de généraliser les techniques ainsi mises au point en 2D pour les rendre exploitables en 3D dans des *images volumiques discrètes*. Dans ce document, nous présentons notre contribution à cet effort de généralisation dans deux directions : l'étude des *distances de chanfrein* et le calcul de l'*axe médian* d'une forme.

Les distances de chanfrein, introduites par Montanari et Hilditch, et ensuite rendues populaires par Borgefors, appartiennent à la classe des distances discrètes, c'est-à-dire à l'ensemble des fonctions de distance définies sur la grille discrète ayant des valeurs entières. Leur avantage est de permettre l'approximation de la distance euclidienne dans le domaine discret tout en ayant un faible coût de calcul. Même si de nombreuses publications traitent des distances de chanfrein, la plupart des auteurs préfèrent visiblement se concentrer sur le problème de la minimisation de l'erreur commise dans l'approximation de la distance euclidienne. Seuls Montanari (dans le cas particulier des distances portant son nom), Thiel puis Borgefors et Kiselman se sont attachés à décrire leurs propriétés et leurs comportements. Nous apportons notre pierre à cet édifice, en présentant une définition différente de la régularité, puis une méthode permettant de construire des *normes de chanfrein*.

L'axe médian d'une forme discrète est l'équivalent du squelette d'une forme dans l'espace continu, tel qu'il a été défini par Blum en 1964. Plusieurs méthodes permettent son calcul, par exemple par de multiples érosions préservant l'homotopie, ou encore par calcul direct à partir de la carte de distance de la forme. Cette dernière méthode est bien connue dans les cas simples, quand les distances utilisées en 2D sont d_4, d_8, $d_{3,4}$ ou encore $d_{5,7,11}$; distances qui sont largement suffisantes en pratique. En 3D par contre, seul le calcul avec les distances d_6, d_{26} et $d_{3,4,5}$ semble avoir été maîtrisé, or ces distances sont très nettement insuffisantes en pratique pour obtenir la qualité escomptée. Il restait donc à développer une méthode qui soit à la fois indépendante de la dimension de l'espace considéré (2D, 3D, etc) mais aussi assez générique pour s'adapter à toutes les fonctions discrètes utilisables pour calculer la carte de distance.

Ce document s'articule en cinq parties. Le premier chapitre présente les notations et certaines définitions générales dont le reste du document dépend. On présente notamment la définition des distances et des normes discrètes.

Le deuxième chapitre définit les distances de chanfrein et précise le calcul de la carte de distance d'une forme. Nous présentons la régularité telle qu'elle est définie par Borgefors et nous montrons sur un contre-exemple, qu'elle ne suffit pas, à elle seule, à définir une norme discrète.

Au troisième chapitre, nous donnons une définition des distances de chanfrein dans l'espace continu, et nous introduisons la notion de *masque de chanfrein rationnel équivalent*. Nous montrons que le comportement de la distance est lié à la forme de l'enveloppe convexe des points du masque rationnel équivalent. Nous montrons que la distance de chanfrein continue est une application linéaire par morceaux, et nous montrons comment effectuer le *calcul direct* de la fonction en tout point. Ensuite nous montrons comment replonger ces résultats dans l'espace discret, et ainsi obtenir des *normes de chanfrein discrètes*.

Dans le chapitre quatre, nous présentons les trois critères principaux employés pour comparer les distances de chanfrein et la distance euclidienne. Pour différents masques de chanfrein, nous effectuons une minimisation de l'erreur commise pour deux de ces critères, afin d'obtenir des *normes de chanfrein optimales*.

Au chapitre cinq, nous présentons le calcul de l'axe médian d'une forme par la méthode des *tables de correspondance*. Nous présentons et justifions un algorithme de calcul des valeurs de la table, indépendant de la dimension et de la distance choisie. Nous introduisons la notion de *voisinage de test* nécessaire à la validité de la méthode, et nous donnons et justifions un algorithme pour le calculer efficacement. Enfin, nous donnons des exemples de résultats directement exploitables, pour certaines normes de chanfrein, ainsi que pour le carré de la distance euclidienne.

Le chapitre six donne quelques exemples d'applications des différents algorithmes et résultats présentés dans ce document, et expose le travail de développement logiciel effectué.

Chapitre 2

Définitions

2.1 Introduction

Dans ce court chapitre, nous présentons une partie de la terminologie et des notations qui seront utilisées tout au long de ce document. Nous rappelons quelques propriétés de géométrie euclidienne et de géométrie discrète qu'il est important d'avoir à l'esprit pour la bonne compréhension des sujets abordés ; nous rappelons notamment les définitions des distances et des normes dans les deux cas.

2.2 Géométrie euclidienne

Dans ce document, on note $\mathbb{N}$ l'ensemble des entiers naturels, $\mathbb{Z}$ l'ensemble des entiers relatifs, $\mathbb{Q}$ l'ensemble des rationnels et $\mathbb{R}$ l'ensemble des réels. Étant donné un ensemble $\mathbb{E}$ quelconque, on note $\mathbb{E}^+$ le sous-ensemble de $\mathbb{E}$ formé par les éléments positifs ou nuls. De même, on note $\mathbb{E}^*$ l'ensemble $\mathbb{E}$ privé de l'élément nul 0. On note enfin $\mathbb{E}^n$ le produit cartésien de n fois l'ensemble $\mathbb{E}$, c'est-à-dire $\mathbb{E}^n = \mathbb{E} \times \ldots \mathbb{E}$, n fois.

2.2.1 Combinaison linéaire

Soit A un ensemble fini de k points p_i de l'espace $\mathbb{R}^n$. Une *combinaisons linéaire* de points de A est une somme $\sum_{i=1}^{k} \lambda_i.p_i$, avec k un entier et λ_i des réels. Si de plus $\sum_{i=1}^{k} \lambda_i = 1$, on parle de *combinaison affine*.

2.2.2 Enveloppe convexe

On dit qu'un ensemble A est *convexe* si pour tout point x et y de A, tout point du segment (x,y) appartient à A (cf. figure 2.1).

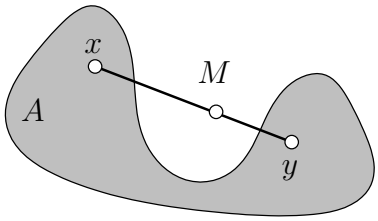


Fig. 2.1. *A n'est pas convexe car le point M du segment (x,y) n'appartient pas à A.*

Une combinaison linéaire affine telle que $\lambda_i \geqslant 0$ pour tout i, est une *combinaison linéaire convexe.*

On appelle *enveloppe convexe* d'un ensemble fini A de points de $\mathbb{R}^n$, et on note $\circledcirc A$, le sous-ensemble de $\mathbb{R}^n$ formé par toutes les combinaisons linéaires convexes des points de A. C'est le plus petit ensemble convexe de points de $\mathbb{R}^n$ qui contienne les points de A.

En deux dimensions, l'enveloppe convexe de A est un polygone, et on appelle *facette* de $\circledcirc A$ chaque segment qui constitue son contour. En trois dimensions, c'est un polyèdre, et les facettes de $\circledcirc A$ qui forment sa frontière sont des polygones. On appelle *points frontières* les points de $\circledcirc A$ qui sont sur les facettes, et *points intérieurs* les autres points.

Dans la suite, on se servira de la propriété suivante :

Propriété 1 *E est convexe si et seulement si $\circledcirc E = E$.*

La preuve est immédiate puisque $\circledcirc E$ est le plus petit ensemble convexe contenant E.

2.3 Géométrie discrète

On s'intéresse au traitement des images numériques. Notre objectif est d'opérer dans le même espace discret que celui qui contient l'information, plus précisément nous cherchons à éviter le passage par le domaine continu (les réels).

2.3.1 Espace discret

Nous notons $\mathbb{E}$ l'espace discret dans lequel sont plongées nos images. Suivant la dimension considérée, on prend pour $\mathbb{E}$ soit $\mathbb{Z}^2$, soit $\mathbb{Z}^3$.

On appelle *réseau fondamental* de $\mathbb{Z}^2$ (resp. $\mathbb{Z}^3$) l'ensemble $\mathbb{Z}^2$ (resp. $\mathbb{Z}^3$) muni de sa base canonique $(\overrightarrow{x}, \overrightarrow{y})$ (resp. $(\overrightarrow{x}, \overrightarrow{y}, \overrightarrow{z})$).

2.3.2 Images et volumes numériques

On appelle image (resp. volume) numérique I un tableau rectangulaire (resp. parallélépipédique rectangle) fini de points, à chacun desquels on associe une valeur dans $\mathbb{Z}$. On note $I[(x,y)]$ (resp. $I[(x,y,z)]$), la valeur du point de I de coordonnées $(x,y) \in \mathbb{Z}^2$ (resp. $(x,y,z) \in \mathbb{Z}^3$).

Traditionnellement, un point d'une image numérique est appelé *pixel* (contraction de l'anglais *picture element*) et, par analogie, un point d'un volume numérique est appelé *voxel* (de *volume element*).

On parle d'image (resp. volume) *binaire*, s'il n'existe que deux valeurs différentes dans I : 0 et 1. Dans l'image, les points à 1 représentent la forme $F \subset I$ et les points à 0 représentent le complément de la forme $\overline{F} \subset I$, aussi appelé complémentaire de la forme ou fond de l'image.

2.3.3 Voisinage

En 2D, un point q est dit *4-voisin* d'un point p s'il partage une arête avec p (cf. figure 2.2). L'ensemble des quatre points qui sont 4-voisins de p forment le *4-voisinage* de p. On dit de même que q est *8-voisin* de p s'ils partagent une arête ou un sommet. L'ensemble des huit points 8-voisins de p forme le *8-voisinage* de p.

En 3D, on dit qu'un point q est *6-voisin* d'un point p s'il partage une face avec p, *18-voisin* s'il partage une face ou une arête avec p, et enfin *26-voisin*, s'il partage une face, une arête ou un sommet avec p. Les 6-voisins, 18-voisins et 26-voisins d'un point p forment respectivement le *6-voisinage*, le *18-voisinage* et le *26-voisinage* de p.

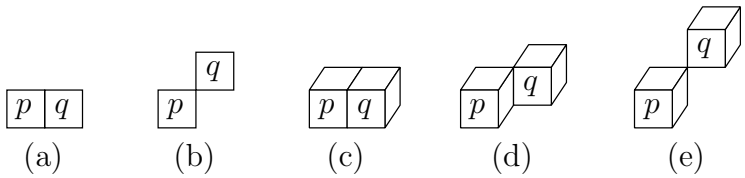


FIG. 2.2. *Les différentes connexités possibles entre p et q : en 2D, par une arête (a) et par un sommet (b) ; en 3D, par une face (c), par une arête (d) et par un sommet (e).*

On appelle chemin *n-connexe* (où $n = 4,8,6,18,26$) une suite de points $p_1, \ldots, p_k$ tels que deux points consécutifs p_i et p_{i+1} du chemin soient n-voisins l'un de l'autre.

2.3.4 Distances et normes discrètes

On appelle fonction discrète toute application $f : \mathbb{E} \times \ldots \times \mathbb{E} \to \mathbb{Z}$. Nous étudierons, en particulier, des fonctions de *distances discrètes* et de *normes discrètes*.

On appelle distance discrète, une fonction $d : \mathbb{E} \times \mathbb{E} \to \mathbb{N}$ qui satisfait les propriétés suivantes. Pour tout point $A,B,C \in \mathbb{E}$,

$$\text{définie positive} \qquad d(A,B) \geqslant 0 \; ; \; d(A,B) = 0 \iff A = B \tag{2.1}$$

$$\text{symétrique} \qquad d(A,B) = d(B,A) \tag{2.2}$$

$$\text{inégalité triangulaire} \qquad d(A,B) \leqslant d(A,C) + d(C,B)\,. \tag{2.3}$$

On appelle norme discrète, une fonction $n : \mathbb{E} \to \mathbb{N}$ qui satisfait les propriétés suivantes. Pour tout vecteur $\overrightarrow{x}, \overrightarrow{y} \in \mathbb{E}$ et tout $\lambda \in \mathbb{Q}$, tel que $\lambda.\overrightarrow{x} \in \mathbb{E}$,

$$\text{définie positive} \qquad n(\overrightarrow{x}) \geqslant 0 \; ; \; n(\overrightarrow{x}) = 0 \iff \overrightarrow{x} = \overrightarrow{0} \tag{2.4}$$

$$\text{homogénéité} \qquad n(\lambda.\overrightarrow{x}) = |\lambda|.n(\overrightarrow{x}) \tag{2.5}$$

$$\text{inégalité triangulaire} \qquad n(\overrightarrow{x} + \overrightarrow{y}) \leqslant n(\overrightarrow{x}) + n(\overrightarrow{y})\,. \tag{2.6}$$

Soit $n : \mathbb{E} \to \mathbb{N}$ une norme discrète, la *fonction de distance associée* $d : \mathbb{E} \times \mathbb{E} \to \mathbb{N}$ définie par $d : (\overrightarrow{x}, \overrightarrow{y}) \longmapsto n(\overrightarrow{x} - \overrightarrow{y})$ est une distance discrète, et elle a de plus les propriétés suivantes. Pour tout $\overrightarrow{x}, \overrightarrow{y}, \overrightarrow{z} \in \mathbb{E}$ et tout $\lambda \in \mathbb{Q}$ tel que $\lambda.\overrightarrow{x}, \lambda.\overrightarrow{y} \in \mathbb{E}$

$$\text{homogénéité} \qquad d(\lambda.\overrightarrow{x}, \lambda.\overrightarrow{y}) = |\lambda|.d(\overrightarrow{x}, \overrightarrow{y}) \tag{2.7}$$

$$\text{invariance par translation} \qquad d(\overrightarrow{x} + \overrightarrow{z}, \overrightarrow{y} + \overrightarrow{z}) = d(\overrightarrow{x}, \overrightarrow{y})\,. \tag{2.8}$$

2.3.5 Exemples

Pour deux points p et q quelconques de $\mathbb{E}$, on définit la distance $d_n(p,q)$ comme le nombre de points (sans compter le premier point) du plus court des n-chemins qui relient p à q. On définit de la sorte d_4 (aussi connue sous le nom de *city block distance*) et d_8 (*chessboard distance*) en 2D, et d_6, d_{18} et d_{26} en 3D.

Les fonctions d_4 en 2D et d_6 en 3D sont équivalentes à la norme $\ell_1 = \sum_{i=1}^{n} |x_i|$ de $\mathbb{R}^n$. Les fonctions d_8 en 2D et d_{26} en 3D sont équivalentes à la norme $\ell_\infty = \max_{i=1}^{n} |x_i|$ de $\mathbb{R}^n$. En conséquence, il est évident que ce sont des fonctions discrètes puisqu'elles ne peuvent prendre que des valeurs entières si p et q sont des points de $\mathbb{E}$, et qu'elles vérifient les propriétés des normes discrètes.

Rosenfeld et Pfaltz dans [Ros68], puis Borgefors dans [Bor84] étudient ces distances et décrivent leurs calculs. Borgefors montre également dans [Bor96] que d_{18} n'a pas un comportement simplement linéaire puisque tout point (x,y,z) tel que $z \leqslant y \leqslant x$, a une distance par rapport à l'origine égale à $d_{18}(x,y,z) = x + \max\{\, 0, (z + y - x + 1)/2 \,\}$.

Malheureusement, ces fonctions ne sont pas invariantes par rotation : si on prend p_1 et q_1 et qu'on leur applique une rotation pour trouver des points p_2 et q_2, il y a fort à parier que l'on n'aura pas $d_n(p_1,q_1) = d_n(p_2,q_2)$. Pour obtenir l'invariance par rotation, il faudrait pouvoir utiliser la distance euclidienne d_E c'est-à-dire la norme $\ell_2 = (\sum_i^n x_i^2)^{1/2}$, mais ce n'est pas une fonction discrète du fait de la racine carrée.

Remarquons maintenant qu'il ne suffit pas pour autant de prendre la partie entière ou l'arrondi de d_E, car de telles fonctions ne vérifient pas les propriétés des distances et des normes.

De nombreux autres exemples de distances discrètes sont donnés dans [Ros68], mais aucune ne présente autant d'attrait et de souplesse pour l'approximation de d_E que la classe de distance que nous allons étudier dans ce document : les distances de chanfrein.

Chapitre 3

Distances de chanfrein

Dans ce chapitre, nous présentons les distances de chanfrein en orientant notre présentation sur leur calcul effectif à partir des masques de chanfrein et de l'algorithme de Rosenfeld, qui permet de calculer la transformée de distance d'une forme. Nous montrons pourquoi prendre des masques symétriques. Nous définissons le générateur d'un masque, dont la connaissance est suffisante pour retrouver, par symétrie, le masque complet. Nous définissons les boules et les boules inverses. Nous montrons que les pondérations du masque doivent être choisies parmi l'ensemble des points visibles.

Ensuite, nous nous attachons à donner certains résultats connus sur le comportement des distances de chanfrein, dus à Verwer, Borgefors et Kiselman. Nous présentons la notion de régularité au sens de Borgefors et Kiselman. Nous donnons ensuite un contre-exemple montrant qu'elle n'est pas suffisante pour nous assurer que la fonction ait les propriétés d'une norme discrète.

3.1 Définition des distances de chanfrein

3.1.1 Masque de chanfrein

On cherche à construire une fonction de distance sur $\mathbb{E}$, à partir d'un ensemble réduit de distances locales dans un voisinage donné. Pour cela, on introduit la notion de pondération, puis celle de masque de chanfrein.

Définition 1 (Pondération) *On appelle pondération $M(\overrightarrow{v},w)$ un vecteur non-nul $\overrightarrow{v} \in \mathbb{E}^*$ auquel on associe un poids $w \in \mathbb{N}^*$.*

L'idée sous-jacente dans la définition des pondérations $(\overrightarrow{v_i},w_i)$ est de choisir des directions $\overrightarrow{v_i}$ suivant lesquelles le déplacement est autorisé et de fixer une mesure de *distance locale* w_i correspondant à la longueur de ce déplacement.

Définition 2 (Masque de chanfrein) *Un masque de chanfrein $\mathcal{M}_C$ est un ensemble central-symétrique de m pondérations $\mathcal{M}_C = \{\, (\overrightarrow{v_i},w_i) \,:\, 1 \leqslant i \leqslant m \,\}$ qui contient au moins une base de $\mathbb{E}$.*

Un masque est central-symétrique si pour toute pondération $(\overrightarrow{v_i},w_i)$, il existe une pondération $(\overrightarrow{v_j},w_j)$ de ce masque telle que $\overrightarrow{v_i} = -\overrightarrow{v_j}$ et $w_i = w_j$. Un tel masque est montré à la figure 3.3 ; il servira de base pour les exemples suivants.

3.1.2 Générateur du masque de chanfrein

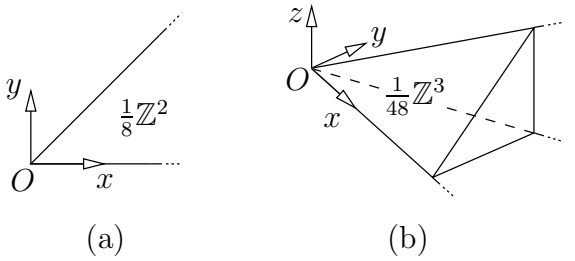
Nous ne nous intéresserons qu'à des masques de chanfrein qui sont symétriques par rapport à leur centre, aux axes $((O,\overrightarrow{x})$, $(O,\overrightarrow{y})$ et $(O,\overrightarrow{z}))$ et aux bissectrices de ces axes. Concrètement, un masque de chanfrein est pour nous 8-symétrique en 2D et 48-symétrique en 3D. Nous introduisons deux sous-ensembles, qui sont les zones dans lesquelles se trouvent les vecteurs suffisant à retrouver tout le masque par symétrie : le premier octant du plan $\mathbb{Z}^2$

$$\tfrac{1}{8}\mathbb{Z}^2 = \left\{ (x,y) \in \mathbb{Z}^2 \ : \ 0 \leqslant y \leqslant x \right\}$$

et le premier 48ème de l'espace $\mathbb{Z}^3$ (cf. Figure 3.1)

$$\tfrac{1}{48}\mathbb{Z}^3 = \left\{ (x,y,z) \in \mathbb{Z}^3 \ : \ 0 \leqslant z \leqslant y \leqslant x \right\} \ .$$

On représente souvent dans ce document, des vecteurs de $\frac{1}{48}\mathbb{Z}^3$ sur un diagramme en projection tel que celui de la figure 3.2. Sur un tel diagramme, un point (x,y,z) de $\frac{1}{48}\mathbb{Z}^3$ sera transformé en un point $(y/x,z/x)$ de $\mathbb{Q}^2$. On se souviendra cependant que cette représentation fait perdre les longueurs relatives des vecteurs.



(a) (b)

FIG. 3.1. *Les régions $\frac{1}{8}\mathbb{Z}^2$ (a) et $\frac{1}{48}\mathbb{Z}^3$ (b).*

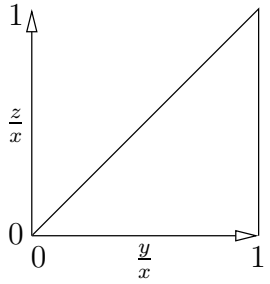


FIG. 3.2. *Diagramme de représentation de $\frac{1}{48}\mathbb{Z}^3$ en projection.*

Définition 3 (Générateur d'un masque de chanfrein) *On appelle générateur d'un masque de chanfrein $\mathcal{M}_C^g$, le sous-ensemble d'un masque de chanfrein $\mathcal{M}_C$ qui est inclus dans $\frac{1}{8}\mathbb{Z}^2$ (ou $\frac{1}{48}\mathbb{Z}^3$ suivant la dimension). Ce sous-ensemble est suffisant pour retrouver par symétrie tous les vecteurs de $\mathcal{M}_C$. On a*

$$\mathcal{M}_C^g = \left\{ \big((x_i,y_i),w_i\big) \in \mathcal{M}_C \ : \ (x_i,y_i) \in \tfrac{1}{8}\mathbb{Z}^2 \right\} \quad \text{en 2D et}$$

$$\mathcal{M}_C^g = \left\{ \big((x_i,y_i,z_i),w_i\big) \in \mathcal{M}_C \ : \ (x_i,y_i,z_i) \in \tfrac{1}{48}\mathbb{Z}^3 \right\} \quad \text{en 3D.}$$

On appelle *taille* du masque de chanfrein, la longueur du côté du plus petit carré en 2D et du plus petit cube en 3D qui englobe les vecteurs du masque. La taille d'un masque $\mathcal{M}_C$ se déduit facilement de son générateur et vaut $1 + 2.\max\{ x_i \ : \ \big((x_i,y_i,z_i),w_i\big) \in \mathcal{M}_C^g \}$.

On montre sur la Figure 3.3, un masque de chanfrein 2D de taille 5, dont le générateur $\mathcal{M}_C^g$ (en gras) comporte les pondérations $\big((1,0),5\big)$, $\big((1,1),7\big)$ et $\big((2,1),11\big)$. On remarque aussi que la partie grise $\frac{1}{8}\mathbb{Z}^2$ est bornée par les deux vecteurs (1,0) et (1,1) qui forment une

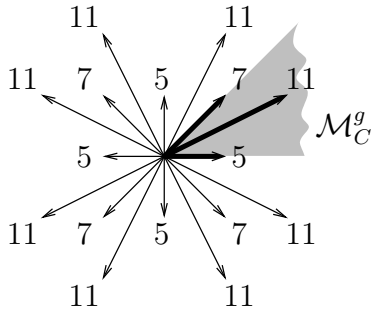


FIG. 3.3. *Masque 2D et son générateur* $\mathcal{M}_C^g = \{((1,0),5),\ ((1,1),7),\ ((2,1),11)\}$ *(en gras) dans* $\frac{1}{8}\mathbb{Z}^2$ *(en gris).*

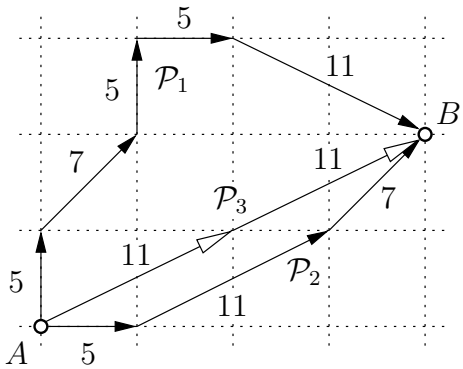


FIG. 3.4. *3 chemins de A à B utilisant le masque de la figure 3.3.*

base de ce sous-espace. De même en 3D, les vecteurs (1,0,0), (1,1,0) et (1,1,1) forment une base de $\frac{1}{48}\mathbb{Z}^3$. Ces deux ensembles de vecteurs jouent un rôle particulier dans le masque et sont de ce fait toujours présents dans $\mathcal{M}_C^g$; en conséquence, tout masque de chanfrein sera toujours de taille au moins égale à 3.

3.1.3 Distances de chanfrein

Soit A et B deux points de $\mathbb{E}$. On appelle chemin de A à B et on note $\mathcal{P}_{AB}$, une suite de l vecteurs $(\overrightarrow{v_{i_j}})_{1 \leqslant j \leqslant l}$ où $\forall j$, $\overrightarrow{v_{i_j}} \in \mathcal{M}_C$, permettant de connecter A à B, c'est-à-dire

$$\overrightarrow{AB} = \sum_{j=1}^{l} \overrightarrow{v_{i_j}} \quad \text{où } \overrightarrow{v_{i_j}} \in \mathcal{M}_C\,.$$

En regroupant les $\overrightarrow{v_{i_j}}$ identiques et en notant n_i leurs nombres d'occurrences, on obtient

$$\overrightarrow{AB} = \sum_{i=1}^{m} n_i.\overrightarrow{v_i} \quad \text{où } \overrightarrow{v_i} \in \mathcal{M}_C\,.$$

On peut alors calculer le coût W du chemin $\mathcal{P}_{AB}$ qui est la somme des poids associés à chaque $\overrightarrow{v_{i_j}}$ composant le chemin :

$$W(\mathcal{P}_{AB}) = \sum_{i=1}^{m} n_i.w_i\,.$$

On pose évidement que le coût du chemin vide (qui n'est formé d'aucun vecteur) est nul. D'autre part, on sait que quels que soient deux points de $\mathbb{E}$, il existe toujours au moins un chemin les reliant puisque le masque $\mathcal{M}_C$ contient des pondérations qui forment au moins par définition une base de $\mathbb{E}$.

Définition 4 (Distance de chanfrein) *On appelle distance de chanfrein d_C entre deux points A et B de $\mathbb{E}$, le minimum des coûts associés à tous les chemins $\mathcal{P}_{AB}$ de longueur finie, liant A à B :*

$$d_C(A,B) = \min_{\mathcal{P}_{AB}} W(\mathcal{P}_{AB})\,.$$

À la Figure 3.4, trois chemins entre deux points A et B sont représentés en utilisant les vecteurs du masque de la figure 3.3. Les coûts de ces trois chemins sont $W(\mathcal{P}_1) = 33$, $W(\mathcal{P}_2) = 23$ et $W(\mathcal{P}_3) = 22$. Par définition, $d_C(A,B)$ est le minimum des coûts de tous les chemins de A à B, c'est à dire $\min\{33,23,22,...\}$. Ici, on se convainc aisément que le minimum est atteint avec le coût du chemin $\mathcal{P}_3$ (matérialisé par les flèches à têtes blanches). La justification de cette affirmation découle des résultats du prochain chapitre.

Lorsqu'une comparaison avec la distance euclidienne sera nécessaire, on fera intervenir dans la formule un *facteur d'échelle* réel, noté s :

$$d_C(A,B) = \frac{1}{s} \min_{\mathcal{P}_{AB}} W(\mathcal{P}_{AB}) \,.$$

Comme on le voit sur la figure 3.4, la distance mesurée entre A et B par la distance de chanfrein ($d_C(A,B) = 22$) est presque 5 fois plus grande que celle mesurée avec la distance euclidienne ($d_E(A,B) = \sqrt{20} \approx 4.47$). La valeur exacte de s dépend du choix effectué pendant le calcul des distances locales w_i du masque : cette phase d'optimisation sera explicitée au chapitre 5. On gardera simplement en mémoire, pour l'instant, que la valeur de s est proche de celle de la distance locale affectée au vecteur (1,0) en 2D ou (1,0,0) en 3D. La plupart du temps, nous omettrons ce facteur d'échelle puisque nous n'effectuerons que des comparaisons entre des valeurs découlant d'une même distance.

3.1.4 Transformation de distance

La transformation de distance (ou carte de distance, de l'anglais *distance map*), notée DT, consiste à étiqueter chaque point d'une forme dans une image à sa distance au complémentaire de la forme. Autrement dit, étant donné une distance discrète d et une image binaire contenant une forme F, la transformée de distance donne une image DT telle que

$$\forall p \in DT,\, DT(p) = \min\left\{\, d(p,q) \;:\; q \in \overline{F} \,\right\}. \tag{3.1}$$

Rosenfeld propose et justifie dans [Ros66], un algorithme séquentiel en deux passages sur l'image, permettant de réaliser la transformation de distance pour les distances d_4 et d_8. Montanari définit ensuite une classe de distances qui sont les précurseurs des distances de chanfrein : on retrouve la notion de masque, mais les valeurs des distances locales sont fixées à leur distance euclidienne (les poids sont donc réels et non entiers). Il reprend l'algorithme de Rosenfeld dans [Mon68] et le justifie pour cette famille de distance. Hilditch dans [Hil69] puis Barrow dans [Bar77] étudient les premiers masques de chanfrein ($d_{2,3}$ en 2D), et adaptent l'algorithme de Montanari sans le justifier. Borgefors enfin, dans [Bor84], donne un algorithme adapté à tout masque de chanfrein de taille 3 en dimensions de 1D à 4D, et bien qu'elle ne donne pas de justification formelle, il est communément admis que cet algorithme est correct, comme elle l'affirme, si les distances locales restent « raisonnablement proches (en proportion) de leurs valeurs pour la distance euclidienne ».

L'algorithme qui est proposé à la Figure 3.5 est le même que celui de Borgefors généralisé à un voisinage quelconque, sauf pour ce qui est de l'ordre de balayage de l'image. Ce changement est sans conséquence sur le résultat mais influe sur la définition du demi-

```
    Procedure CalculeDT ( M_C^{1/2}, DT, x_max, y_max, z_max) ;
 1  for x = 0 to x_max, for y = 0 to y_max, for z = 0 to z_max do     // Balayage avant.
 2    if DT[(x,y,z)] ≠ 0 then                                  // c.-à-d. si on est dans l'objet.
 3      {
 4        value = +∞ ;
 5        for each (v_i,w_i) in M_C^{1/2} do
 6          value = min { value ,DT[(x,y,z) − v_i] + w_i } ;
 7        DT[(x,y,z)] = value ;
 8      }
 9  for x = x_max to 0, for y = y_max to 0, for z = z_max to 0 do    // Balayage arrière.
10    if DT[(x,y,z)] ≠ 0 then
11      {
12        value = DT[(x,y,z)] ;
13        for each (v_i,w_i) in M_C^{1/2} do
14          value = min { value ,DT[(x,y,z) + v_i] + w_i } ;
15        DT[(x,y,z)] = value ;
16      }
```

FIG. 3.5. *Algorithme d'extraction de la carte de distance.* ***Entrées*** *:* $\mathcal{M}_C^{1/2}$ *le demi-masque de la distance choisie, DT l'image binaire sur laquelle le calcul est demandé,* $x_{\max}$*,* $y_{\max}$ *et* $z_{\max}$ *sont les bornes maximales des coordonnées dans l'image.* ***Sorties*** *: chaque point de DT est étiqueté à sa valeur de distance au fond.*

masque $\mathcal{M}_C^{1/2}$ qui est utilisé dans l'algorithme (cf. figure 3.6) :

$$\mathcal{M}_C^{1/2} = \left\{ \big((x_i,y_i),w_i\big) \in \mathcal{M}_C \ : \ \begin{array}{l} (x_i > 0) \ ou \\ (x_i = 0) \ et \ (y_i > 0) \end{array} \right\} \quad \text{en 2D et}$$

$$\mathcal{M}_C^{1/2} = \left\{ \big((x_i,y_i,z_i),w_i\big) \in \mathcal{M}_C \ : \ \begin{array}{l} (x_i > 0) \ ou \\ (x_i = 0) \ et \ (y_i > 0) \ ou \\ (x_i = 0) \ et \ (y_i = 0) \ et \ (z_i > 0) \end{array} \right\} \quad \text{en 3D.}$$

Ce changement de définition du demi-masque nous permet en particulier de toujours avoir $\mathcal{M}_C^g \subseteq \mathcal{M}_C^{1/2} \subset \mathcal{M}_C$ comme on peut le voir sur la figure 3.6 (a).

On applique l'algorithme (sans tenir compte des coordonnées z puisqu'on est ici en 2D) à l'image (a) de la figure 3.7. On applique un balayage en x croissant (de gauche à droite) et en y croissant (de haut en bas) comme indiqué sur la figure 3.6 (b), en cherchant pour chaque point $DT[(x,y)]$ de l'image quel est le vecteur $-\overrightarrow{v_i}$ du demi-masque $-\mathcal{M}_C^{1/2}$ de la figure 3.6 qui permet d'obtenir la plus petite valeur de $DT[(x,y) - \overrightarrow{v_i}] + w_i$. A l'issue de ce premier balayage, on obtient l'image 3.7 (b). Il reste à faire le deuxième balayage en x décroissant et y décroissant en cherchant pour le point $DT[(x,y)]$ le $\overrightarrow{v_i}$ de $\mathcal{M}_C^{1/2}$ qui minimise $DT[(x,y) + \overrightarrow{v_i}] + w_i$. Si le minimum ainsi trouvé reste plus grand que la valeur trouvée au premier passage et qui est conservée en $DT[(x,y)]$, alors on l'ignore. On obtient finalement l'image 3.7 (c).

En remplaçant le calcul de minimum par un calcul de maximum dans l'algorithme précédant, on obtient la *transformation de distance inverse*. Cet algorithme permet à partir

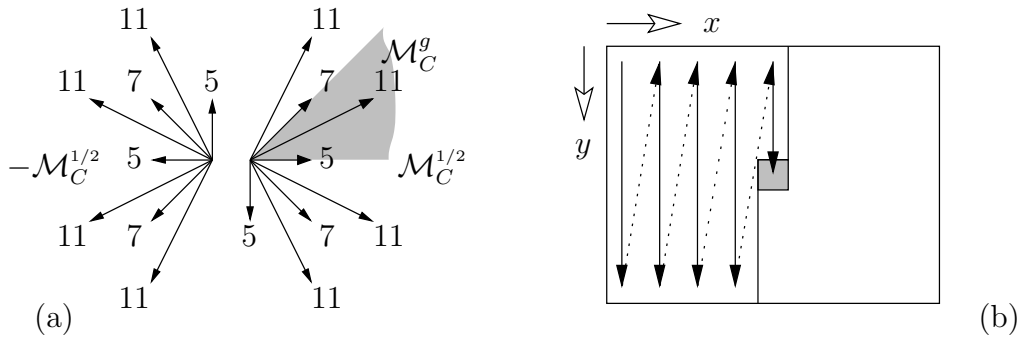


FIG. 3.6. *En (a), les deux demi-masques $\mathcal{M}_C^{1/2}$ et son symétrique $-\mathcal{M}_C^{1/2}$ de $d_{5,7,c=11}$. En (b), le premier balayage avec le demi-masque $-\mathcal{M}_C^{1/2}$ de haut en bas et de droite à gauche ; le deuxième balayage avec $\mathcal{M}_C^{1/2}$ est symétrique.*

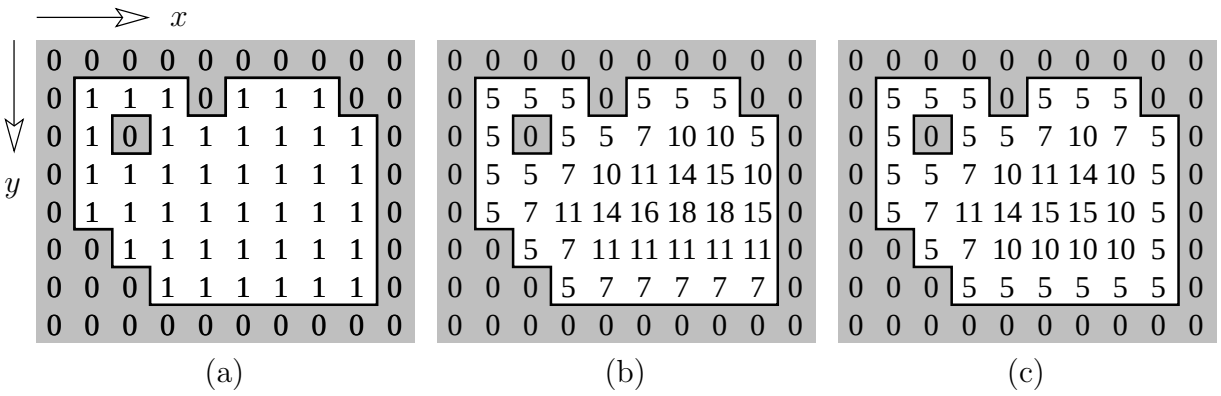


FIG. 3.7. *Image binaire (a), résultat de la première passe de l'algorithme (b) et résultat final (c).*

de points $p_1 \dots p_n$ d'une image, initialisés à des valeurs $r_1 \dots r_n$, de marquer tous les points p qui sont à une distance telle que $d_C(p,p_i) < r_i$. Cet algorithme permet à partir de certains points particuliers d'une carte de distance d'un objet F (les points de l'axe médian de F, dont le calcul est détaillé au chapitre 6) et du masque de chanfrein qui a servi à les obtenir, de reconstruire tous les points de F sans perte.

Pour les deux algorithmes (transformation directe et transformation inverse), le coût de calcul dépend linéairement à la fois du nombre de points dans l'image et du nombre de vecteurs dans le masque de chanfrein. Soit m_C le nombre de pondérations du masque, et soit L la longueur du côté de l'image de dimension n, la complexité de ces deux algorithmes est donc de l'ordre de $O(m_C.L^n)$.

3.1.5 Boules de chanfrein

Définition 5 (Boule de chanfrein) *On appelle boule de chanfrein B_d de centre p et de rayon r, l'ensemble des points q défini par $B_d(p,r) = \{\, q \in \mathbb{E} \;:\; d(p,q) \leqslant r \,\}$.*

Cette notion classique de topologie va nous être d'une importance capitale dans la suite de ce document. En pratique, on peut obtenir une boule de chanfrein en prenant une image carrée de côté $2.n+1$, où n est une valeur choisie suffisamment grande pour que la boule ne déborde pas de l'image. On initialise tous les points de l'image à une valeur infinie, sauf le centre que l'on initialise à 0 : ce sera le centre de la boule. Les points extérieurs à l'image doivent être considérés comme ayant eux aussi une valeur infinie, mais ils ne doivent évidement pas pouvoir être changés par l'algorithme. On opère ensuite une transformation de distance et on garde les points dont la valeur est inférieure au rayon. La figure 3.8 (a) est le résultat de ce calcul pour un rayon de 21 et en utilisant le masque de la figure 3.6.

35	32	30	27	26	25	26	27	30	32	35
32	28	25	22	21	20	21	22	25	28	32
30	25	21	18	16	15	16	18	21	25	30
27	22	18	14	11	10	11	14	18	22	27
26	21	16	11	7	5	7	11	16	21	26
25	20	15	10	5	0	5	10	15	20	25
26	21	16	11	7	5	7	11	16	21	26
27	22	18	14	11	10	11	14	18	22	27
30	25	21	18	16	15	16	18	21	25	30
32	28	25	22	21	20	21	22	25	28	32
35	32	30	27	26	25	26	27	30	32	35

(a)

0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	2	1	0	0	0	0
0	0	1	4	6	7	6	4	1	0	0
0	0	4	8	11	12	11	8	4	0	0
0	1	6	11	15	17	15	11	6	1	0
0	2	7	12	17	22	17	12	7	2	0
0	1	6	11	15	17	15	11	6	1	0
0	0	4	8	11	12	11	8	4	0	0
0	0	1	4	6	7	6	4	1	0	0
0	0	0	0	1	2	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0

(b)

FIG. 3.8. *Pour la distance $d_{5,7,c=11}$, boule directe B_d de rayon 21 (a), et boule inverse B_d^{-1} de rayon 22 (b).*

Bien que dans le domaine discret, on ne distingue pas la boule ouverte de la boule fermée (puisque la boule fermée contient les mêmes points que la boule ouverte dont le rayon est augmenté de 1), on distingue par contre la boule directe de la boule inverse :

Définition 6 (Boule inverse) *La boule inverse $B_d^{-1}(p,r)$ de centre p et de rayon r est l'ensemble $B_d^{-1}(p,r) = \{\, q \in \mathbb{E} \;:\; r - d(p,q) > 0 \,\}$.*

La boule inverse se calcule toujours sur une image de côté $2.n+1$, en mettant la valeur du rayon r au centre et des 0 partout ailleurs, puis en utilisant l'algorithme de transformation de distance inverse. La figure 3.8 (b) montre le résultat d'un tel calcul. On remarque que la boule inverse de rayon 22 de la figure 3.8 (b) comporte les mêmes points que la boule

directe 3.8 (a) de rayon 21 mais que leurs valeurs diffèrent ; ainsi géométriquement parlant, on a :

Lemme 1 $B_d(p,r) = B_d^{-1}(p,r+1)$. [Thi94]

Preuve : Pour tout $p \in \mathbb{E}$ et tout rayon $r \in \mathbb{N}$, on a

$$\begin{aligned} q \in B_d(p,r) \quad &\Longleftrightarrow \quad d(p,q) \leqslant r \iff d(p,q) - (r+1) < 0 \\ &\Longleftrightarrow \quad (r+1) - d(p,q) > 0 \iff q \in B_d^{-1}(p,r+1). \quad \square \end{aligned}$$

Verwer dans [Ver91b] affirme sans donner de justifications que les boules de chanfrein sont des polygones discrets en 2D et des polyèdres discrets en 3D. Nous le démontrerons dans le cadre des normes de chanfrein au chapitre 4.

3.1.6 Points visibles

Au vu de l'algorithme de la Figure 3.5, il apparaît clairement qu'il n'est pas souhaitable d'avoir dans le générateur $\mathcal{M}_C^g$ du masque, deux pondérations $(\overrightarrow{v_i},w_i)$ et $(\overrightarrow{v_j},w_j)$ telles que $\overrightarrow{v}_i^g = \lambda.\overrightarrow{v}_j^g$, $\lambda \in \mathbb{N}$. En effet, si $w_i = \lambda.w_j$ alors les pondérations sont redondantes et ralentissent le calcul inutilement. Au contraire, si $w_i \neq \lambda.w_j$ alors les pondérations sont contradictoires : le poids d'un chemin comportant λ fois $\overrightarrow{v}_j^g$ pourra être calculé soit avec w_i, soit avec $\lambda.w_j$, ce qui donne deux coûts différents possibles pour un même chemin et, bien que le calcul de minimum fasse toujours choisir la valeur la plus faible suivant le cas, cela aboutit inévitablement à rompre l'homogénéité des valeurs de distance dans cette direction. Pour ces raisons, on ne choisit pour construire $\mathcal{M}_C^g$ que des *points visibles* :

Définition 7 (Point visible) *Un point $P(x,y) \in \mathbb{Z}^2$ (resp. $P(x,y,z) \in \mathbb{Z}^3$) est dit visible depuis l'origine s'il n'y a aucun point du réseau fondamental de $\mathbb{Z}^2$ (resp. $\mathbb{Z}^3$) qui soit situé sur le segment continu $[O,P]$ en dehors de O et P. Une condition nécessaire et suffisante est* $\mathrm{pgcd}(x,y) = 1$ *(resp.* $\mathrm{pgcd}(x,y,z) = 1$*)* [Har78, chap. 3].

On trouve les points visibles avec un crible en x, y et z, en appliquant le même principe que le crible d'Eratostène pour les nombres premiers : tout point est considéré comme visible à l'initialisation, puis pour chaque point examiné, on élimine de l'ensemble des points visibles tous ses multiples. Le résultat d'une telle recherche dans $\frac{1}{8}\mathbb{Z}^2$ et dans $\frac{1}{48}\mathbb{Z}^3$, est donné figure 3.9.

On définit une distance de chanfrein en donnant la liste des vecteurs et des distances locales qui forment le générateur de son masque. Par exemple, on notera $d_{\mathsf{a}=5,\mathsf{b}=7,\mathsf{c}=11}$ en 2D, et $d_{\mathsf{a}=4,\mathsf{b}=6,\mathsf{c}=7,\mathsf{d}=9,\mathsf{e}=10}$ en 3D, les distances qui correspondent respectivement aux générateurs

$$\mathcal{M}_C^g = \{\, \big((1,0),5\big),\big((1,1),7\big),\big((2,1),11\big) \,\} \quad \text{et}$$

$$\mathcal{M}_C^g = \{\, \big((1,0,0),4\big),\big((1,1,0),6\big),\big((1,1,1),7\big),\big((2,1,0),9\big),\big((2,1,1),10\big) \,\}\,.$$

Les points visibles $\mathsf{a} = (1,0)$ et $\mathsf{b} = (1,1)$ en 2D, et $\mathsf{a} = (1,0,0)$, $\mathsf{b} = (1,1,0)$ et $\mathsf{c} = (1,1,1)$ en 3D étant toujours présents dans $\mathcal{M}_C^g$, on écrit, par abus de langage et par souci de concision

(a)

a	(1,0)
b	(1,1)
c	(2,1)
d	(3,1)
e	(3,2)
f	(4,1)
g	(4,3)

a	(1,0,0)
b	(1,1,0)
c	(1,1,1)
d	(2,1,0)
e	(2,1,1)
f	(2,2,1)

g	(3,1,0)
h	(3,1,1)
i	(3,2,0)
j	(3,2,1)
k	(3,2,2)
l	(3,3,1)
m	(3,3,2)

n	(4,1,0)
o	(4,1,1)
p	(4,2,1)
q	(4,3,0)
r	(4,3,1)
s	(4,3,2)
t	(4,3,3)
u	(4,4,1)
v	(4,4,3)

(b)

FIG. 3.9. *Les premiers points visibles en 2D (a) et en 3D (b) (nom, coordonnées) pour* $x \leqslant 4$.

dans la notation, $d_{5,7,\mathsf{c}=11}$ et $d_{4,6,7,\mathsf{d}=9,\mathsf{e}=10}$. Autant que possible, on confond aussi la distance locale w_{a} et le point visible a de manière à obtenir des notations qui sont comparables à celles de Borgefors et d'autres auteurs qui désignent par a non pas le vecteur mais la valeur de distance locale qui lui est attachée.

On désigne par $\mathcal{V}_i$ l'ensemble des points visibles de $\frac{1}{8}\mathbb{Z}^2$ ou $\frac{1}{48}\mathbb{Z}^3$ tels que $x \leqslant i$ et on note $\mathcal{V}_i^\star = \mathcal{V}_i \setminus \mathcal{V}_{i-1}$, une couche unique à $x = i$ constant de cet ensemble. On montre à la figure 3.10, les premiers points visibles 3D en utilisant la projection perspective définie au §3.1.2.

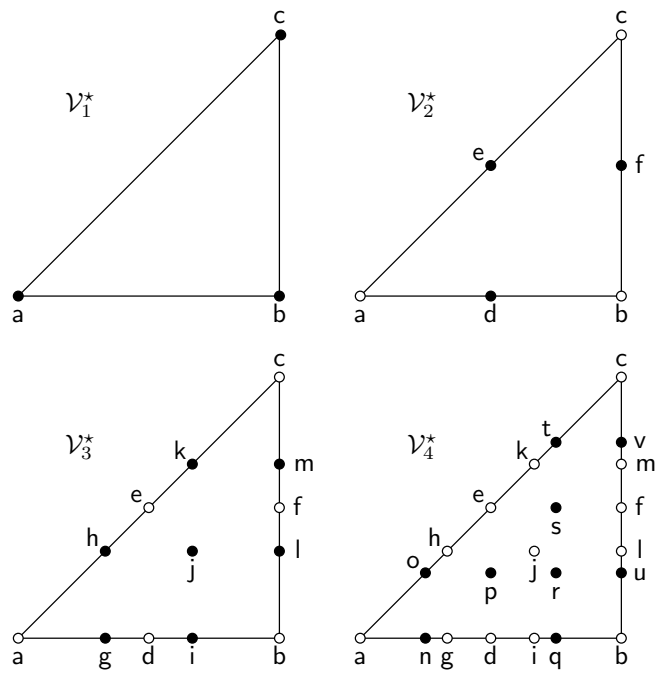


FIG. 3.10. *Points visibles de* $\frac{1}{48}\mathbb{Z}^3$ *en projection (points* $\bullet$ *de* $\mathcal{V}_n^\star$*, points* $\circ$ *de* $\mathcal{V}_{n-1}$*).*

3.1.7 Symétries

Pour désigner un vecteur précis du masque de chanfrein $\mathcal{M}_C$, on donne sa construction par symétrie à partir du vecteur correspondant de $\mathcal{M}_C^g$. On note ainsi $\overrightarrow{v}/\mathsf{a}$ le symétrique du vecteur $\overrightarrow{v}$ d'un masque 2D par rapport à la droite (O,a), et de même, on note $\overrightarrow{v}/\mathsf{ab}$, le symétrique d'un vecteur $\overrightarrow{v}$ par rapport au plan $(O,\mathsf{a},\mathsf{b})$. On a les trois fonctions suivantes :

$$\begin{array}{rcl} (x,y,z) & \longmapsto & (x,y,z)/\mathsf{ab} = (x,y,-z)\,, \\ (x,y,z) & \longmapsto & (x,y,z)/\mathsf{bc} = (y,x,z) \quad \text{et} \\ (x,y,z) & \longmapsto & (x,y,z)/\mathsf{ac} = (x,z,y)\,. \end{array}$$

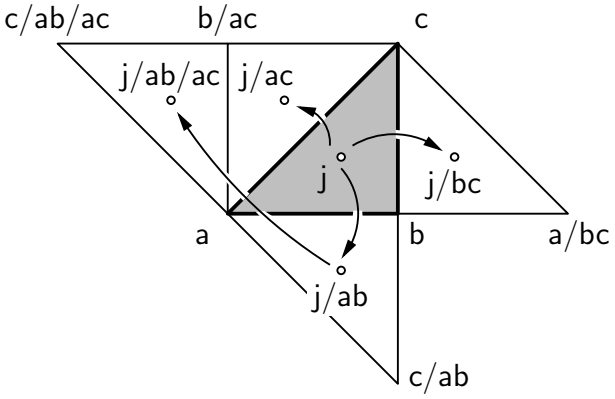


FIG. 3.11. *Symétriques des points visibles de $\mathcal{M}_C^g$ en projection.*

Par exemple sur la figure 3.11, pour un générateur contenant la pondération $(\overrightarrow{v_j},w_j)$, où j est le point visible (3,2,1), on désigne respectivement par j/ab, j/bc, j/ac, les vecteurs (3,2, − 1), (2,3,1) et (3,1,2). Ces trois vecteurs ont évidement la même distance locale w_{j} que le point j. Sur le même principe, on note $\overrightarrow{v}/\mathsf{ab}/\mathsf{ac}$ (plutôt que $(\overrightarrow{v}/\mathsf{ab})/\mathsf{ac}$) le vecteur obtenu par symétrie de plan $(O,\mathsf{a},\mathsf{c})$ à partir du vecteur $\overrightarrow{v}/\mathsf{ab}$.

3.2 Étude des distances de chanfrein

3.2.1 La fonction d_C est une distance discrète

Verwer [Ver91a] montre qu'une distance de chanfrein satisfait toujours les propriétés d'une distance discrète, telles qu'elles sont définies au §2.3.4.

Théorème 1 *Étant donné un masque de chanfrein $\mathcal{M}_C$, la distance de chanfrein correspondante d_C est bien une distance discrète.*

Preuve : Soient A, B et C trois points de $\mathbb{E}$.

▷ ***Définie positive*** : Si $A = B$ alors $d_C(A,B) = 0$ puisque le chemin nul est le chemin de poids minimal reliant A à B. Inversement, si $A \neq B$ alors il existe au moins un chemin de A à B, car le vecteur unitaire $\overrightarrow{v_{\mathsf{a}}}$ est toujours présent dans le masque et peux donc servir à

la réalisation d'un chemin entre tout couple de point de $\mathbb{E}$. Comme le chemin ainsi formé contient au moins un vecteur du masque et comme tous les vecteurs du masque ont un poids strictement positif alors $W(\mathcal{P}_{AB}) > 0$.

▷ ***Symétrique*** : Si $\mathcal{P}_1 = (\overrightarrow{v_{i_1}}, \ldots, \overrightarrow{v_{i_m}})$ est un chemin minimal de A à B, construit à partir des pondérations du masque, alors on peut construire le chemin $\mathcal{P}_2 = (-\overrightarrow{v_{i_m}}, \ldots, -\overrightarrow{v_{i_1}})$ de B à A, qui est composé des vecteurs symétriques mis dans l'ordre inverse, car le masque est symétrique (c'est-à-dire que pour tout vecteur $\overrightarrow{v_i}$, il existe un vecteur $\overrightarrow{v_j}$ du masque tel que $\overrightarrow{v_i} = -\overrightarrow{v_j}$ et $w_i = w_j$). Par construction, on a donc $W(\mathcal{P}_2) = W(\mathcal{P}_1)$. Si $\mathcal{P}_2$ est un chemin minimal alors $d_C(A,B) = d_C(B,A)$.

Supposons que $\mathcal{P}_2$ ne soit pas un chemin minimal, et donc qu'il existe $\mathcal{P}_3 = (\overrightarrow{v_{j_1}}, \ldots, \overrightarrow{v_{j_n}})$ qui le soit. On peut alors construire le chemin $\mathcal{P}_4 = (-\overrightarrow{v_{j_n}}, \ldots, -\overrightarrow{v_{j_1}})$ de A à B. Or le masque est symétrique, on obtient donc comme précédemment que $W(\mathcal{P}_4) = W(\mathcal{P}_3) < W(\mathcal{P}_2) = W(\mathcal{P}_1)$, ce qui est en contradiction avec le fait que $\mathcal{P}_1$ soit un chemin minimal.

▷ ***Inégalité triangulaire*** : Soit $\mathcal{P}_1$ un chemin minimal de A à B. Si $\mathcal{P}_2 = (\overrightarrow{v_{i_1}}, \ldots, \overrightarrow{v_{i_m}})$ est un chemin minimal de A à C, et que $\mathcal{P}_3 = (\overrightarrow{v_{j_1}}, \ldots, \overrightarrow{v_{j_n}})$ est un chemin minimal de C à B, alors leur concaténation $\mathcal{P}_4 = \mathcal{P}_2\mathcal{P}_3 = (\overrightarrow{v_{i_1}}, \ldots, \overrightarrow{v_{i_m}}, \overrightarrow{v_{j_1}}, \ldots, \overrightarrow{v_{j_n}})$ a pour longueur $d_C(A,C) + d_C(C,B)$.

Supposons que $d_C(A,B) > d_C(A,C) + d_C(C,B)$, alors $W(\mathcal{P}_1) > W(\mathcal{P}_4)$, ce qui contredit que $\mathcal{P}_1$ soit un chemin minimal. □

C'est la découverte de cette preuve de Verwer (n'ayant jamais à notre connaissance été publiée ailleurs que dans sa thèse), qui a remis en question les résultats publiés dans [Thi94] aux §3.4 et §3.5.1, ainsi que dans nos premiers articles [Rem99, Rem00c, Rem00b]. Cette preuve met en lumière que toutes les publications existantes sur le sujet tentent d'établir non pas les conditions pour qu'une distance de chanfrein d_C soit une distance mais plutôt pour qu'elle soit une norme. En effet, on cherche à ce que les valeurs de la fonction soient prévisibles, ce qui est le cas dès que la fonction est une norme. Le paragraphe suivant illustre par un exemple les phénomènes que l'on cherche à éviter.

3.2.2 Un exemple surprenant

Étudions le cas de la distance $d_{3,1}$ en 2D qui est définie par le générateur $\mathcal{M}_C^g = \{\,\big((1,0),3\big),\big((1,1),1\big)\,\}$ représenté figure 3.12 (a).

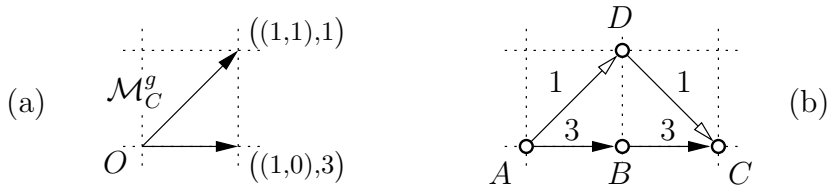


FIG. 3.12. *Cas de $d_{3,1}$: $\mathcal{M}_C^g = \big\{\big((1,0),3\big),\big((1,1),1\big)\big\}$ (a) et 2 chemins de A à C (b).*

Le schéma (b) de la figure 3.12 illustre le calcul de la valeur de distance entre les points A et C, qui est par définition le minimum des coûts des chemins les reliant. Deux de ces

chemins sont représentés : $\mathcal{P}_{ABC}$ et $\mathcal{P}_{ADC}$. On a $W(\mathcal{P}_{ABC}) = 6$ alors que $W(\mathcal{P}_{ADC}) = 2$. En d'autres termes, le chemin le plus court n'est pas la ligne droite. Il est important de se convaincre que cette observation ne remet absolument pas en cause l'inégalité triangulaire ainsi que l'a montré Verwer, puisqu'on a toujours bien $d_{3,1}(A,C) \leqslant d_{3,1}(A,D) + d_{3,1}(D,C)$ et $d_{3,1}(A,C) \leqslant d_{3,1}(A,B) + d_{3,1}(B,C)$.

Cet exemple est bien entendu un cas d'école mais illustre certaines propriété potentiellement indésirables des distances de chanfrein. La figure 3.13 (a) représente la distance à l'origine (case 0 hachurée) de certains points de $\frac{1}{8}\mathbb{Z}^2$. Les points proches de la frontière sont indiqués (en grisé) afin de montrer explicitement les symétries qui permettent d'obtenir l'étiquetage de tout $\mathbb{Z}^2$. On peut observer dans la région décrite que deux sous-réseaux cohabitent. Les points du premier (en gris) sont étiquetés par des chemins constitués uniquement de vecteurs de la famille de a tandis que ceux du deuxième (en blanc) sont étiquetés par des chemins constitués d'un certain nombre de vecteurs de même famille que a et d'un seul b.

On peut aussi observer figure 3.13 (b), que les points inférieurs à 4 (en gris), qui forment par symétrie la boule de rayon 4 dans la figure 3.13 (c), ne forment pas un ensemble convexe, ce qui peut avoir un effet négatif sur des applications comme par exemple la détermination de l'axe médian qui fait l'objet du chapitre 6.

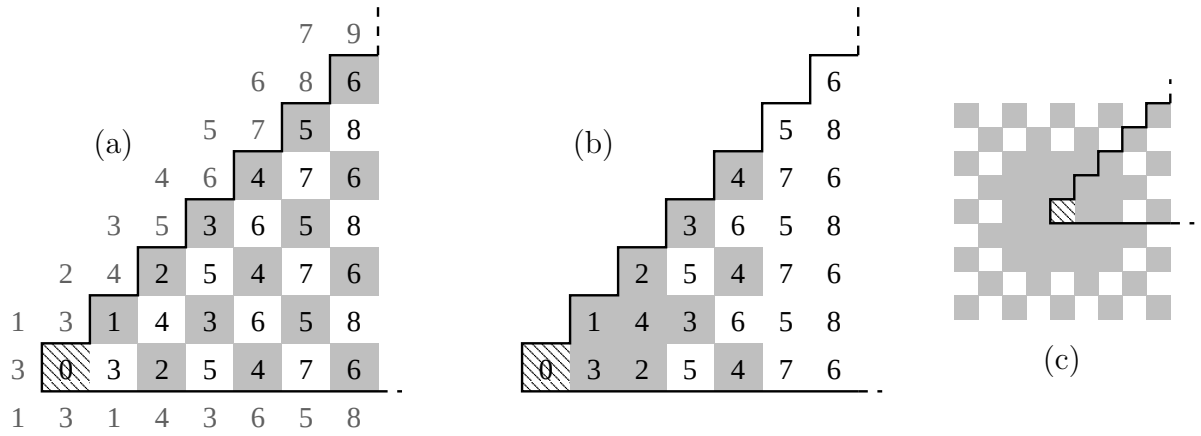


FIG. 3.13. *Cas de $d_{3,1}$: carte de distance (a), points de $\frac{1}{8}\mathbb{Z}^2$ faisant partie de la boule de rayon 4 (b, en gris) et la même boule complète, obtenue par symétrie (c).*

En résumé, il est important de noter que certains masque $\mathcal{M}_C$ peuvent provoquer des effets indésirables, parfois assez subtils et plus difficiles à détecter que dans cet exemple.

3.2.3 Régularité et semi-régularité

C'est sans doute dans le but d'éviter le comportement de l'exemple précédant, que Borgefors introduit dans [Bor91a] les notions de régularité et de semi-régularité.

Définition 8 (régularité) *Soit p et q deux points pouvant être connectés en utilisant autant de fois que nécessaire, une seule pondération du masque. Si le chemin ainsi formé*

définit la distance entre p et q, c'est-à-dire est le chemin minimal, alors la fonction de distance est dite semi-régulière. *Si il n'existe pas d'autre chemin minimal, alors la distance est dite* régulière.

Borgefors montre ainsi que les masques du type $\mathcal{M}_C^g = \{ (\overrightarrow{v_\mathsf{a}},w_\mathsf{a}),(\overrightarrow{v_\mathsf{b}},w_\mathsf{b}) \}$ en 2D correspondent à une distance régulière si et seulement si $w_\mathsf{a} < w_\mathsf{b} < 2.w_\mathsf{a}$ (et semi-régulière si on remplace une au moins des inégalités strictes par une inégalité large). On se convainc aisément que ces conditions permettent d'éviter l'exemple du §3.2.2 : on s'assure en particulier que la pondération $(\overrightarrow{v_\mathsf{b}},w_\mathsf{b})$ n'est plus systématiquement privilégiée par rapport à $(\overrightarrow{v_\mathsf{a}},w_\mathsf{a})$.

Kiselman étudie dans [Kis96] la régularité sous un aspect théorique. Il introduit en particulier un formalisme mathématique pour définir les cartes de distance (cf. §3.1.4). Sa définition sur la base de la convergence d'une série de *convolutions infimales* semble être inspirée par l'algorithme parallèle : on cherche, à partir de chaque point de DT et dans chaque direction de $\mathcal{M}_C$, quelle est la plus petite valeur de distance pouvant être propagée.

Kiselman montre pour tout masque de chanfrein et en toute dimension, que si une distance de chanfrein est homogène, alors elle est semi-régulière au sens de Borgefors, mais il ne prouve malheureusement pas la réciproque. On retient donc qu'il est nécessaire de chercher la semi-régularité, mais qu'elle n'est *a priori* pas suffisante.

Borgefors applique ensuite ces travaux dans [Bor96] pour le cas des masques 3D du type $\mathcal{M}_C^g = \{ (\overrightarrow{v_\mathsf{a}},w_\mathsf{a}),(\overrightarrow{v_\mathsf{b}},w_\mathsf{b}),(\overrightarrow{v_\mathsf{c}},w_\mathsf{c}) \}$ et confirme que $d_{3,4,5}$ est une fonction régulière. Dans [Bor00], l'étude des masques de taille 3 est poussée jusqu'à la 4ème dimension. Borgefors y recommande l'usage de $\mathcal{M}_C^g = \{ \big((1,0,0,0),3\big), \big((1,1,0,0),4\big), \big((1,1,1,0),5\big), \big((1,1,1,1),6\big) \}$ qui est l'extension du masque précédant et qui réalise un bon compromis en minimisant l'erreur commise par rapport à d_E tout en gardant des valeurs faibles sur la carte de distance.

Après l'étude des masques de taille 3 en 2D, 3D et 4D, qui ne soulève pas de problèmes théoriques particuliers et semble confirmer que la semi-régularité est bien équivalente à l'homogénéité, Borgefors et Svensson se lancent dans l'étude plus ardue des masques de taille 5 en 3D dans [Bor01], et plus spécifiquement des masques du type $\mathcal{M}_C^g = \{ (\overrightarrow{v_\mathsf{a}},w_\mathsf{a}), (\overrightarrow{v_\mathsf{b}},w_\mathsf{b}), (\overrightarrow{v_\mathsf{c}},w_\mathsf{c}), (\overrightarrow{v_\mathsf{e}},w_\mathsf{e}) \}$. Mais cette étude soulève à nos yeux un problème et nous conduit dans le prochain paragraphe à développer un contre-exemple.

3.2.4 La semi-régularité n'est pas l'homogénéité

Dans [Bor01], Borgefors et Svensson étudient les masques $\mathcal{M}_C^g = \{ \big((1,0,0), w_\mathsf{a}\big), \big((1,1,0), w_\mathsf{b}\big), \big((1,1,1), w_\mathsf{c}\big), \big((2,1,1), w_\mathsf{e}\big) \}$. Elles montrent que la distance de chanfrein correspondante est semi-régulière si

$$w_\mathsf{e} \leqslant w_\mathsf{a} + w_\mathsf{c} \ , \ \ w_\mathsf{e} \leqslant 2.w_\mathsf{b} \ , \ \ w_\mathsf{b} \leqslant w_\mathsf{c} \ , \ \ 3.w_\mathsf{b} \leqslant 2.w_\mathsf{e} \ \text{ et } \ 4.w_\mathsf{c} \leqslant 3.w_\mathsf{e} \ . \tag{3.2}$$

Borgefors et Svensson isolent ensuite une série de huit sous-cas différents qui tous correspondent à la semi-régularité de la distance. Nous nous intéressons au cas numéro II de leur article qui correspond aux conditions supplémentaires suivantes :

$$w_\mathsf{e} \leqslant w_\mathsf{a} + w_\mathsf{b} \ , \ \ \frac{1}{2}.w_\mathsf{b} + w_\mathsf{c} \leqslant w_\mathsf{e} \ \text{ et } \ w_\mathsf{a} + 2.w_\mathsf{b} - w_\mathsf{c} \leqslant w_\mathsf{e} \ . \tag{3.3}$$

On remarque tout d'abord que ces contraintes sont réalisables puisqu'on peut par exemple choisir $w_{\mathsf{a}} = 20$, $w_{\mathsf{b}} = 28$, $w_{\mathsf{c}} = 31$ et $w_{\mathsf{e}} = 47$.

Supposons qu'un tel masque $\mathcal{M}_C$ satisfaisant (3.2) et (3.3), corresponde à une norme de chanfrein n_C si et seulement si la distance d_C correspondante est semi-régulière.

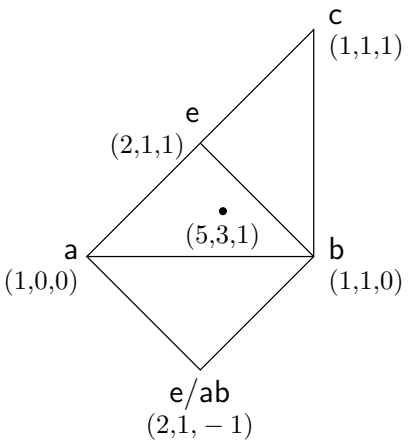


FIG. 3.14. *Contre-exemple : la régularité n'implique pas l'homogénéité. Positions relatives des points du Masque, et des points* (10,6,2) *et* (5,3,1) *(confondus sur cette représentation en projection).*

Si on calcule les distances à l'origine des points de $\frac{1}{48}\mathbb{Z}^3$ de la même façon qu'à la figure 3.13, et qu'on étudie les valeurs des points (10,6,2) et (5,3,1) qui sont représentés à la figure 3.14, on peut vérifier que pour $n_{20,28,31,\mathsf{e}=47}$, on trouve $DT[(10,6,2)] = 244 = 2.w_{\mathsf{b}} + 4.w_{\mathsf{e}}$ car le chemin $2.\overrightarrow{v_{\mathsf{b}}} + 3.\overrightarrow{v_{\mathsf{e}}} + \overrightarrow{v_{\mathsf{e/ab}}} = 2.(1,1,0) + 3.(2,1,1) + (2,1,-1)$ est minimal ; mais si on cherche la valeur de $n_C(5,3,1)$, on trouve que $DT[(5,3,1)] = 123 = w_{\mathsf{a}} + 2.w_{\mathsf{b}} + w_{\mathsf{e}}$ car $\overrightarrow{v_{\mathsf{a}}} + 2.\overrightarrow{v_{\mathsf{b}}} + \overrightarrow{v_{\mathsf{e}}} = (1,0,0) + 2.(1,1,0) + (2,1,1)$ est minimal, or si n_C était homogène, on devrait avoir $n_C(10,6,2) = 2.n_C(5,3,1)$, et donc :

Théorème 2 *Une distance de chanfrein régulière ou semi-régulière (au sens de Borgefors) n'est pas forcement homogène et donc ne vérifie pas nécessairement les propriétés d'une norme de chanfrein.*

Dans le chapitre suivant, nous détaillerons notre approche qui, bien que d'origine différente, est par beaucoup d'aspects comparable à celle de Borgefors. Nous montrerons que nous apportons une solution simple au problème consistant à obtenir un masque qui donne une *norme de chanfrein.*

3.3 Conclusion

Dans un premier temps, nous avons défini les distances de chanfrein et présenté l'algorithme qui permet de calculer la transformée de distance d'une forme à partir d'une image

discrète. Nous avons défini le générateur $\mathcal{M}_C^g$ qui permet de retrouver, par symétrie, le masque de chanfrein $\mathcal{M}_C$. Nous avons présenté les notions de boule et de boule inverse et avons montré le lien qui les unit. Nous avons montré que les directions de pondérations $\overrightarrow{v_i}$ doivent être choisies parmi l'ensemble des points visibles.

Dans un deuxième temps, nous avons repris la démonstration de Verwer qui montre que toute distance de chanfrein vérifie bien les propriétés d'une distance discrète. Nous avons présenté la régularité au sens de Borgefors et Kiselman, et avons montré que dans un cadre général, elle n'est pas suffisante pour qu'on obtienne les propriétés d'une norme discrète.

Chapitre 4

Normes de chanfrein

4.1 Introduction

Ce chapitre présente notre méthode pour déterminer des normes de chanfrein discrètes.

Nous y étudions tout d'abord les normes de chanfrein dans l'espace continu et en tirons des propriétés qui permettent de mieux comprendre leurs comportements.

Ensuite, sur la base de ces propriétés, nous proposons une technique qui permet de construire un masque de chanfrein donnant une norme, de telle façon que sa valeur en tout point soit connue, sans passer par l'algorithme de transformation de distance de Rosenfeld.

Enfin, nous montrons comment replonger ces résultats continus dans l'espace discret, sans perdre les propriétés obtenues, afin d'obtenir des normes de chanfrein discrètes.

4.2 Norme de chanfrein dans l'espace continu

Définition 9 (distance de chanfrein continue) *Pour un masque de chanfrein $\mathcal{M}_C$ donné, on redéfinit la distance de chanfrein continue, notée $d_C^{\mathbb{R}}$, comme étant la fonction de $\mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^+$ qui s'exprime par*

$$d_C^{\mathbb{R}}(P,Q) = \min\left\{ \sum_{(\overrightarrow{v_i},w_i)\in\mathcal{M}_C} \lambda_i.w_i \ : \ \sum_{(\overrightarrow{v_i},w_i)\in\mathcal{M}_C} \lambda_i.\overrightarrow{v_i} = \overrightarrow{PQ} \ \text{ et } \ \forall i, \lambda_i \in \mathbb{R}^+ \right\}. \tag{4.1}$$

Cette définition est l'analogue continu de la définition discrète classique, (définition 4 vue au §3.1.3), et qui se résume par

$$d_C(P,Q) = \min\left\{ \sum_{(\overrightarrow{v_i},w_i)\in\mathcal{M}_C} n_i.w_i \ : \ \sum_{(\overrightarrow{v_i},w_i)\in\mathcal{M}_C} n_i.\overrightarrow{v_i} = \overrightarrow{PQ} \ \text{ et } \ \forall i, n_i \in \mathbb{N} \right\}. \tag{4.2}$$

Soit P et Q deux points de $\mathbb{E}$, on définit les normes de chanfrein associées : $n_C^{\mathbb{R}}(\overrightarrow{PQ}) = d_C^{\mathbb{R}}(P,Q)$ et $n_C(\overrightarrow{PQ}) = d_C(P,Q)$.

Une propriété surprenante de $n_C^{\mathbb{R}}$ quand on a observé le comportement de n_C sur les contre-exemples du §3.2.2 et du §3.2.4 est la suivante :

Théorème 3 *Pour tout masque de chanfrein $\mathcal{M}_C$, la fonction $n_C^{\mathbb{R}}$ est une norme sur $\mathbb{R}^n$.*

Preuve : On a vu au §3.2.1 la preuve de Verwer que d_C vérifie l'inégalité triangulaire. Sa démonstration tient toujours pour $n_C^{\mathbb{R}}$. Il ne reste donc qu'à prouver que pour tout λ réel, on a $n_C^{\mathbb{R}}(\lambda.\overrightarrow{v}) = |\lambda|.n_C^{\mathbb{R}}(\overrightarrow{v})$ afin de vérifier les propriétés énoncées au §2.3.4.

On remarquera tout d'abord que si $\lambda = 0$, alors pour tout vecteur $\overrightarrow{v}$, on a

$$n_C^{\mathbb{R}}(\lambda.\overrightarrow{v}) = n_C^{\mathbb{R}}(\overrightarrow{0}) = 0 = |\lambda|.n_C^{\mathbb{R}}(\overrightarrow{v})\,.$$

On suppose maintenant que $\lambda > 0$. D'après la définition, on a

$$n_C^{\mathbb{R}}(\lambda.\overrightarrow{PQ}) = \min\left\{ \sum_{(\overrightarrow{v_i},w_i)\in\mathcal{M}_C} \lambda_i.w_i \ : \sum_{(\overrightarrow{v_i},w_i)\in\mathcal{M}_C} \lambda_i.\overrightarrow{v_i} = \lambda.\overrightarrow{PQ} \ \text{ et } \ \forall i, \lambda_i \geqslant 0 \right\},$$

puis, comme $\lambda \neq 0$, on pose $\lambda'_i = \lambda_i/\lambda$, et on substitue λ_i par $\lambda.\lambda'_i$. On remarque aussi que $\lambda_i \geqslant 0$ et $\lambda > 0$ implique $\lambda'_i \geqslant 0$. On a

$$n_C^{\mathbb{R}}(\lambda.\overrightarrow{PQ}) = \min\left\{ \sum_{(\overrightarrow{v_i},w_i)\in\mathcal{M}_C} \lambda.\lambda'_i.w_i \ : \sum_{(\overrightarrow{v_i},w_i)\in\mathcal{M}_C} \lambda.\lambda'_i.\overrightarrow{v_i} = \lambda.\overrightarrow{PQ} \ \text{ et } \ \forall i, \lambda'_i \geqslant 0 \right\}.$$

On factorise les λ, pour obtenir

$$n_C^{\mathbb{R}}(\lambda.\overrightarrow{PQ}) = \min\left\{ \lambda.\sum_{(\overrightarrow{v_i},w_i)\in\mathcal{M}_C} \lambda'_i.w_i \ : \lambda.\sum_{(\overrightarrow{v_i},w_i)\in\mathcal{M}_C} \lambda'_i.\overrightarrow{v_i} = \lambda.\overrightarrow{PQ} \ \text{ et } \ \forall i, \lambda'_i \geqslant 0 \right\}.$$

En mettant λ en facteur, et en simplifiant l'équation vectorielle, on a

$$n_C^{\mathbb{R}}(\lambda.\overrightarrow{PQ}) = \lambda.\min\left\{ \sum_{(\overrightarrow{v_i},w_i)\in\mathcal{M}_C} \lambda'_i.w_i \ : \sum_{(\overrightarrow{v_i},w_i)\in\mathcal{M}_C} \lambda'_i.\overrightarrow{v_i} = \overrightarrow{PQ} \ \text{ et } \ \forall i, \lambda'_i \geqslant 0 \right\}.$$

D'après la définition de $n_C^{\mathbb{R}}$ et puisque $\lambda > 0$, on a $n_C^{\mathbb{R}}(\lambda.\overrightarrow{PQ}) = |\lambda|.n_C^{\mathbb{R}}(\overrightarrow{PQ})$.

Il reste à traiter le cas où $\lambda < 0$. On sait que $n_C^{\mathbb{R}}$ est symétrique et donc que $n_C^{\mathbb{R}}(\lambda.\overrightarrow{v}) = n_C^{\mathbb{R}}(-\lambda.\overrightarrow{v})$ où $-\lambda > 0$. D'après ce qui précède, on a donc $n_C^{\mathbb{R}}(\lambda.\overrightarrow{v}) = |-\lambda|.n_C^{\mathbb{R}}(\overrightarrow{v}) = |\lambda|.n_C^{\mathbb{R}}(\overrightarrow{v})$ ce qui conclut la démonstration. □

Ce qui différencie le comportement de n_C et de $n_C^{\mathbb{R}}$, c'est l'impossibilité dans le premier cas d'utiliser les λ_i réels dans (4.1) à la place des n_i entiers dans (4.2). En d'autres termes, dans le domaine discret, le minimum calculé dans le domaine continu ne peut pas toujours être atteint car il n'existe pas de combinaison linéaire entière (avec les n_i) permettant d'étiqueter le point. En conséquence, le point est étiqueté par une autre combinaison linéaire entière de valeur égale ou supérieure.

Sur l'exemple du §3.2.2, on a $n_C(\overrightarrow{OB}) = w_{\mathsf{a}} = 3$ car $\overrightarrow{v_{\mathsf{a}}}$ est le chemin discret de coût minimal de O à B, alors qu'on a $n_C^{\mathbb{R}}(\overrightarrow{OB}) = w_{\mathsf{b}} = 1 = (w_{\mathsf{b}} + w_{\mathsf{b/ab}})/2$ puisque $\overrightarrow{OB} = (\overrightarrow{v_{\mathsf{a}}} + \overrightarrow{v_{\mathsf{a/ab}}})/2$ est alors minimal.

4.2.1 Masque rationnel équivalent

Nous allons maintenant étudier le comportement de la norme de chanfrein continue $n_C^{\mathbb{R}}$. Cette étude mettra en lumière la nécessité de définir les notions de masque de chanfrein rationnel équivalent et de cône.

Définition 10 (Masque rationnel équivalent) *Pour un masque de chanfrein $\mathcal{M}_C$ donné, on définit le masque rationnel équivalent par*

$$\mathcal{M}'_C = \left\{ \left(\frac{x_i}{w_i}, \frac{y_i}{w_i}, \frac{z_i}{w_i}\right) \in \mathbb{Q}^3 \ : \ (x_i, y_i, z_i, w_i) \in \mathcal{M}_C \right\}.$$

Dans la suite, on notera souvent $\frac{\overrightarrow{v_i}}{w_i}$ la pondération de $\mathcal{M}'_C$ correspondant à $(\overrightarrow{v_i},w_i)$ de $\mathcal{M}_C$.

En effectuant une mise à l'échelle de chaque pondération $(\overrightarrow{v_i},w_i) \in \mathcal{M}_C$ par sa valeur de distance locale w_i, on obtient des pondérations $\left(\frac{\overrightarrow{v_i}}{w_i},\frac{w_i}{w_i}\right) = \left(\frac{\overrightarrow{v_i}}{w_i},1\right)$. Les points du masque rationnel équivalent sont donc tous à une distance locale unitaire vis-à-vis de leur propre direction. On peut ainsi étudier le masque de chanfrein rationnel équivalent sous un aspect purement géométrique (dans l'espace projectif $w_i = 1$).

4.2.2 Cône continu

En 2D, on appelle *cône* une région comprise entre 2 vecteurs de $\mathcal{M}_C$ (ou de $\mathcal{M}'_C$ suivant le contexte). De même, un cône en 3D est défini par 3 vecteurs.

Définition 11 (Cône continu) *On appelle cône continu, noté $<i,j>$ (resp. $<i,j,k>$), la zone de $\mathbb{R}^2$ (resp. $\mathbb{R}^3$) délimitée par les vecteurs $\overrightarrow{v_i}$ et $\overrightarrow{v_j}$ (resp. $\overrightarrow{v_i}$, $\overrightarrow{v_j}$ et $\overrightarrow{v_k}$) :*

$$<i,j> = \left\{ M \in \mathbb{R}^2 \ : \ \overrightarrow{OM} = \lambda_i.\overrightarrow{v_i} + \lambda_j.\overrightarrow{v_j} \text{ avec } \lambda_i,\lambda_j \in \mathbb{R}^+ \right\} \quad \text{et}$$

$$<i,j,k> = \left\{ M \in \mathbb{R}^3 \ : \ \overrightarrow{OM} = \lambda_i.\overrightarrow{v_i} + \lambda_j.\overrightarrow{v_j} + \lambda_k.\overrightarrow{v_k} \text{ avec } \lambda_i,\lambda_j,\lambda_k \in \mathbb{R}^+ \right\}.$$

4.2.3 Déterminant d'un cône

En 3D, on définit le déterminant d'un cône de la manière suivante :

Définition 12 (Déterminant d'un cône) *Soit $\mathcal{M}_C$ un masque de chanfrein 3D, soit $<i,j,k>$ un cône et soit $(\overrightarrow{v_i},w_i)$, $(\overrightarrow{v_j},w_j)$ et $(\overrightarrow{v_k},w_k)$ les pondérations correspondantes de $\mathcal{M}_C$. Le déterminant de $<i,j,k>$ vaut*

$$\Delta_{i,j,k} = \begin{vmatrix} x_i & x_j & x_k \\ y_i & y_j & y_k \\ z_i & z_j & z_k \end{vmatrix}.$$

La définition 2D est analogue : pour un cône $<i,j>$, on a $\Delta_{i,j} = \begin{vmatrix} x_i & x_j \\ y_i & y_j \end{vmatrix}$.

On rappelle que le déterminant de deux vecteurs (resp. de trois vecteurs) est l'aire du parallélogramme (resp. le volume du parallélépipède) défini par ces vecteurs. On remarque que $\Delta_{i,j,k}$ ne dépend pas de w_i, w_j et w_k.

4.2.4 Propriétés de $n_C^{\mathbb{R}}$ et enveloppe convexe des points de $\mathcal{M}'_C$

Le comportement de $n_C^{\mathbb{R}}$ est lié à la position relative des points de $\mathcal{M}'_C$, et plus particulièrement à leur position par rapport à leur enveloppe convexe.

Étant donné un masque de chanfrein $\mathcal{M}_C$, soit $\mathcal{M}'_C$ son masque rationnel équivalent. On note $\circledcirc\mathcal{M}'_C$ l'enveloppe convexe des points de $\mathcal{M}'_C$. Les trois théorèmes suivants apportent une vision de plus en plus précise d'un même phénomène.

Théorème 4 *Seules participent au calcul de $n_C^{\mathbb{R}}$ les pondérations $(\overrightarrow{v_i},w_i) \in \mathcal{M}_C$ telles que $\frac{\overrightarrow{v_i}}{w_i} \in \mathcal{M}'_C$ soit sur la frontière de $\circledcirc\mathcal{M}'_C$, c'est-à-dire telles que $\frac{\overrightarrow{v_i}}{w_i}$ ne soit pas un point intérieur de $\circledcirc\mathcal{M}'_C$.*

Théorème 5 *Le contour de $\circledcirc\mathcal{M}'_C$ en 2D est un polygone, et chaque arête $\left(\frac{\overrightarrow{v_i}}{w_i},\frac{\overrightarrow{v_j}}{w_j}\right)$ correspond à un cône $<i,j>$ où la valeur de $n_C^{\mathbb{R}}$ ne dépend que des pondérations $(\overrightarrow{v_i},w_i)$, et $(\overrightarrow{v_j},w_j)$. De même en 3D, la surface de $\circledcirc\mathcal{M}'_C$ est formée de triangles $\left(\frac{\overrightarrow{v_i}}{w_i},\frac{\overrightarrow{v_j}}{w_j},\frac{\overrightarrow{v_k}}{w_k}\right)$ et chacun d'eux correspond à un cône $<i,j,k>$ où la valeur de $n_C^{\mathbb{R}}$ ne dépend que de ces trois pondérations.*

Théorème 6 *La fonction $n_C^{\mathbb{R}}$ est une application linéaire par morceau où chaque morceau est un cône qui correspond à une facette de $\circledcirc\mathcal{M}'_C$.*

Preuve :

La preuve qui suit est donnée pour le cas 2D mais reste valide en 3D ; ce n'est que pour faciliter la lecture des figures illustrant la preuve qu'on s'imposera cette limitation.

On considère tout d'abord un masque de chanfrein $\mathcal{M}_C$ ne contenant qu'une pondération $(\overrightarrow{v_i},w_i)$ et son symétrique $(-\overrightarrow{v_i},w_i)$. Considérons les points M tels que $\overrightarrow{OM} = \lambda_i.\overrightarrow{v_i}$ avec $\lambda_i \in \mathbb{R}$; ils forment la droite D_i comme illustré figure 4.1. On nomme D_i^+ la demi-droite (privée de O) formée des points tels que $\lambda_i > 0$, et D_i^- la demi-droite symétrique ($\lambda_i < 0$).

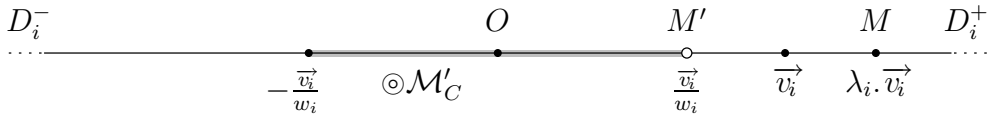


FIG. 4.1. *Masque $\mathcal{M}'_C$ à deux vecteurs.*

Si $M \in D_i^+$, nous pouvons déterminer la valeur de la norme de chanfrein continue associée à $\overrightarrow{OM} = \lambda_i.\overrightarrow{v_i}$ en utilisant la définition 9. On trouve $n_C^{\mathbb{R}}(\overrightarrow{OM}) = \lambda_i.w_i$ car $\lambda_i.\overrightarrow{v_i}$ est la seule combinaison linéaire positive possible des vecteurs de $\mathcal{M}_C$ pour atteindre M ; elle réalise donc nécessairement le minimum. De même, si $M \in D_i^-$ alors la seule combinaison linéaire positive des vecteurs de $\mathcal{M}_C$ est $\overrightarrow{OM} = \lambda_i.(-\overrightarrow{v_i})$ et la valeur de $n_C^{\mathbb{R}}$ reste identique.

On définit la fonction f qui à tout point M de $\mathbb{R}^2 \setminus \{O\}$ associe le point M' tel que

$$\overrightarrow{OM'} = \frac{\overrightarrow{OM}}{n_C^{\mathbb{R}}(\overrightarrow{OM})} \ .$$

Or d'après ce qui précède, pour tout $M \in D_i^+$, on a M' constant et tel que

$$\overrightarrow{OM'} = \frac{\lambda_i.\overrightarrow{v_i}}{\lambda_i.w_i} = \frac{\overrightarrow{v_i}}{w_i} \ .$$

Symétriquement, on a pour tout $M \in D_i^-$, un point M' constant et tel que $\overrightarrow{OM'} = -\frac{\overrightarrow{v_i}}{w_i}$.

Supposons maintenant que $\mathcal{M}_C$ contienne d'autres pondérations. Par définition, $n_C^{\mathbb{R}}$ ne pourrait être qu'inférieure ou égale à $\lambda_i.w_i$ puisque l'ajout de pondération ne pourrait que faire baisser la valeur du minimum ; ce qui implique que M' soit plus loin de O que $\frac{\overrightarrow{v_i}}{w_i}$ et

$-\frac{\overrightarrow{v_i}}{w_i}$, c'est-à-dire ne soit pas dans la partie grisée de la droite qui est l'enveloppe convexe des deux vecteurs de $\mathcal{M}'_C$.

Introduisons maintenant une deuxième pondération $(\overrightarrow{v_j},w_j)$ dans $\mathcal{M}_C$. Puisque nous ne choisissons que des directions suivant les points visibles (cf. §3.1.6), les vecteurs $\overrightarrow{v_i}$ et $\overrightarrow{v_j}$ sont linéairement indépendants et donc $\mathcal{M}'_C$ forme ce qu'on choisit d'appeler une *base à coefficients positifs* de $\mathbb{R}^2$: tout vecteur de $\mathbb{R}^2$ s'exprime comme une combinaison linéaire unique à coefficients positifs de vecteurs de $\mathcal{M}'_C$ telle que n'y interviennent pas simultanément deux vecteurs symétriques l'un de l'autre.

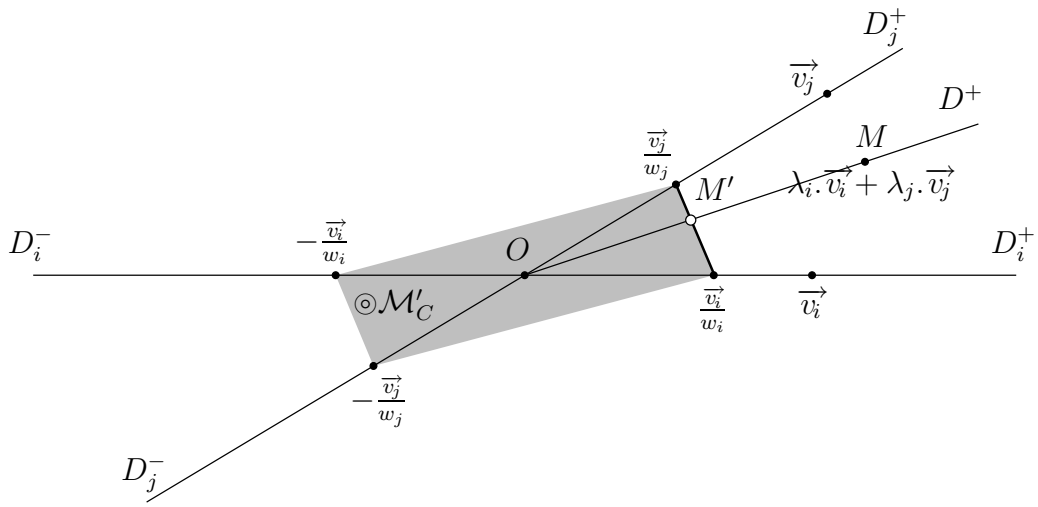


FIG. 4.2. *Masque $\mathcal{M}'_C$ à quatre vecteurs.*

Le même raisonnement que précédemment permet de savoir que $\overrightarrow{v_j}$ définit la droite D_j et que tout point M sur D_j^+ a pour image par f un point M' tel que $\overrightarrow{OM'} = \frac{\lambda_j.\overrightarrow{v_j}}{\lambda_j.w_j} = \frac{\overrightarrow{v_j}}{w_j}$ car seul $\overrightarrow{v_j}$ permet d'atteindre M. De même, symétriquement, on a pour tout point M de D_j^- un point image M' tel que $\overrightarrow{OM'} = -\frac{\overrightarrow{v_j}}{w_j}$. De façon plus générale, choisissons un point M quelconque de $\mathbb{R}^2$; il est de la forme $\overrightarrow{OM} = \lambda_i.\overrightarrow{v_i} + \lambda_j.\overrightarrow{v_j}$ où $\lambda_i,\lambda_j \in \mathbb{R}^+$ et $\overrightarrow{v_i}$, $\overrightarrow{v_j}$ sont deux vecteurs de $\mathcal{M}_C$, c'est-à-dire M est dans le cône $<i,j>$. Comme il n'existe pas d'autre combinaison linéaire positive possible des vecteurs de $\mathcal{M}_C$ pour exprimer $\overrightarrow{OM}$, alors $n_C^{\mathbb{R}}(\overrightarrow{OM}) = \lambda_i.w_i + \lambda_j.w_j$. En conséquence, le point M', image de M par f est tel que

$$\overrightarrow{OM'} = \frac{\lambda_i.\overrightarrow{v_i} + \lambda_j.\overrightarrow{v_j}}{\lambda_i.w_i + \lambda_j.w_j} \quad \text{où} \quad \lambda_i,\, \lambda_j \in \mathbb{R}^+ \ . \tag{4.3}$$

Or on a

$$\frac{\lambda_i.\overrightarrow{v_i} + \lambda_j.\overrightarrow{v_j}}{\lambda_i.w_i + \lambda_j.w_j} = \frac{\lambda_i.w_i}{\lambda_i.w_i + \lambda_j.w_j} \cdot \frac{\overrightarrow{v_i}}{w_i} + \frac{\lambda_j.w_j}{\lambda_i.w_i + \lambda_j.w_j} \cdot \frac{\overrightarrow{v_j}}{w_j} \,,$$

et si on pose

$$\lambda'_i = \frac{\lambda_i.w_i}{\lambda_i.w_i + \lambda_j.w_j} \quad \text{et} \quad \lambda'_j = \frac{\lambda_j.w_j}{\lambda_i.w_i + \lambda_j.w_j}$$

alors on remarque que l'expression (4.3) est de la forme

$$\overrightarrow{OM'} = \lambda'_i \cdot \frac{\overrightarrow{v_i}}{w_i} + \lambda'_j \cdot \frac{\overrightarrow{v_j}}{w_j} \quad \text{où} \quad \lambda'_i + \lambda'_j = 1 \quad \text{et} \quad \lambda'_\mathrm{i}, \lambda'_\mathrm{j} \in \mathbb{R}^+ \ ,$$

ce qui signifie que M' est une combinaison convexe des points $\frac{\overrightarrow{v_i}}{w_i}$ et $\frac{\overrightarrow{v_j}}{w_j}$. Or l'ensemble des combinaisons convexes est l'enveloppe convexe, ici le segment $\left(\frac{\overrightarrow{v_i}}{w_i}, \frac{\overrightarrow{v_j}}{w_j}\right)$ comme on peut le voir sur la figure 4.2. Si on fait le même raisonnement pour les points des trois autres cônes, on trouve que pour tout point M de $\mathbb{R}^2$, son image M' est nécessairement sur le contour de l'enveloppe convexe $\circledcirc\mathcal{M}'_C$ (le rectangle gris).

Comme précédemment, remarquons que si $\mathcal{M}_C$ comportait plus de pondérations, alors la valeur de $n_C^{\mathbb{R}}$ ne pourrait être que plus petite et donc qu'un point M quelconque ne pourrait pas avoir d'image dans le rectangle gris puisque cela signifierait une valeur supérieure pour $n_C^{\mathbb{R}}$.

Rajoutons maintenant une troisième pondération $(\overrightarrow{v_i}, w_i)$ (et son symétrique) à $\mathcal{M}_C$. Le masque $\mathcal{M}'_C$ n'est plus une base à coefficients positifs de $\mathbb{R}^2$: il y a deux vecteurs de trop. Pour tout point M de $\mathbb{R}^2$, on peut calculer la norme par la définition 9, et on trouve $n_C^{\mathbb{R}}(\overrightarrow{OM}) = \lambda_i.\overrightarrow{v_i} + \lambda_j.\overrightarrow{v_j} + \lambda_k.\overrightarrow{v_k}$ où $\lambda_i, \lambda_j, \lambda_k \in \mathbb{R}^+$ et $\overrightarrow{v_i}$, $\overrightarrow{v_j}$, $\overrightarrow{v_k}$ sont les vecteurs qui permettent d'atteindre le minimum. En effet, il n'est possible d'utiliser qu'au plus trois vecteurs en même temps : si on utilisait un vecteur de plus, on utiliserait forcément un symétrique de ceux déjà utilisés et donc on pourrait simplifier l'expression. Comme précédemment, on peut poser

$$\lambda'_i = \frac{\lambda_i.w_i}{\lambda_i.w_i + \lambda_j.w_j + \lambda_k.w_k} \ , \ \lambda'_j = \frac{\lambda_j.w_j}{\lambda_i.w_i + \lambda_j.w_j + \lambda_k.w_k} \ \text{et} \ \lambda'_k = \frac{\lambda_k.w_k}{\lambda_i.w_i + \lambda_j.w_j + \lambda_k.w_k} \ ,$$

d'où

$$\overrightarrow{OM'} = \lambda'_i \cdot \frac{\overrightarrow{v_i}}{w_i} + \lambda'_j \cdot \frac{\overrightarrow{v_j}}{w_j} + \lambda'_k \cdot \frac{\overrightarrow{v_k}}{w_k} \quad \text{où} \quad \lambda'_i + \lambda'_j + \lambda'_k = 1 \quad \text{et} \quad \lambda'_\mathrm{i}, \lambda'_\mathrm{j}, \lambda'_\mathrm{k} \in \mathbb{R}^+ \ ,$$

ce qui signifie que M' se trouve dans l'enveloppe convexe des points $\frac{\overrightarrow{v_i}}{w_i}$, $\frac{\overrightarrow{v_j}}{w_j}$ et $\frac{\overrightarrow{v_k}}{w_k}$: le triangle $\left(\frac{\overrightarrow{v_i}}{w_i}, \frac{\overrightarrow{v_j}}{w_j}, \frac{\overrightarrow{v_k}}{w_k}\right)$ (cf. figure 4.3). Or M' est sur la demi-droite D^+ qui passe par O et contient M, donc M' est un point de $\left(\frac{\overrightarrow{v_i}}{w_i}, \frac{\overrightarrow{v_j}}{w_j}, \frac{\overrightarrow{v_k}}{w_k}\right) \cap D^+$ (représenté par le segment en gras). De plus, c'est le point le plus loin de O car sinon $n_C^{\mathbb{R}}$ ne serait pas minimale ; c'est donc encore une fois un point de la frontière de $\circledcirc\mathcal{M}'_C$.

Or, si M' est sur une facette (ici un segment du contour) de $\circledcirc\mathcal{M}'_C$, cela signifie que la position de M' s'exprime comme une combinaison linéaire de seulement deux vecteurs de $\mathcal{M}'_C$. Sur la figure 4.3 (a), la position de M' est donnée par $\frac{\overrightarrow{v_i}}{w_i}$ et $\frac{\overrightarrow{v_k}}{w_k}$, et sur la figure 4.3 (b), elle dépend de $\frac{\overrightarrow{v_i}}{w_i}$ et $\frac{\overrightarrow{v_j}}{w_j}$. Cela signifie donc que le minimum atteint en considérant trois pondérations, était déjà obtenu avec deux seulement de ces trois pondérations.

Avant de conclure, attardons nous sur le résultat obtenu pour les masques à six vecteurs. Remarquons tout d'abord que, quel que soit le cas de figure considéré (cas (a) ou (b) sur la figure 4.3), M' est toujours un point du bord de $\circledcirc\mathcal{M}'_C$, et que sa position, sur une

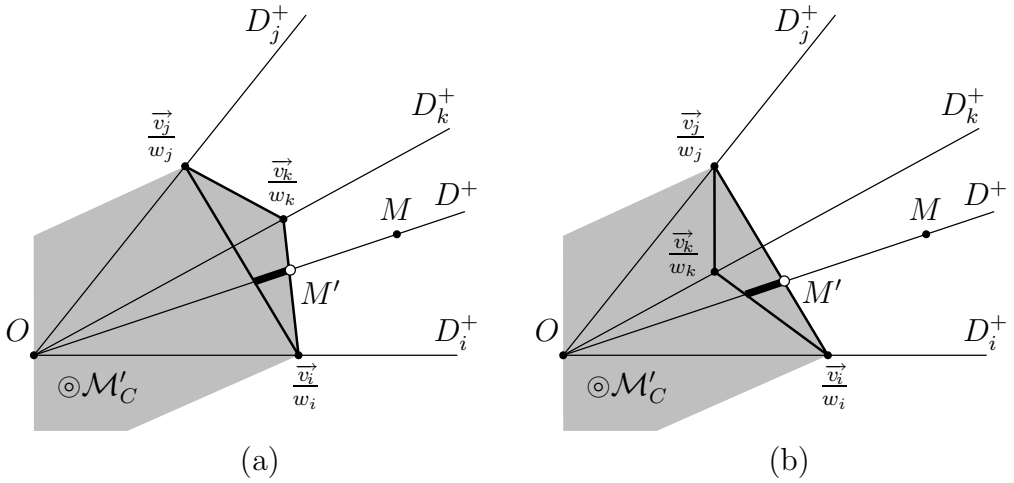


FIG. 4.3. *Masque $\mathcal{M}'_C$ à six vecteurs.*

arête, indique que seules deux pondérations interviennent véritablement dans le calcul de la norme.

Remarquons ensuite que dans le cas (b), jamais la pondération $(\overrightarrow{v_k},w_k)$ n'interviendra puisque $(\overrightarrow{v_i},w_i)$ et $(\overrightarrow{v_j},w_j)$ donnent toujours le minimum. En effet, tous les points M des cônes $<i,k>$ et $<k,j>$ donnent par f un point M' qui ne dépend que de $\frac{\overrightarrow{v_i}}{w_i}$ et $\frac{\overrightarrow{v_j}}{w_j}$.

Remarquons enfin que dans tout cône correspondant à un segment du contour de $\circledcirc\mathcal{M}'_C$, comme par exemple le cône $<i,k>$ dans le cas (a) et le cône $<i,j>$ dans le cas (b), la fonction $n_C^{\mathbb{R}}$ est une application linéaire, car la position de M' est linéaire par rapport aux deux vecteurs, et parce que la valeur de $n_C^{\mathbb{R}}(\overrightarrow{OM})$ dépend linéairement du rapport entre les positions de M et M' sur la demi-droite D^+.

En conclusion, la même démarche constructive permet de prouver les trois théorèmes sur n'importe quel masque de chanfrein (à plus de six pondérations), en ajoutant les pondérations une par une, avec leurs symétriques.

On peut aussi réaliser la même preuve en n'appliquant qu'une seule fois le procédé, en considérant d'entrée la totalité du masque $\mathcal{M}_C$ (sans distinguer les pondérations de leurs symétriques) et en exprimant que M' se trouve dans $\circledcirc\mathcal{M}'_C$, mais qu'il est le point le plus éloigné possible de O sur sa demi-droite porteuse D^+, c'est-à-dire que M' est sur le bord de $\circledcirc\mathcal{M}'_C$. $\square$

On dit qu'un point M *dépend* d'un cône $<i,j>$ (resp. $<i,j,k>$ en 3D) si M est un point de ce cône et si ce cône correspond à une facette de $\circledcirc\mathcal{M}'_C$, c'est-à-dire si le calcul de $n_C^{\mathbb{R}}(\overrightarrow{OM})$ ne dépend que des pondérations $(\overrightarrow{v_i},w_i)$ et $(\overrightarrow{v_j},w_j)$ (resp. $(\overrightarrow{v_i},w_i)$, $(\overrightarrow{v_j},w_j)$ et $(\overrightarrow{v_k},w_k)$).

4.2.5 Formule directe de $n_C^{\mathbb{R}}$

L'idéal serait d'avoir une formule qui permette à partir des coordonnées d'un vecteur de savoir directement la valeur de sa norme. Cette formule existe par exemple pour la norme

euclidienne : $\forall(x,y,z) \in \mathbb{R}^3$, $n_E\big((x,y,z)\big) = \sqrt{x^2+y^2+z^2}$. Malheureusement, il n'est pas possible uniquement à partir de la définition de $n_C^{\mathbb{R}}$ d'obtenir une telle formule. Cependant, nous avons vu (cf. théorème 6) que $n_C^{\mathbb{R}}$ est une application linéaire par morceaux : connaissant le cône dont dépend $\overrightarrow{v}$, c'est-à-dire les deux vecteurs en 2D (resp. les trois vecteurs en 3D) de $\mathcal{M}'_C$ qui permettent d'exprimer $\overrightarrow{v}$ avec la valeur minimale, il est alors possible de donner la valeur de la norme directement à partir des coordonnées de $\overrightarrow{v}$ en opérant un changement de base.

Théorème 7 (Formule directe) *Soit $\overrightarrow{OM}$ un vecteur de coordonnées (x,y) de $\mathbb{R}^2$ (resp. (x,y,z) de $\mathbb{R}^3$). Soit $<i,j>$ (resp. $<i,j,k>$) le cône dont dépend M. On a*

$$n_C^{\mathbb{R}}(\overrightarrow{OM}) = \frac{-1}{\Delta_{i,j}} \cdot \begin{vmatrix} x_i & x_j & x \\ y_i & y_j & y \\ w_i & w_j & 0 \end{vmatrix} \quad \textit{dans } \mathbb{R}^2, \text{ et}$$

$$n_C^{\mathbb{R}}(\overrightarrow{OM}) = \frac{-1}{\Delta_{i,j,k}} \cdot \begin{vmatrix} x_i & x_j & x_k & x \\ y_i & y_j & y_k & y \\ z_i & z_j & z_k & z \\ w_i & w_j & w_k & 0 \end{vmatrix} \quad \textit{dans } \mathbb{R}^3.$$

Preuve :

Nous ne donnerons que la preuve concernant le cas 3D ; la démonstration pour le 2D est analogue. On considère un point de coordonnées (x,y,z) dans le cône continu $<i,j,k>$. Puisque $n_C^{\mathbb{R}}$ dépend linéairement des pondérations $(\overrightarrow{v_i},w_i)$, $(\overrightarrow{v_j},w_j)$ et $(\overrightarrow{v_k},w_k)$, on sait que $n_C^{\mathbb{R}}\big((x,y,z)\big) = \alpha.w_i + \beta.w_j + \gamma.w_k$ où α, β et γ sont tels que $(x,y,z) = \alpha.\overrightarrow{v_i} + \beta.\overrightarrow{v_j} + \gamma.\overrightarrow{v_k}$.

On calcule α, β et γ par la méthode des déterminants, et il vient

$$\alpha = \frac{1}{\Delta_{i,j,k}} \cdot \begin{vmatrix} x_j & x_k & x \\ y_j & y_k & y \\ z_j & z_k & z \end{vmatrix}, \quad \beta = \frac{-1}{\Delta_{i,j,k}} \cdot \begin{vmatrix} x_i & x_k & x \\ y_i & y_k & y \\ z_i & z_k & z \end{vmatrix}, \quad \gamma = \frac{1}{\Delta_{i,j,k}} \cdot \begin{vmatrix} x_i & x_j & x \\ y_i & y_j & y \\ z_i & z_j & z \end{vmatrix}.$$

Donc la norme a pour valeur

$$n_C^{\mathbb{R}}\big((x,y,z)\big) = \frac{1}{\Delta_{i,j,k}} \cdot \left\{ \begin{vmatrix} x_j & x_k & x \\ y_j & y_k & y \\ z_j & z_k & z \end{vmatrix}.w_i - \begin{vmatrix} x_i & x_k & x \\ y_i & y_k & y \\ z_i & z_k & z \end{vmatrix}.w_j + \begin{vmatrix} x_i & x_j & x \\ y_i & y_j & y \\ z_i & z_j & z \end{vmatrix}.w_k \right\},$$

qui est un développement par mineur de la formule du théorème. □

En développant les déterminants du théorème 7, on peut tirer pour une norme de chanfrein donnée, les valeurs δ_x, δ_y et δ_z permettant de calculer directement la valeur de n_C pour tout $(x,y,z) \in <i,j,k>$:

$$n_C(x,y,z) = \underbrace{\frac{1}{\Delta_{i,j,k}} \begin{vmatrix} y_i & y_j & y_k \\ z_i & z_j & z_k \\ w_i & w_j & w_k \end{vmatrix}}_{\delta_x}.x + \underbrace{\frac{-1}{\Delta_{i,j,k}} \begin{vmatrix} x_i & x_j & x_k \\ z_i & z_j & z_k \\ w_i & w_j & w_k \end{vmatrix}}_{\delta_y}.y + \underbrace{\frac{1}{\Delta_{i,j,k}} \begin{vmatrix} x_i & x_j & x_k \\ y_i & y_j & y_k \\ w_i & w_j & w_k \end{vmatrix}}_{\delta_z}.z\,. \tag{4.4}$$

Nous nommons ce vecteur $(\delta_x,\delta_y,\delta_z)$ le gradient discret du cône $<i,j,k>$.

4.3 Stratégie de construction d'une norme de chanfrein continue

Nous voulons dans la suite établir une stratégie tirant parti des résultats précédents, afin de construire des masques de chanfrein qui aient un comportement prévisible. Nous proposons la technique suivante :

1. Choisir un ensemble de points visibles $\overrightarrow{v_i}$ et leur attribuer une distance locale w_i afin de former des pondérations $(\overrightarrow{v_i},w_i)$ d'un masque de chanfrein $\mathcal{M}_C$. Une fois que les $\overrightarrow{v_i}$ sont choisis, les w_i représentent alors les inconnues du problème. Suivant ces valeurs, le comportement de $n_C^{\mathbb{R}}$ se modifie : les vecteurs du masque qui permettent la construction des chemins minimaux changent. On cherche à trouver les bonnes valeurs de w_i pour que le comportement de $n_C^{\mathbb{R}}$ soit aisément prévisible.
2. Déterminer le masque rationnel équivalent $\mathcal{M}'_C$.
3. Choisir une triangulation des points de $\mathcal{M}'_C$, c'est-à-dire choisir quels sont les cônes sur lesquels on veut que $n_C^{\mathbb{R}}$ soit une application linéaire.
4. S'assurer que la triangulation choisie est convexe, car ainsi elle sera égale à l'enveloppe convexe $\circledcirc\mathcal{M}'_C$, ce qui entraînera que toutes les pondérations sont utilisées et que la formule directe est correcte dans les cônes correspondants aux facettes de la triangulation choisie. Cette condition de convexité se traduit par des contraintes sur les distances locales w_i.

La suite de l'exposé traite uniquement de l'application de cette stratégie au cas des masques de chanfrein 3D, mais les méthodes mises en œuvre sont immédiatement transposables au cas 2D.

4.3.1 Boule rationnelle équivalente

On appelle *boule rationnelle équivalente* le polyèdre qui est formé en choisissant une triangulation des points du masque rationnel équivalent. La figure 4.4 donne quatre exemples de telles triangulations qui vont nous permettre d'illustrer notre technique.

- Le premier exemple est le cas le plus simple ; il est basé sur un masque de chanfrein $\mathcal{M}_C = \{(\overrightarrow{v_\mathsf{a}},w_\mathsf{a}),(\overrightarrow{v_\mathsf{b}},w_\mathsf{b}),(\overrightarrow{v_\mathsf{c}},w_\mathsf{c})\}$. La triangulation de la boule comporte 48 triangles, tous semblables à celui marqué d'un bord noir : le triangle $\left(\frac{\overrightarrow{v_\mathsf{a}}}{w_\mathsf{a}},\frac{\overrightarrow{v_\mathsf{b}}}{w_\mathsf{b}},\frac{\overrightarrow{v_\mathsf{c}}}{w_\mathsf{c}}\right)$. C'est le cas le plus simple que nous aborderons.
- Le deuxième exemple est basé sur un masque $\mathcal{M}_C = \{(\overrightarrow{v_\mathsf{a}},w_\mathsf{a}),\ (\overrightarrow{v_\mathsf{b}},w_\mathsf{b}),\ (\overrightarrow{v_\mathsf{c}},w_\mathsf{c}),\ (\overrightarrow{v_\mathsf{e}},w_\mathsf{e})\}$. Le sommet $\frac{\overrightarrow{v_\mathsf{e}}}{w_\mathsf{e}}$ qui se situe sur le plan $\left(O,\frac{\overrightarrow{v_\mathsf{a}}}{w_\mathsf{a}},\frac{\overrightarrow{v_\mathsf{c}}}{w_\mathsf{c}}\right)$, permet de couper chacun des triangles de la boule (a) en deux nouveaux triangles.
- Le troisième exemple est basé sur le même masque que le précédent mais sa triangulation est différente : une arête joint les sommets $\frac{\overrightarrow{v_\mathsf{e}}}{w_\mathsf{e}}$ et $\frac{\overrightarrow{v_{\mathsf{e}/\mathsf{ab}}}}{w_\mathsf{e}}$ au lieu de joindre $\frac{\overrightarrow{v_\mathsf{a}}}{w_\mathsf{a}}$ et $\frac{\overrightarrow{v_\mathsf{b}}}{w_\mathsf{b}}$ (en gris). Contrairement aux deux premiers exemples, cette triangulation est «à cheval» sur les bords de $\frac{1}{48}\mathbb{R}^3$ (l'analogue continu de $\frac{1}{48}\mathbb{Z}^3$). On appellera *triangulation transverse* toute triangulation qui croise ainsi les bords de $\frac{1}{48}\mathbb{R}^3$.

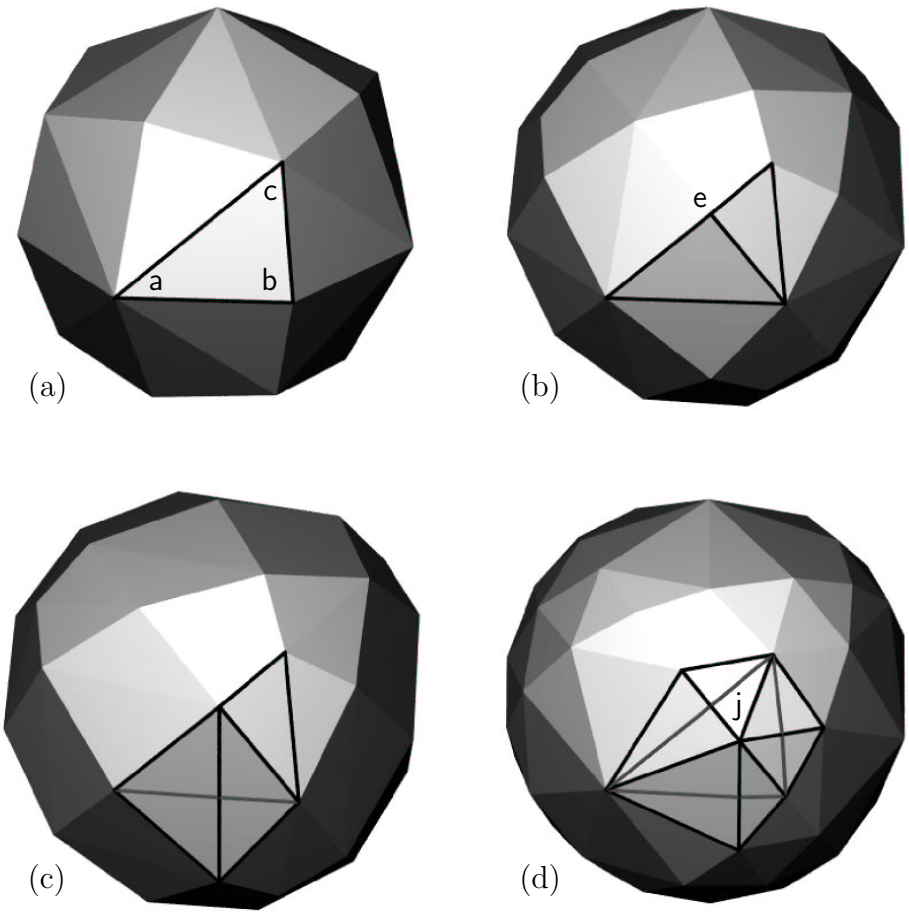


FIG. 4.4. *Quatre exemples de boules rationnelles équivalentes.*

– Le quatrième et dernier exemple est sur le même principe. À partir d'un masque $\mathcal{M}_C = \{(\overrightarrow{v_{\mathsf{a}}},w_{\mathsf{a}}),\ (\overrightarrow{v_{\mathsf{b}}},w_{\mathsf{b}}),\ (\overrightarrow{v_{\mathsf{c}}},w_{\mathsf{c}}),\ (\overrightarrow{v_{\mathsf{j}}},w_{\mathsf{j}})\}$, on a construit une triangulation dont aucun triangle n'est complètement inclus dans $\frac{1}{48}\mathbb{R}^3$.

4.3.2 Générateur complété du masque de chanfrein

On remarque que dans chacun des quatre cas précédents, la triangulation peut être déduite par symétrie sur le même principe que les pondérations du masque de chanfrein, à partir d'un sous-ensemble de ses triangles (ceux indiqués en noir). Ce sous-ensemble constitue le *générateur de la triangulation* et est noté $\triangle$. La triangulation complète (qui interviendra peu) sera notée $\mathrm{Sym}(\triangle)$ afin de montrer qu'on l'obtient par symétrie à partir de $\triangle$. Nous utiliserons donc un schéma en projection comme défini au §3.1.2 ainsi que les noms définis au §3.1.6, pour décrire nos choix pour $\triangle$, comme illustré à la figure 4.5.

Le générateur $\mathcal{M}_C^g$ ne contient pas toujours toutes les pondérations nécessaires à la définition de la sous-triangulation $\triangle$; c'est le cas dès que $\triangle$ est transverse, comme sur les exemples (c) et (d). En conséquence, nous introduisons la notion de *générateur complété*,

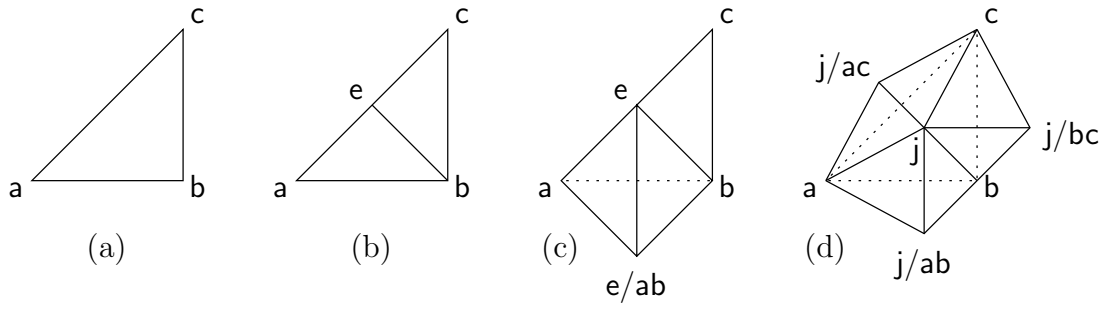


FIG. 4.5. *Les quatre représentations en projection de $\triangle$, correspondant aux quatre boules de la figure 4.4.*

noté $\mathcal{M}_C^{\triangle}$, qui est tout simplement l'ensemble des pondérations sur lesquelles s'appuie $\triangle$. Nous avons ainsi respectivement pour les cas (a) et (b) $\mathcal{M}_C^g = \mathcal{M}_C^{\triangle}$, pour le cas (c) $\mathcal{M}_C^{\triangle} = \{ (\overrightarrow{v_{\mathsf{a}}},w_{\mathsf{a}}), (\overrightarrow{v_{\mathsf{b}}},w_{\mathsf{b}}), (\overrightarrow{v_{\mathsf{c}}},w_{\mathsf{c}}), (\overrightarrow{v_{\mathsf{e}}},w_{\mathsf{e}}), (\overrightarrow{v_{\mathsf{e/ab}}},w_{\mathsf{e}}) \}$, et pour le cas (d) $\mathcal{M}_C^{\triangle} = \{ (\overrightarrow{v_{\mathsf{a}}},w_{\mathsf{a}}), (\overrightarrow{v_{\mathsf{b}}},w_{\mathsf{b}}), (\overrightarrow{v_{\mathsf{c}}},w_{\mathsf{c}}), (\overrightarrow{v_{\mathsf{j}}},w_{\mathsf{j}}), (\overrightarrow{v_{\mathsf{j/ab}}},w_{\mathsf{j}}), (\overrightarrow{v_{\mathsf{j/bc}}},w_{\mathsf{j}}), (\overrightarrow{v_{\mathsf{j/ac}}},w_{\mathsf{j}}) \}$.

Par construction, on a toujours $\mathcal{M}_C^g \subseteq \mathcal{M}_C^{\triangle} \subseteq \mathcal{M}_C^{1/2} \subset \mathcal{M}_C$.

4.3.3 Convexité de la boule rationnelle équivalente

Nous cherchons maintenant à obtenir une série de contraintes sur les valeurs des distances locales w_i du masque $\mathcal{M}_C$ dont la satisfaction signifie que la boule est convexe. Pour cela, nous allons tout d'abord introduire un *critère de convexité locale* entre deux facettes contiguës de la boule, puis nous généraliserons son emploi à toute la boule.

4.3.3.1 Convexité locale

Soit P, Q, R et S, quatre points de $\mathbb{R}^3$. On note $\delta^p(P,Q,R,S)$ le volume signé du parallélépipède orienté défini par les vecteurs $\overrightarrow{PQ}$, $\overrightarrow{PR}$ et $\overrightarrow{PS}$. Ce volume δ^p s'exprime en fonction du produit mixte des vecteurs par

$$\delta^p(P,Q,R,S) = \begin{vmatrix} x_Q - x_P & x_R - x_P & x_S - x_P \\ y_Q - y_P & y_R - y_P & y_S - y_P \\ z_Q - z_P & z_R - z_P & z_S - z_P \end{vmatrix} .$$

Ce déterminant 3×3 peut s'écrire encore sans les soustractions, sous la forme du déterminant 4×4 suivant :

$$\delta^p(P,Q,R,S) = \begin{vmatrix} x_Q & x_R & x_S & x_P \\ y_Q & y_R & y_S & y_P \\ z_Q & z_R & z_S & z_P \\ 1 & 1 & 1 & 1 \end{vmatrix} .$$

Considérons maintenant deux faces contiguës de la triangulation comme celles représentées à la figure 4.6. Elles sont définies par la donnée de leurs quatre sommets P, Q, R et S. On note $\delta^t(P,Q,R,S)$ le volume signé du tétraèdre orienté. Ce volume est positif lorsque l'angle φ est supérieur à 180°, nul quand $\varphi = 180°$ et négatif dès que $\varphi < 180°$.

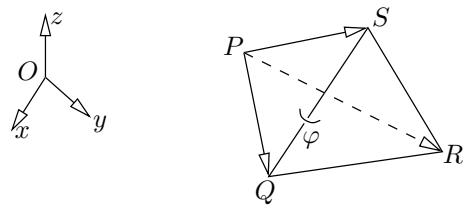


FIG. 4.6. *Tétraèdre* (P,Q,R,S) *direct.*

On dira que les facettes (P,Q,S) et (Q,R,S) sont *localement convexes* si et seulement si $\varphi \geqslant 180°$, ce qui est équivalent à $\delta^t(P,Q,R,S) \geqslant 0$. Or on a la relation entre les volumes $\delta^t(P,Q,R,S) = \frac{1}{6} \cdot \delta^p(P,Q,R,S)$, donc (P,Q,S) et (Q,R,S) sont localement convexes si et seulement si $\delta^p \geqslant 0$.

On en tire le théorème suivant :

Théorème 8 (Critère de convexité locale) *Soit deux triangles orientés contigus* (P,Q,S) *et* (Q,R,S) *de la triangulation* $\mathrm{Sym}(\triangle)$ *de* $\mathcal{M}'_C$*. Le critère de convexité locale entre ces deux facettes de la boule, noté* $\mathrm{CCL}(P,Q,R,S)$*, est*

$$w_P.\Delta_{Q,R,S} - w_Q.\Delta_{R,S,P} + w_R.\Delta_{Q,S,P} - w_S.\Delta_{Q,R,P} \geqslant 0 \ . \tag{4.5}$$

Preuve : On prend les P, Q, R et S parmi les points de $\mathcal{M}'_C$, et on utilise les coordonnées obtenues à partir des pondérations correspondantes de $\mathcal{M}_C$; on obtient

$$\delta^p(P,Q,R,S) = \begin{vmatrix} \frac{x_Q}{w_Q} & \frac{x_R}{w_R} & \frac{x_S}{w_S} & \frac{x_P}{w_P} \\ \frac{y_Q}{w_Q} & \frac{y_R}{w_R} & \frac{y_S}{w_S} & \frac{y_P}{w_P} \\ \frac{z_Q}{w_Q} & \frac{z_R}{w_R} & \frac{z_S}{w_S} & \frac{z_P}{w_P} \\ 1 & 1 & 1 & 1 \end{vmatrix} \ .$$

On factorise ensuite les dénominateurs pour obtenir

$$\delta^p(P,Q,R,S) = \frac{1}{w_P.w_Q.w_R.w_S} \cdot \begin{vmatrix} x_Q & x_R & x_S & x_P \\ y_Q & y_R & y_S & y_P \\ z_Q & z_R & z_S & z_P \\ w_Q & w_R & w_S & w_P \end{vmatrix} \ .$$

Or les distances locales w_i sont toujours strictement positives, donc $\delta^p(P,Q,R,S)$ est positif si et seulement si le déterminant est positif. On développe le déterminant par mineurs et on obtient la formule (4.5). □

4.3.3.2 Convexité globale

Pour que la triangulation $\mathrm{Sym}(\triangle)$ choisie sur les points de $\mathcal{M}'_C$ soit convexe, il suffit que chaque couple de faces contiguës vérifie le critère de convexité locale, c'est-à-dire que chaque arête entre les deux triangles soit située «vers l'extérieur» de la boule.

En pratique cependant, on ne compare que certains couples de faces, c'est-à-dire que l'on teste certaines arêtes de la triangulation mais pas leurs symétriques. En effet, il suffit d'appliquer le critère uniquement sur les arêtes qui sont au moins en partie dans $\frac{1}{48}\mathbb{R}^3$, comme indiqué pour les quatre exemples à la figure 4.7.

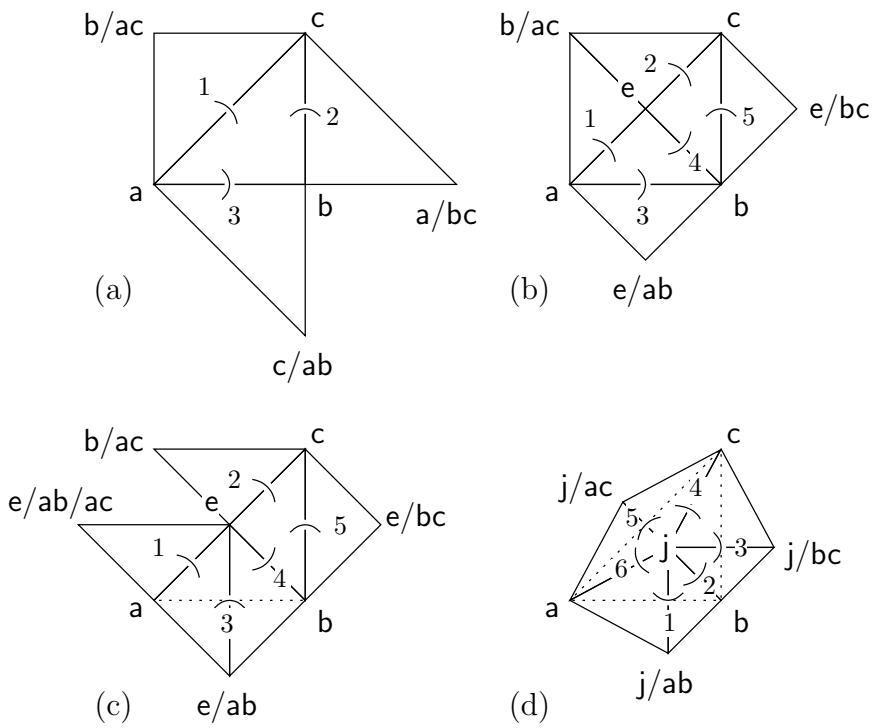


FIG. 4.7. *Contraintes de convexité locale pour les exemples de la figure 4.4.*

Pour le cas (a) de la figure 4.7, on a les contraintes suivantes :

1	CCL(b/ac,a,b,c) : $w_{\mathsf{a}} + w_{\mathsf{c}} \leqslant 2.w_{\mathsf{b}}$
2	CCL(a,b,a/bc,c) : $w_{\mathsf{b}} \leqslant 2.w_{\mathsf{a}}$
3	CCL(c/ab,b,c,a) : $w_{\mathsf{b}} \leqslant w_{\mathsf{c}}$

Pour le cas (b), on a :

1	CCL(b/ac,a,b,e) : $w_{\mathsf{e}} \leqslant 2.w_{\mathsf{b}}$
2	CCL(b/ac,e,b,c) : $w_{\mathsf{e}} \leqslant 2.w_{\mathsf{b}}$
3	CCL(e/ab,b,e,a) : $w_{\mathsf{a}} + w_{\mathsf{b}} \leqslant w_{\mathsf{e}}$
4	CCL(a,b,c,e) : $w_{\mathsf{e}} \leqslant w_{\mathsf{a}} + w_{\mathsf{c}}$
5	CCL(e,b,e/bc,c) : $w_{\mathsf{b}} + 2.w_{\mathsf{c}} \leqslant 2.w_{\mathsf{e}}$

Pour le cas (c), on a :

1	CCL(e/ab/ac,a,e/ab,e) : $2.w_{\mathsf{a}} \leqslant w_{\mathsf{e}}$
2	CCL(b/ac,e,b,c) : $w_{\mathsf{e}} \leqslant 2.w_{\mathsf{b}}$
3	CCL(a,e/ab,b,e) : $w_{\mathsf{e}} \leqslant w_{\mathsf{a}} + w_{\mathsf{b}}$
4	CCL(e/ab,b,c,e) : $w_{\mathsf{b}} \leqslant w_{\mathsf{c}}$
5	CCL(e,b,e/bc,c) : $w_{\mathsf{b}} + 2.w_{\mathsf{c}} \leqslant 2.w_{\mathsf{e}}$

Enfin, pour le cas (d), on a :

1	CCL(a,j/ab,b,j)	: $w_{\mathsf{j}} \leqslant w_{\mathsf{a}} + 2.w_{\mathsf{b}}$
2	CCL(j/ab,b,j/bc,j)	: $5.w_{\mathsf{b}} \leqslant 2.w_{\mathsf{j}}$
3	CCL(b,j/bc,c,j)	: $2.w_{\mathsf{j}} \leqslant 3.w_{\mathsf{b}} + 2.w_{\mathsf{c}}$
4	CCL(j/bc,c,j/ac,j)	: $2.w_{\mathsf{c}} \leqslant w_{\mathsf{j}}$
5	CCL(a,j,c,j/ac)	: $2.w_{\mathsf{j}} \leqslant 3.w_{\mathsf{a}} + 3.w_{\mathsf{c}}$
6	CCL(j/ab,j,j/ac,a)	: $3.w_{\mathsf{a}} \leqslant w_{\mathsf{j}}$

4.4 Retour au domaine discret

Puisque $n_C^{\mathbb{R}}$ est toujours une norme sur $\mathbb{R}^n$, il est bien évident qu'un des moyens d'obtenir que n_C soit une norme discrète est de s'assurer que pour tout vecteur $\overrightarrow{v}$ de $\mathbb{Z}^n$, on ait $n_C^{\mathbb{R}}(\overrightarrow{v}) = n_C(\overrightarrow{v})$.

Dès que la triangulation choisie pour la boule rationnelle équivalente est convexe, alors les cônes continus sur lesquels $n_C^{\mathbb{R}}$ est linéaire sont connus. Il ne nous reste plus qu'à replonger nos résultats dans le domaine discret, en commençant par définir les cônes discrets.

Définition 13 (Cône discret) *On définit le* cône discret $\ll i,j,k \gg$ *comme l'ensemble des points de* $\mathbb{Z}^3$ *qui sont inclus dans le cône continu* $< i,j,k >$.

4.4.1 Cône régulier

On sait que pour tout point M dépendant d'un cône discret $\ll i,j \gg$, on a

$$n_C^{\mathbb{R}}(\overrightarrow{OM}) = \lambda_i.w_i + \lambda_j.w_j \quad \text{où} \quad \overrightarrow{OM} = \lambda_i.\overrightarrow{v_i} + \lambda_j.\overrightarrow{v_j} \quad \text{et} \quad \lambda_i,\lambda_j \in \mathbb{R}^+ . \tag{4.6}$$

On cherche dans quel cas on peut avoir $n_C^{\mathbb{R}} \neq n_C$. On choisit de décomposer λ_i et λ_j en leurs parties entières et leurs parties fractionnaires ; on obtient :

$$\lambda_i = n_i + \lambda'_i \quad \text{et} \quad \lambda_j = n_j + \lambda'_j \quad \text{où} \quad n_i,n_j \in \mathbb{N} \quad \text{et} \quad \lambda'_i,\lambda'_j \in [0,1[. \tag{4.7}$$

On réécrit alors la formule (4.6) en tenant compte de cette décomposition :

$$\begin{aligned} n_C^{\mathbb{R}}(\overrightarrow{OM}) &= (\lambda'_i.w_i + \lambda'_j.w_j) + (n_i.w_i + n_j.w_j) \\ \text{où} \quad \overrightarrow{OM} &= (\lambda'_i.\overrightarrow{v_i} + \lambda'_j.\overrightarrow{v_j}) + (n_i.\overrightarrow{v_i} + n_j.\overrightarrow{v_j}) \\ &\quad \text{avec } n_i,n_j \in \mathbb{N} \quad \text{et} \quad \lambda'_i,\lambda'_j \in [0,1[. \end{aligned} \tag{4.8}$$

Pour que $n_C^{\mathbb{R}} \neq n_C$, il suffit que dans (4.8) l'un au moins de λ'_i et λ'_j soit non nul, car il est alors impossible de formuler la valeur correcte de la norme du vecteur $\overrightarrow{OM'} = \lambda'_i.\overrightarrow{v_i} + \lambda'_j.\overrightarrow{v_j}$. La figure 4.8 illustre un tel cas : du point M on déduit «en sens inverse» par un nombre entier de translations suivant $\overrightarrow{v_i}$ et $\overrightarrow{v_j}$, la position du point M'. Ce point M' appartient à ce que nous appelons le *premier parallélogramme* du cône $< i,j >$ (indiqué en gris sur la figure) et qui est formé de tous les points P de $\mathbb{R}^2$ atteints par la combinaison linéaire :

$$\overrightarrow{OP} = \lambda'_i.\overrightarrow{v_i} + \lambda'_j.\overrightarrow{v_j} \quad \text{où} \quad \lambda'_i,\lambda'_j \in [0,1[. \tag{4.9}$$

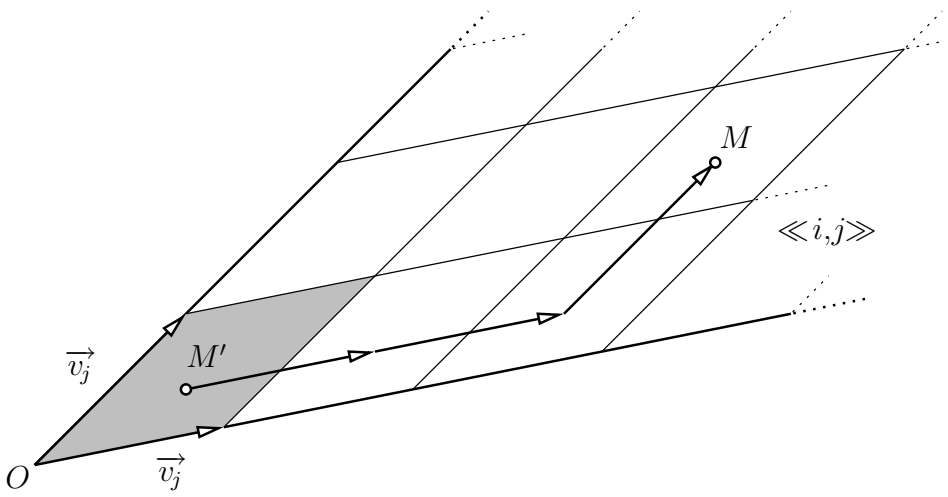


FIG. 4.8. *Pavage du cône ≪i,j≫ par des parallélogrammes identiques au premier (en gris). Tout point M de ≪i,j≫ correspond à un point M' dans ce premier parallélogramme.*

De même en 3D, on parle de *premier parallélépipède* d'un cône $<i,j,k>$.

On garde cependant en mémoire que M' est bel et bien un point du cône discret ≪i,j≫ et donc un point de la grille discrète, car M appartient à la grille discrète, et la translation de vecteur $n_i.\overrightarrow{v_i} + n_j.\overrightarrow{v_j}$ a lieu en restant sur la grille.

Donc en résumé, pour savoir si $n_C^{\mathbb{R}} = n_C$ dans un cône $<i,j>$, il est nécessaire et suffisant de trouver s'il existe de tels points M' de la grille discrète $\mathbb{E}$ dans le premier parallélogramme du cône $<i,j>$.

Pour déterminer si de tels points M' existent, nous proposons de nous servir du théorème suivant, dû à Minkowski :

Théorème 9 *Le parallélogramme défini par $\overrightarrow{v_i}$ et $\overrightarrow{v_j}$ (resp. le parallélépipède défini par $\overrightarrow{v_i}$, $\overrightarrow{v_j}$ et $\overrightarrow{v_k}$) ne contient aucun point de $\mathbb{Z}^2$ (resp. $\mathbb{Z}^3$) autre que ses sommets si et seulement si $\Delta_{i,j} = \pm 1$ (resp. $\Delta_{i,j,k} = \pm 1$).* [Har78, chap. 3]

On en tire que si $\Delta_{i,j} = \pm 1$ (resp. $\Delta_{i,j,k} = \pm 1$) alors il n'existe pas de point M' de $\mathbb{Z}^2$ (resp. $\mathbb{Z}^3$) dans le premier parallélogramme (resp. premier parallélépipède) et donc que $n_C^{\mathbb{R}} = n_C$ dans tout ce cône ≪i,j≫.

Définition 14 (Cône régulier) *On appelle cône régulier tout cône discret ≪i,j≫ (resp. ≪i,j,k≫) tel que $\Delta_{i,j} = \pm 1$ (resp. $\Delta_{i,j,k} = \pm 1$).*

On appelle *triangle régulier* tout triangle de Sym($\triangle$) correspondant à un cône régulier et on appelle *triangulation régulière*, une triangulation qui ne comporte que des triangles réguliers. Par abus de langage, on appelle *masque régulier* un masque pour lequel on a fixé une triangulation régulière.

On observe sur la figure 4.9, que les triangles des exemples (a) et (b) définissent bien des cônes réguliers mais que deux des trois triangles de (c) et que les six triangles de (d) ne sont pas réguliers.

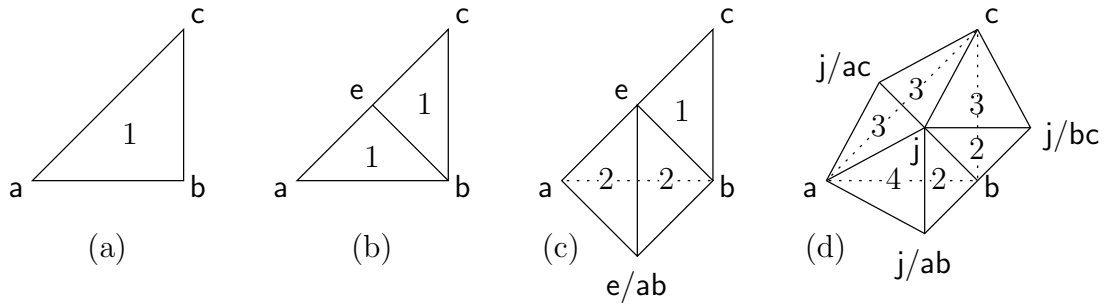


FIG. 4.9. *Les déterminants des triangles correspondant aux quatre triangulations de la figure 4.4.*

Le schéma (a) justifie en dimension 3 ce qui était déjà connu, et que Kiselman a prouvé dans [Kis96] : les masques de taille 3 à n pondérations en dimension n définissent toujours une norme discrète puisqu'ils ne sont formés que d'un seul type de cône et que le déterminant de ce cône vaut toujours 1.

Le schéma (c) correspond à la triangulation du masque proposé par Borgefors et Svensson (cas II de la figure 3 de [Bor01]) et qui nous a servi de base au contre-exemple du §3.2.4. Les deux triangles (a,e/ab,e) et (e/ab,b,e) sont les deux triangles non réguliers, ce qui justifie une nouvelle fois que dans ce cas, on ait $n_C^{\mathbb{R}} \neq n_C$ et donc que n_C ne soit pas une norme discrète. Inversement, la triangulation (b) est identique au cas I de la figure 3 dans [Bor01] et correspond bien à une norme.

En conclusion, puisque dans tout cône régulier on a $n_C^{\mathbb{R}} = n_C$, on cherche à faire en sorte que le générateur $\triangle$ de la triangulation soit régulier et donc par symétrie la triangulation complète $\mathrm{Sym}(\triangle)$ soit régulière.

4.4.2 Triangulation de Farey

La technique qui suit a été publiée dans [Rem00b] et [Rem00c], et constitue une extension en 3D de celle publiée dans [Thi94], qui était basée sur la comparaison entre les suites de Farey et les masques de chanfrein 2D.

Cette technique permet de construire une triangulation régulière à partir de règles de subdivision de la triangulation de l'exemple (a) de la figure 4.4.

Les *suites de Farey* F_n d'ordre n sont les suites croissantes de fractions irréductibles entre 0 et 1, dont les dénominateurs n'excèdent pas n. Donc $\frac{h}{k} \in F_n$ si $0 \leqslant h \leqslant k \leqslant n$ et

pgcd$(h,k) = 1$. Par exemple, on a

$$\begin{aligned} F_1 &= \left\{ \frac{0}{1}, \frac{1}{1} \right\} , \\ F_2 &= \left\{ \frac{0}{1}, \frac{1}{2}, \frac{1}{1} \right\} , \\ F_3 &= \left\{ \frac{0}{1}, \frac{1}{3}, \frac{1}{2}, \frac{2}{3}, \frac{1}{1} \right\} , \\ F_4 &= \left\{ \frac{0}{1}, \frac{1}{4}, \frac{1}{3}, \frac{1}{2}, \frac{2}{3}, \frac{3}{4}, \frac{1}{1} \right\} . \end{aligned}$$

Une propriété notable des suites de Farey est que pour tout couple d'éléments $\frac{h}{k}$ et $\frac{h'}{k'}$ consécutifs d'une suite F_n on a $k.h' - h.k' = 1$.

Thiel montre dans [Thi94, p. 26-27] que pour un masque de chanfrein 2D, si on choisit des pondérations $\big((x_i,y_i),w_i\big)$ et $\big((x_j,y_j),w_j\big)$ successives dans l'ordre angulaire et telles que $\frac{y_i}{x_i}$ et $\frac{y_j}{x_j}$ soient des éléments successifs d'une même suite F_n, alors (x_i,y_i) et (x_j,y_j) sont des points visibles et de plus, on a automatiquement $\Delta_{i,j} = 1$.

L'extension aux masques 3D fait intervenir les ensembles de Farey (voir par exemple [Gra92]).

Définition 15 (Ensemble de Farey) *Les ensembles de Farey $\widehat{F}_n$ d'ordre n sont les ensembles de points irréductibles $\left(\frac{y}{x},\frac{z}{x}\right)$ dans $[0,1]^2$ dont le dénominateur n'excède pas n.*

Donc $\left(\frac{y}{x},\frac{z}{x}\right)$ est un élément de $\widehat{F}_n$ si $x \leqslant n$, $0 \leqslant y \leqslant x$, $0 \leqslant z \leqslant x$ et si la forme est irréductible, c'est-à-dire si pgcd$(x,y,z) = 1$; donc $(x,y,z) \in \mathcal{V}_n$ ou $(x,z,y) \in \mathcal{V}_n$ (cf. §3.1.6). On en tire la correspondance entre les points visibles 3D et les éléments des ensembles de Farey:

$$\widehat{F}_n = \left\{ \left(\frac{y}{x},\frac{z}{x}\right) \in \mathbb{Q}^2 \ : \ (x,y,z) \in \mathcal{V}_n \cup \mathcal{V}_n/\mathsf{ac} \right\}$$

où $\mathcal{V}_n/\mathsf{ac}$ est l'ensemble des symétriques par rapport au plan $(O,\mathsf{a},\mathsf{c})$ des points de $\mathcal{V}_n$.

Étant donné deux points $\left(\frac{y}{x},\frac{z}{x}\right)$ et $\left(\frac{y'}{x'},\frac{z'}{x'}\right)$ de $\widehat{F}_n$, on définit l'opération $\widehat{+}$ par

$$\left(\frac{y}{x},\frac{z}{x}\right) \widehat{+} \left(\frac{y'}{x'},\frac{z'}{x'}\right) = \left(\frac{y+y'}{x+x'}, \frac{z+z'}{x+x'}\right) .$$

Étant donné $\widehat{A}$ et $\widehat{B}$ deux points de $\widehat{F}_n$, on appelle *point médian* de $\widehat{A}$ et $\widehat{B}$, le point $\widehat{A}\widehat{+}\widehat{B}$ de $\mathbb{Q}^2$.

Théorème 10 (Mönkemeyer) *Soit $(\widehat{Q},\widehat{R},\widehat{S})$ un triangle régulier de $\widehat{F}_n$, et soit $\widehat{P}$ un point de $\widehat{F}_{n+1}$, inclus dans le triangle et distinct des sommets ; alors $\widehat{P}$ est le médian de deux des sommets* [Gra92].

On en déduit que ce point $\widehat{P}$ de $\widehat{F}_{n+1}$ est obligatoirement situé sur les bords du triangle régulier $(\widehat{Q},\widehat{R},\widehat{S})$ de $\widehat{F}_n$.

On appelle *triangulation de Farey* $F_n^{\triangle}$, l'ensemble de Farey $\widehat{F}_n$, muni d'une triangulation de ses points telle que tous les triangles soient réguliers. On dit qu'une séquence

de triangulation de Farey est *compatible* si la triangulation $F_n^{\triangle}$ est un raffinement de la triangulation de $F_{n-1}^{\triangle}$.

Par le théorème 10, on a donc un procédé de construction de tout $\widehat{F}_{n+1}$ à partir de $F_n^{\triangle}$. La figure 4.4.2 montre deux exemples de séquences compatibles de raffinements du triangle (a,b,c).

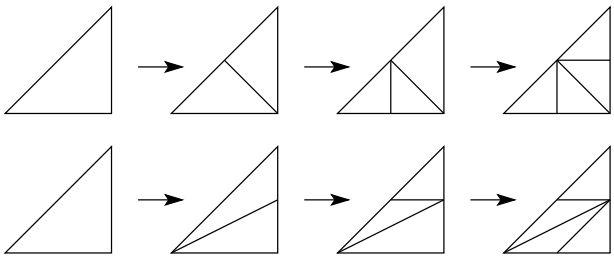


FIG. 4.10. *Séquences compatibles de triangulations de Farey dans* $\widehat{F}_2$.

Ce procédé permet par raffinements successifs d'une triangulation régulière de Farey, de fabriquer des triangulations régulières de plus en plus complexes ; malheureusement il a deux défauts importants :

- tout d'abord, l'intervention des points médians conduit, au fur et à mesure des raffinements successifs, à la formation de triangles de plus en plus allongés, or on peut désirer que l'aspect de la triangulation de la boule soit à peu près uniforme sur toute la surface ;
- d'autre part, le raffinement ne permet pas de construire toutes les triangulations régulières imaginables (cf. figure 4.11).

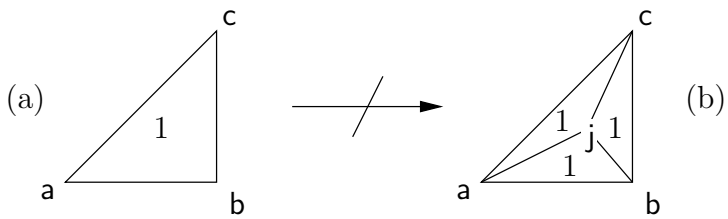


FIG. 4.11. *La triangulation régulière (b) ne peut pas être obtenue par raffinement successif de celle de (a).*

Finalement, cette technique ne permet pas de trouver toutes les triangulations régulières possibles, mais en revanche toutes les triangulations qu'elle permet de construire sont régulières et donc correspondent automatiquement à des normes discrètes.

4.4.3 Cônes non-réguliers et surcharge de $\mathcal{M}_C^g$

Après avoir montré l'intérêt des triangulations régulières, on va maintenant changer d'objectif et montrer qu'il est également possible de construire une norme avec une triangulation non-régulière.

On a vu que si $\Delta_{i,j} = \pm 1$ (resp. $\Delta_{i,j,k} = \pm 1$) alors c'est qu'il existe au moins un point M' dans le premier parallélogramme (resp. parallélépipède) de $\ll i,j \gg$ (resp. $\ll i,j,k \gg$) tel que $n_C(\overrightarrow{OM'}) > n_C^{\mathbb{R}}(\overrightarrow{OM'})$, car il n'existe pas de moyen d'atteindre M' par une combinaison linéaire entière des pondérations $(\overrightarrow{v_i},w_i)$ et $(\overrightarrow{v_j},w_j)$.

Soit M' un tel point un point incorrectement étiqueté vis à vis de n_C lors de la transformation de distance avec le masque de chanfrein $\mathcal{M}_C$. Nous proposons d'ajouter dans le masque de chanfrein $\mathcal{M}_C$ une pondération qui force la valeur de $n_C(\overrightarrow{OM})$ à être étiqueté correctement lors de la transformation de distance.

Définition 16 (Surcharge d'un masque) *On corrige l'erreur commise lors de la transformation de distance, en rajoutant la nouvelle pondération* $\left(\overrightarrow{OM'},n_C^{\mathbb{R}}(\overrightarrow{OM'})\right)$ *au générateur* $\mathcal{M}_C^g$*. On appelle* surcharger un masque*, l'opération consistant à rajouter dans le générateur* $\mathcal{M}_C^g$*, une pondération pour chaque point* M' *incorrectement étiqueté par* n_C *; le générateur résultant s'appelle* générateur surchargé*.*

Notons maintenant que l'opération de surcharge nécessite de calculer par la formule directe (cf. §4.2.5) la valeur de $n_C^{\mathbb{R}}(\overrightarrow{OM'})$; mais qu'elle n'est pas forcément entière puisque $\Delta_{i,j} \neq \pm 1$ (resp. $\Delta_{i,j,k} \neq \pm 1$). Si on ne tient à construire que des masques de chanfrein où les distances locales w_i sont des entiers (et non des rationnels), alors la surcharge n'est pas toujours possible, et la triangulation est inexploitable puisque n_C n'est pas prévisible sur le cône.

En pratique, on observe parfois des cas où le point M' est correctement étiqueté à la valeur $n_C^{\mathbb{R}}(\overrightarrow{OM'})$ même sans l'ajout de la pondération supplémentaire. Cela se produit par exemple quand M' est sur une frontière de cône et que l'autre cône est, lui, régulier et donc «corrige» l'erreur du premier cône.

L'opération de surcharge induit inévitablement un coût algorithmique plus grand pour le calcul de la transformée de distance puisque le nombre de vecteurs dans le masque est plus grand. Il est bien évidement préférable de trouver des triangulations qui soient les plus régulières possibles afin de garder un coût de calcul faible.

Dans la suite de ce document, on ne distinguera plus les masques réguliers de ceux qui auront été surchargés avec succès (puisqu'ils auront les mêmes propriétés). On les nommera indifféremment *masques réguliers.*

4.5 Conclusion

Dans ce chapitre, nous avons montré que tout masque de chanfrein définit une norme de chanfrein continue $n_C^{\mathbb{R}}$, et que seules participent au calcul effectif de $n_C^{\mathbb{R}}$, les pondérations du masque rationnel équivalent $\mathcal{M}'_C$ qui sont des points extrêmes de son enveloppe convexe $\circledcirc\mathcal{M}'_C$. Nous avons montré que $n_C^{\mathbb{R}}$ est une application linéaire par morceaux sur les cônes, et nous avons donné ensuite le moyen de calculer $n_C^{\mathbb{R}}(\overrightarrow{v})$ directement à partir des coordonnées de $\overrightarrow{v}$.

Nous avons exposé notre méthode pour construire efficacement un masque de chanfrein, en introduisant les notions de triangulation et de générateur complété du masque de chan-

frein. Nous avons montré comment obtenir, sur la base d'un critère local de convexité de la boule de chanfrein, une norme dont le comportement est connu à l'avance. Ce critère local de convexité CCL se traduit par une suite de contraintes sur les valeurs des distances locales w_i formant les pondérations du masque de chanfrein $\mathcal{M}_C$.

Enfin, nous avons montré comment replonger ces résultats dans le domaine discret. Nous avons défini la notion de cône discret régulier, et nous avons donné une méthode basée sur les triangulations de Farey qui permet de construire une triangulation régulière, et ainsi une norme, à partir de tout masque de chanfrein. Nous avons aussi montré comment obtenir une norme par une technique appelée surcharge, à partir d'une triangulation non-régulière, si cela s'avère nécessaire.

Il nous reste cependant à montrer des réalisations pertinentes des contraintes ainsi trouvées ; c'est l'objet du prochain chapitre.

Chapitre

5

Optimisation des normes de chanfrein

5.1 Introduction

Dans le chapitre précédant, nous avons trouvé sous quelles conditions un masque de chanfrein induit une norme. Nous montrons dans ce chapitre comment trouver des masques qui satisfassent ces contraintes, mais aussi qui répondent au besoin d'approcher autant que possible la distance euclidienne d_E.

Nous présentons tout d'abord deux méthodes parmi celles proposées dans la littérature, ainsi que les trois critères d'évaluation des masques les plus couramment utilisés.

Nous présentons ensuite notre méthode d'optimisation qui comporte trois étapes distinctes : l'énumération des distances locales, la surcharge des masques non-réguliers et le calcul de l'erreur.

Enfin, nous présentons une série de masques optimaux 3D pour des triangulations régulières et non-régulières.

5.2 Optimisation

Contrairement à ce qu'on trouve dans la majorité des articles publiés (où il n'est pas tenu compte des propriétés de n_C durant l'optimisation), nous cherchons ici à optimiser un masque de chanfrein sous la contrainte que celui-ci donne une norme. Nous avons vu au chapitre 4 qu'un masque de chanfrein doit vérifier un certain nombre de critères locaux de convexité pour donner une norme (cf. exemples au §4.3.3.2) ; ces contraintes restreignent considérablement le domaine de solutions de notre optimisation.

D'autres part, notre objectif est d'obtenir des normes de chanfrein n_C dont les valeurs soient aussi proches que possible de celles de la norme euclidienne n_E. Pour cela, il nous faut bien évidement pouvoir comparer les résultats de n_C et n_E. Nous faisons donc intervenir un facteur d'échelle réel noté s dans l'expression de n_C comme présenté au §3.1.3. En tirant parti des propriétés des normes de chanfrein, nous obtenons grâce à la formule directe (théorème 7 du §4.2.5) que la valeur *mise à l'échelle* de n_C pour un vecteur (x,y,z) appartenant à un cône discret $\ll i,j,k \gg$ est :

$$\frac{1}{s} \cdot n_C\big((x,y,z)\big) = \frac{1}{s} \cdot \frac{1}{\Delta_{i,j,k}} \cdot \begin{vmatrix} x_i & x_j & x_k & x \\ y_i & y_j & y_k & y \\ z_i & z_j & z_k & z \\ w_i & w_j & w_k & 0 \end{vmatrix}. \tag{5.1}$$

Sachant que nous choisissons à la création du masque, chacune des directions de pondérations (x_i,y_i,z_i), ainsi que la triangulation désirée (ce qui donne les contraintes de convexité locale), les variables de notre optimisation sont bien sûr les distances locales w_i , mais aussi le facteur d'échelle global s.

5.3 Différentes méthodes et différents critères d'erreur

5.3.1 La méthode de Borgefors

La première méthode d'optimisation est apparue dans [Bor84]. L'auteur a visiblement recherché une méthode très générale puisque l'article décrit l'optimisation pour les cas 1D, 2D, 3D et 4D, et ce pour différentes fonctions discrètes, parmi lesquelles les distances de chanfrein. Cependant, l'étude relative aux distances de chanfrein ne porte que sur les masques de taille 3, et il faut attendre des articles ultérieurs, et en particulier [Bor86], pour voir l'étude complétée en 2D, pour des masques de taille 3, 5 et 7.

Vossepoel propose ensuite dans [Vos88] l'utilisation d'un facteur d'échelle réel $s \neq w_{\mathsf{a}}$ afin d'améliorer les résultats obtenus. Borgefors prend en compte cette idée dans [Bor91a], et montre que le calcul peut être fait de façon exacte comme dans [Bor84] et non de façon numérique comme dans [Vos88].

Borgefors propose dans [Bor91a] la notion de régularité (dite au sens de Borgefors) telle qu'elle est rappelée au §3.2.3 . Kiselman montre dans [Kis96] que cette régularité est suffisante pour assurer qu'un masque de taille 3 en toute dimension induit bien une norme et donc permette l'utilisation de la formule (5.1). Il montre aussi que toute norme de chanfrein découle nécessairement d'un masque régulier, mais il souligne enfin que la régularité d'un masque de taille supérieure à 3 n'implique peut être pas forcément que la fonction associée soit une norme. Nous avons montré au §3.2.4 que tel n'est effectivement pas le cas.

Borgefors généralise son étude en liant régularité et optimisation dans le cas des masques réguliers de taille 3 en 4D dans [Bor00]. Elle se tourne ensuite vers l'optimisation d'un masque 3D du type $\mathcal{M}_C^g = \{(\overrightarrow{v_{\mathsf{a}}},w_{\mathsf{a}}), (\overrightarrow{v_{\mathsf{b}}},w_{\mathsf{b}}), (\overrightarrow{v_{\mathsf{c}}},w_{\mathsf{c}}), (\overrightarrow{v_{\mathsf{e}}},w_{\mathsf{e}})\}$ dans [Bor01].

L'optimisation de Borgefors consiste a évaluer le maximum de la fonction

$$f_1 : (x,y,z) \longmapsto \left| \frac{n_C\big((x,y,z)\big)}{s} - n_E\big((x,y,z)\big) \right| \tag{5.2}$$

où (x,y,z) est un point de $\frac{1}{48}\mathbb{R}^3$ qui est sur un plan $x = M$ constant donné. Le calcul est effectué analytiquement en simplifiant l'expression (5.2) dans chaque cône pour la rendre de la forme :

$$f_1 : (x,y,z) \longmapsto \alpha.M + \beta.y + \gamma.z - \sqrt{M^2 + y^2 + z^2}\,, \tag{5.3}$$

où α, β et γ sont des constantes pour un cône donné si le masque est régulier. On cherche dans chaque cône la position (x,y,z) du maximum MaxDiff de f_1 en utilisant intensivement le lemme suivant issu de [Bor96, Lemme 1] :

Lemme 2 *La fonction* $f(\eta) = \alpha.\eta + \gamma.\xi + \beta.M - \sqrt{M^2 + \xi^2 + \eta^2}$, *où* $|\alpha| < 1$ *et* $M > 1$,

atteint son maximum $f_{\max} = \beta.M + \gamma.\xi - \sqrt{M^2+\xi^2}.\sqrt{1-\alpha^2}$ *pour* $\eta = \frac{\alpha.\sqrt{M^2+\xi^2}}{\sqrt{1-\alpha^2}}$ *si* $\xi \neq 0$, *et* $f_{\max} = (\beta - \sqrt{1-\alpha^2}).M$ *pour* $\eta = \frac{\alpha.M}{\sqrt{1-\alpha^2}}$ *si* $\xi = 0$.
On calcule ensuite le maximum global qui est le maximum de tous les maxima précédemment trouvés.

Lorsqu'on considère le cas du contre-exemple du §3.2.4, la formule directe (5.1) n'est pas directement valable pour n_C, et il est donc nécessaire de procéder à l'évaluation de l'erreur commise en recourant à une astuce : plusieurs sous réseaux cohabitent sur l'image, et sur chacun d'eux la formule directe est valide. L'étiquetage produit étant le minimum pour chaque point des différents étiquetages de chaque sous-réseau, trouver le minimum de l'erreur commise est donc possible en calculant le minimum de l'erreur commise par chaque sous-réseau, puis en gardant comme erreur globale, le minimum de chacune de ces erreurs. Cette démarche est assez lourde puisqu'elle oblige à considérer de très nombreux cas, et qu'elle risque d'entraîner des erreurs de calculs.

Le principal défaut de cette méthode à nos yeux est que plus la taille du masque est grande et/ou la dimension élevée, et plus le nombre de cônes est grand, plus le calcul est fastidieux et le risque d'erreur humaine important. Cela rend quasiment impossible la systématisation du procédé. D'autre part, il est possible d'obtenir un masque dont la triangulation ne soit pas régulière (puisque l'optimisation est opérée indépendamment de la triangulation), et donc ne donnant pas une norme : rien ne garantis donc l'obtention finale d'un masque de norme optimal.

5.3.2 Méthode de Verwer

Le choix de l'optimisation sur un plan $x = M$ est discuté par différents auteurs : premièrement car le résultat est fonction de M, mais aussi car il est souligné que l'évaluation de l'erreur est ainsi biaisée par un effet de parallaxe : plus on s'éloigne de l'axe x, plus la mesure de l'erreur est forte, alors qu'on voudrait qu'elle soit uniforme sur toute la boule. Cela conduit Vossepoel [Vos88] puis Verwer [Ver91b] à développer des techniques d'optimisation où le point (x,y,z) parcourt la sphère de rayon unitaire afin de rendre isotrope la mesure de l'erreur.

Thiel montre dans [Thi94] que chercher comme Verwer le maximum de f_1 avec (x,y,z) qui parcourt la portion de sphère de rayon 1 contenue dans $\frac{1}{48}\mathbb{R}^3$ est équivalent à chercher le maximum de la fonction d'erreur relative

$$f_2 : (x,y,z) \longmapsto \left| \frac{n_C\big((x,y,z)\big)/s - n_E\big((x,y,z)\big)}{n_E\big((x,y,z)\big)} \right| \tag{5.4}$$

où (x,y,z) parcourt $\frac{1}{48}\mathbb{R}^3$. La valeur d'erreur obtenue, notée MAE (uk: *Maximum Absolute Error*), n'entraîne plus l'erreur de parallaxe observée avec MaxDiff.

Toujours dans [Ver91b], Verwer reprend une idée de critère d'évaluation de l'erreur suggérée (mais pas développée) dans [Bor86] : évaluer la moyenne de l'erreur commise plutôt que son maximum. La méthode proposée consiste, après avoir exprimé $\overrightarrow{v} =$

$(\cos\phi\sin\theta, \sin\phi\sin\theta, \cos\theta)$ en coordonnées polaires, à minimiser la variance σ définie par

$$\sigma^2 = \frac{12}{\pi} \cdot \int_0^{\pi/4} \int_{f(\phi)}^{\pi/2} \big(n_C(\overrightarrow{v})/s - 1\big)^2 \, p(\theta,\phi) \, d\theta \, d\phi \tag{5.5}$$

sous contrainte que le biais β

$$\beta = \frac{12}{\pi} \cdot \int_0^{\pi/4} \int_{f(\phi)}^{\pi/2} \big(n_C(\overrightarrow{v})/s - 1\big) \, p(\theta,\phi) \, d\theta \, d\phi \tag{5.6}$$

soit nul. Ce critère est habituellement nommé MSE (uk: *Mean Square Error*).

Le domaine d'intégration est triangulaire puisqu'il est contenu dans $\frac{1}{48}\mathbb{R}^3$, et donc les bornes sur θ dépendent de ϕ : on a $f(\phi) = \arctan(1/\sin\phi)$. D'autre part, Beckers et Smeulders justifient dans [Bec90] la présence dans la formule de la densité de probabilité $p(\theta,\phi) = \sin\theta$.

L'intégration de ces formules est possible de façon exacte en 2D mais par trop complexe en 3D ; Verwer effectue donc numériquement le calcul grâce au logiciel Mathematica. On note que dans les tableaux donnés par Verwer, le coefficient $12/\pi$ a été oublié durant le calcul de σ^2 ; les valeurs proposées sont donc justes à un coefficient $\sqrt{12/\pi}$ près.

5.4 Notre méthode

La méthode qui est détaillée dans cette partie a été publiée dans [Rem00b] ; elle reprend en l'étendant le principe de celle présentée dans [Thi94].

Son principe est simple : on énumère l'espace des solutions des contraintes de convexité locale, c'est-à-dire on trouve toutes les distances locales w_i les vérifiant. Puis, pour chacun des masques trouvés, si la triangulation n'est pas régulière (cf. §4.4.3), on surcharge le masque pour le rendre correct. Si la surcharge n'est pas possible, on rejette ce masque. Cette phase de surcharge constitue la principale différence dans la méthode par rapport à [Thi94].

On dispose à l'issue de cette phase d'un masque qui induit une norme de chanfrein ; il ne reste plus qu'à évaluer l'erreur commise, et à conserver les masques qui produisent les plus faibles erreurs. On appelle donc *masque optimal*, un masque de chanfrein qui réalise une erreur plus faible que tous les masques précédemment énumérés.

Notre méthode permet d'utiliser tous les critères d'erreur (MaxDiff, MAE et MSE) puisque l'énumération et le calcul d'erreur sont deux étapes distinctes. En pratique cependant, on ne choisit que des critères d'erreur pour lesquels il existe un moyen de calcul systématique efficace. En l'occurrence, pour le calcul de la variance (MSE) en 3D, il est nécessaire d'intégrer numériquement les formules (5.6) et (5.5) afin de tirer s et σ. La convergence nécessaire pour ce calcul fait qu'il est difficile de trouver un compromis entre la vitesse nécessaire à l'énumération et la précision évidement cruciale du résultat.

Notre approche est très différente de la plupart de celles précédemment publiées puisque notre but n'est pas de calculer un masque optimal dont les distances locales w_i sont des réels, et seulement ensuite de trouver des masques entiers qui s'en rapprochent, mais bel et bien

de trouver les meilleurs masques *effectifs* puisque nous les énumérons et les évaluons *tous*. En effet, on constate dans certains articles que rien ne garantit l'optimalité des masques entiers proposés (cf. contre-exemple dans [Thi94, p. 61]).

5.4.1 Comment énumérer ?

L'énumération brute que nous effectuons consiste simplement à imbriquer un certain nombre de boucles `for`, une par variable w_i à énumérer. Nous choisissons un ordre sur les i : l'ordre lexicographique sur les points visibles. En d'autres termes, on balaye tous les masques pour lesquels $w_{\mathsf{a}} = n$ avant de passer aux masques pour lesquels $w_{\mathsf{a}} = n+1$. Notre choix n'est pas anodin, il conditionne par exemple la propriété d'optimalité d'un masque ; il se justifie par le besoin d'avoir les meilleurs masques possibles pour des faibles valeurs de w_{a}. En effet, w_{a} est lié au facteur d'échelle s et donc à la taille des entiers qui sont stockés dans la carte de distance. Vu la taille importante de données que peut représenter le stockage d'un volume numérique en mémoire, il est crucial d'utiliser les plus petits entiers possibles pour le codage des valeurs des points de l'image. En pratique, on n'utilise jamais de w_{a} supérieurs à 255, ce qui nous permet d'utiliser un codage sur 16 bits (non signés) pour les points d'une image de côté 2×255 (en considérant que la plus grande boule possible dans l'image aura un rayon de l'ordre de 255 points).

Les contraintes de convexité locale nous donnent un certain nombre d'inégalités qu'il est souvent nécessaire de manipuler quelque peu avant de procéder à l'énumération. On cherche en particulier à tirer une borne supérieure et une borne inférieure pour chaque w_i.

Dans le cas où il nous manque une des bornes pour une des variables w_i, on calcule une ou plusieurs contraintes de convexité supplémentaires qui découlent de celles déjà obtenues. On se sert du fait que si on fusionne plusieurs facettes d'une triangulation convexe pour former un nouveau triangle, alors la triangulation résultante doit encore être convexe. On tire ainsi des contraintes plus faibles qui ne servent qu'à l'énumération.

5.4.2 Surcharge du masque

Nous devons maintenant déterminer si le masque trouvé a besoin ou non d'être surchargé (cf. §4.4.3). Pour cela nous proposons une méthode basée sur l'algorithme de la figure 5.1, qui permet de calculer efficacement à partir du générateur complété (introduit au §4.3.2), une carte de distance à l'origine dans $\frac{1}{48}\mathbb{Z}^3$, en un seul balayage. Le résultat produit est du même type que la figure 3.13 (b). Soit L la longueur du côté de l'image, n sa dimension et $m_C^{\triangle}$ le nombre de pondérations dans le générateur complété $\mathcal{M}_C^{\triangle}$ du masque de chanfrein, la complexité de cet algorithme est de l'ordre de $O(m_C^{\triangle} \cdot \frac{1}{2^n.n!} \cdot L^n)$ où le terme $\frac{1}{2^n.n!} \cdot L^n$ donne le nombre de points dans les cônes $\frac{1}{8}\mathbb{Z}^2$ et $\frac{1}{48}\mathbb{Z}^3$ en fonction de la dimension n.

Afin de vérifier si le masque a besoin d'être surchargé, on calcule une carte de distance à l'origine CT^g, puis pour chacun des points M' identifiés au §4.4.1 et §4.4.3, on compare la valeur trouvée dans CT^g avec la valeur attendue (obtenue par la formule directe). Pour chacune des valeurs où une différence est constatée, on surcharge le masque, puis on recalcule CT^g, et on passe au point M' suivant. Si la surcharge n'est pas possible (par exemple si on refuse les w_i rationnels), alors le masque est rejeté car il ne définit pas une norme. Au

```
    Procedure CalculeCT^g ( L, M_C^△, CT^g ) ;
1   CT^g[(0,0,0)] = 0 ;
2   for x = 1 to L − 1 , for y = 0 to x , for z = 0 to y do
3    {
4      min = +∞ ;
5      for each (v⃗,w) in M_C^△ do
6       {
7         (x′,y′,z′) = (x,y,z) − v⃗ ;
8         if (x′,y′,z′) ∈ (1/48)Z^3 and CT^g[(x′,y′,z′)] + w < min
9               then min = CT^g[(x′,y′,z′)] + w ;
10      }
11     CT^g[(x,y,z)] = min ;
12   }
```

FIG. 5.1. *Algorithme rapide de transformée de distance à l'origine dans* $\frac{1}{48}\mathbb{Z}^3$. ***Entrées*** *:* L *la largeur de l'image,* $\mathcal{M}_C^{\triangle}$ *le générateur complété du masque de* d_C. ***Sortie*** *:* CT^g *l'image de distance a l'origine, de taille* L^3.

contraire, si à la fin on a pu surcharger avec succès le masque, alors il définit bien une norme dans l'espace discret, et il est envisageable de calculer ses valeurs d'erreur.

5.4.3 Calcul de l'erreur

Le point de départ de notre calcul d'erreur est encore une fois constitué par la carte de distance à l'origine CT^g. En effet, il est parfaitement raisonnable en pratique d'estimer les erreurs MaxDiff et MAE en lisant directement les valeurs de n_C sur une série de quelques plans verticaux consécutifs, et ensuite de calculer les maxima correspondants aux formules (5.2) et (5.4) respectivement. On constate qu'il suffit de choisir 10 plans pour lesquels $x \geqslant 40$ pour que les points du réseau soient suffisamment denses vus depuis l'origine, et qu'on obtienne ainsi des valeurs équivalentes à celles données dans [Bor84], [Ver91b], [Bor96] et [Bor01] (pour ne citer que le cas 3D).

Dans l'évaluation de MaxDiff, on choisit toujours dans la suite le facteur d'échelle $s = w_{\mathsf{a}}$ afin que nos résultats soient comparables avec ceux de Borgefors.

Dans l'évaluation de MAE, on opère en deux temps. On balaye CT^g sur les plans choisis pour chercher le maximum $e_{\max}$ et le minimum $e_{\min}$ de l'erreur constatée en utilisant la formule (5.4). L'erreur maximum $e_{\max}$ sera donnée dans les tableaux qui suivent sous le nom de $\mathrm{MAE}_{\mathsf{a}}$ car elle reflète l'erreur commise lorsqu'on se limite à un facteur d'échelle entier $s = w_{\mathsf{a}}$. On calcule ensuite le facteur d'échelle optimal $s_{\,\mathrm{opt}}$ grâce à la formule suivante :

$$s_{\,\mathrm{opt}} = s\left(\frac{e_{\min} + e_{\max}}{2} + 1\right). \tag{5.7}$$

On calcule ensuite sur la base de ce nouveau facteur d'échelle optimal $s_{\,\mathrm{opt}}$, le taux d'erreur optimal correspondant $\mathrm{MAE}_{\mathrm{opt}}$ grâce à la formule

$$\mathrm{MAE}_{\mathrm{opt}} = \frac{w_{\mathsf{a}}}{s_{\,\mathrm{opt}}}(\mathrm{MAE}_{\mathsf{a}} + 1) - 1\,. \tag{5.8}$$

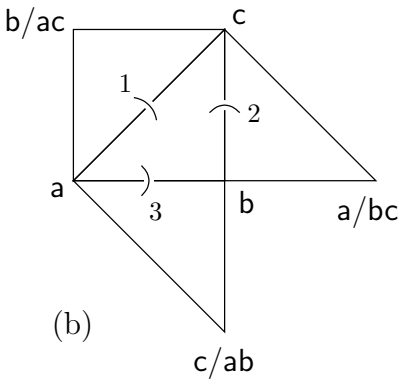
Ces formules de conversion sont tirées de [Thi94, p. 57-58] où elles sont justifiées.

5.5 *Application aux masques réguliers*

Nous allons maintenant présenter l'application de notre méthode d'optimisation à différents masques réguliers.

Le premier masque régulier choisi est le masque de taille 3 défini par le générateur $\mathcal{M}_C^g = \{(\overrightarrow{v_\mathsf{a}},w_\mathsf{a}),\ (\overrightarrow{v_\mathsf{b}},w_\mathsf{b}),\ (\overrightarrow{v_\mathsf{c}},w_\mathsf{c})\}$. Afin de faciliter la lecture, nous notons ce masque $\{\mathsf{a},\mathsf{b},\mathsf{c}\}$ en fonction des points visibles qui le constituent. Nous noterons de même les masques ultérieurs. Ce masque est le plus étudié en 3D car il est le plus simple à construire et le plus rapide lors de la transformation de distance. On le retrouve notamment dans [Bor84], dans [Ver91b] et dans [Bor96].

(a)



(b)

(c)

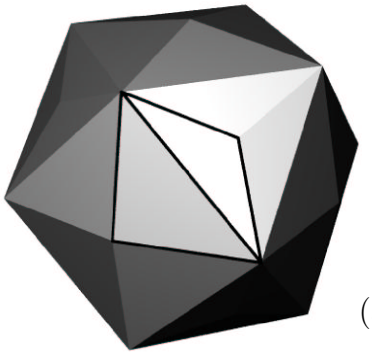
1	CCL(b/ac,a,b,c) : $w_\mathsf{a} + w_\mathsf{c} \leqslant 2.w_\mathsf{b}$
2	CCL(a,b,a/bc,c) : $w_\mathsf{b} \leqslant 2.w_\mathsf{a}$
3	CCL(c/ab,b,c,a) : $w_\mathsf{b} \leqslant w_\mathsf{c}$

$$w_\mathsf{b} \in [w_\mathsf{a} \ldots 2w_\mathsf{a}]$$
$$w_\mathsf{c} \in [w_\mathsf{b} \ldots 2w_\mathsf{b} - w_\mathsf{a}]$$

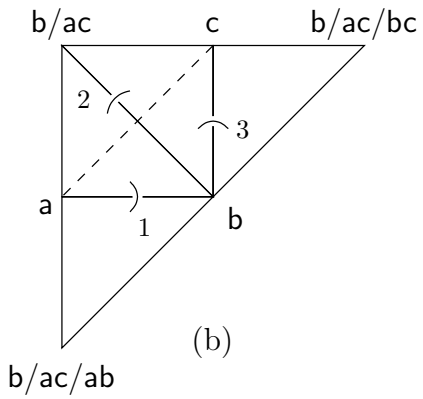
(d)

FIG. 5.2. *Triangulation naturelle du masque* $\{\mathsf{a},\mathsf{b},\mathsf{c}\}$ *représentée en 3D (a) et en projection (b), contraintes associées (c) et domaine de recherche (d).*

Il existe deux triangulations convexes possibles des points de ce masque ; la première d'entre elles est représentée à la figure 5.2 (a). Le générateur $\triangle$ de cette triangulation est représenté en projection en (b) ainsi que les contraintes de convexité locale correspondantes dont les expressions algébriques sont données en (c). Lorsqu'on utilise la valeur de la norme euclidienne de $\overrightarrow{v_i}$ pour pondérer chaque w_i où $i \in \{\mathsf{a},\mathsf{b},\mathsf{c}\}$, les contraintes de convexité locale sont vérifiées ; de ce fait, nous appelons cette triangulation du masque, une *triangulation naturelle*. Après manipulation, on tire des contraintes de convexité locale, les bornes de l'espace de solution de l'énumération qui sont données en (d). L'autre triangulation du masque $\{\mathsf{a},\mathsf{b},\mathsf{c}\}$, qui n'est pas une triangulation naturelle, est donnée à la figure 5.3 (a), ainsi que son générateur en projection (b), ses contraintes de convexité locale (c) et le domaine d'énumération correspondant (d).



(a)



(b)

(c)

1	CCL(b/ac/ab,b,b/ac,a)	: $w_{\mathsf{a}} \leqslant w_{\mathsf{b}}$
2	CCL(a,b,c,b/ac)	: $2.w_{\mathsf{b}} \leqslant w_{\mathsf{a}} + w_{\mathsf{c}}$
3	CCL(b/ac,b,b/ac/bc,c)	: $2.w_{\mathsf{c}} \leqslant 3.w_{\mathsf{b}}$

$$w_{\mathsf{b}} \in [w_{\mathsf{a}} \ldots 2w_{\mathsf{a}}]$$
$$w_{\mathsf{c}} \in [2w_{\mathsf{b}} - w_{\mathsf{a}} \ldots \tfrac{3}{2}w_{\mathsf{b}}]$$
(d)

FIG. 5.3. *Triangulation non-naturelle du masque* {a,b,c} *représentée en 3D (a) et en projection (b), contraintes associées (c) et domaine de recherche (d).*

L'énumération des masques répondant aux contraintes de la triangulation naturelle de {a,b,c} (cf. figure 5.2 (d)), combinée avec un calcul d'erreur systématique, permet d'obtenir le tableau de la figure 5.4, qui liste tous les masques optimaux pour $w_{\mathsf{a}} \in [1..30]$. La présence d'une étoile ($\star$) dans une colonne indique que la valeur d'erreur pour ce critère est plus faible que toutes celles précédemment observées ; le masque est donc optimal pour ce critère. Nous ne reprenons pas dans ce document les résultats de l'optimisation pour la triangulation non-naturelle, publiés dans [Rem00b], car ils sont systématiquement moins performants que ceux de la triangulation naturelle, et présentent de ce fait peu d'intérêt.

On remarque en particulier que $n_{3,4,5}$ réalise une bonne performance pour les différents critères et donc constitue un très bon choix si on désire que w_{a} reste assez petit, sinon on lui préférera sans doute $n_{26,37,45}$ avec $s = 27{,}661$ qui réalise le meilleur score pour $\mathrm{MAE}_{\mathrm{opt}}$. Enfin, il est important de signaler qu'au delà de $w_{\mathsf{a}} = 30$, l'amélioration de l'erreur devient de plus en plus faible et de ce fait, peu intéressante. Il est bien plus pertinent de subdiviser la triangulation afin de rendre le polyèdre de la boule rationnelle équivalente plus « ronde ».

Notre première idée fut d'étudier le masque {a,b,c,j} muni de la triangulation indiquée à la figure 5.5 (a). Cette triangulation est régulière ; par contre il n'existe aucun raffinement qui permette de la construire à partir de la triangulation naturelle de {a,b,c}.

Comme précédemment, on tire de la représentation en projection (b), six contraintes de convexité locale données en (c), et dont on tire des bornes d'énumération (d). On remarque cependant que les contraintes (c) ne permettent pas de tirer directement toutes les bornes d'énumération, comme cela arrive pour w_{b} par exemple. Comme vu au §5.4.1, on se sert du fait que si la triangulation de {a,b,c,j} est convexe, alors il faut que la triangulation de {a,b,c} qu'on obtient en fusionnant les facettes (a,b,j), (b,c,j) et (a,j,c), le soit aussi. On peut donc rajouter les bornes d'énumération plus faibles trouvées à la figure 5.2 (d), et on obtient finalement celles de la figure 5.5 (d).

w_{a}	w_{b}	w_{c}	MaxDiff ($\times M\%$)	$\mathrm{MAE}_{\mathsf{a}}$ (%) ($s = w_{\mathsf{a}}$)	$\mathrm{MAE}_{\mathrm{opt}}$ (% : s_{opt})
1	1	1	⋆ 73.20508	⋆ 42.26497	⋆ 26.79492 : 0.78868
1	2	2	⋆ 58.57864	⋆ 41.42136	⋆ 17.15729 : 1.20711
2	3	3	⋆ 23.20508	⋆ 13.39746	⋆ 12.70167 : 1.98406
2	3	4	29.28932	22.47449	⋆ 10.10205 : 2.22474
3	4	5	⋆ 11.80826	⋆ 10.55416	⋆ 7.94457 : 3.07253
4	6	7	17.08436	14.56439	⋆ 6.78789 : 4.29129
5	7	8	13.20508	⋆ 9.54451	8.50259 : 5.04801
7	9	11	16.06222	⋆ 9.27353	8.62534 : 6.95034
7	10	12	14.28571	12.48583	⋆ 6.38962 : 7.40111
8	11	13	⋆ 10.73214	9.68705	7.79712 : 8.14026
10	13	16	13.20508	⋆ 8.62780	8.32899 : 10.02758
11	16	19	15.20564	13.18091	⋆ 6.32055 : 11.70978
12	16	19	14.87175	⋆ 8.58621	8.47009 : 11.98478
12	17	21	15.42560	13.34486	⋆ 6.25506 : 12.80069
16	21	26	⋆ 10.70508	9.33023	8.17386 : 16.17104
17	22	27	14.38155	⋆ 8.49206	8.40667 : 16.98415
19	25	31	⋆ 10.52632	9.51922	8.13531 : 19.24316
19	27	33	14.97047	13.00479	⋆ 6.10540 : 20.23546
22	29	35	14.11417	⋆ 8.42524	8.27559 : 22.03041
26	37	45	14.77448	12.85693	⋆ 6.07705 : 27.66178
27	35	43	13.94582	⋆ 8.42430	8.37754 : 27.01165
27	36	44	⋆ 10.49574	9.49441	7.56833 : 27.48345
29	38	46	14.58439	⋆ 8.42030	8.35512 : 28.97937

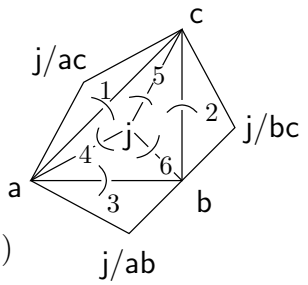
FIG. 5.4. *Résultat de l'énumération des masques* {a,b,c} *pour la triangulation naturelle et pour* $w_{\mathsf{a}} \in [1 \ldots 30]$ *; une étoile (⋆) indique que le masque est optimal pour le critère correspondant.*

Le tableau de la figure 5.6 donne les mesures d'erreur des masques {a,b,c,j}. On constate que l'amélioration apportée par la nouvelle pondération $(\overrightarrow{v_{\mathsf{j}}}, w_{\mathsf{j}})$ n'est pas très intéressante, en effet le maximum d'erreur pour les deux premiers critères a toujours lieu vers le milieu de l'arête (a,c), qui elle n'a pas été changée par l'introduction du point j. Pour améliorer significativement l'erreur, il est donc nécessaire de briser cette arête, par exemple en deux en rajoutant le point e.

La figure 5.7 montre la triangulation naturelle d'un masque {a,b,c,e}. Cette figure correspond au cas I présenté dans [Bor01]. Nous donnons en (c) les contraintes de convexité locale, dont on tire les bornes de l'énumération (d).

À la figure 5.8, on peut voir que les erreurs mesurées sont bien meilleures que précédemment, et ce pour les trois critères MaxDiff, $\mathrm{MAE}_{\mathsf{a}}$ et $\mathrm{MAE}_{\mathrm{opt}}$, ce qui prouve la pertinence du choix de e : la norme produite est bien plus proche de n_E que $n_{\mathsf{a,b,c}}$ avec seulement une pondération de plus dans le générateur. On retient donc par exemple $n_{3,4,5,\mathsf{e}=7}$, qui améliore

(a)



(b)

(c)

1	CCL(j/ac,a,j,c)	: $3.w_{\mathsf{a}} + 3.w_{\mathsf{c}} \leqslant 2.w_{\mathsf{j}}$
2	CCL(j/bc,c,j,b)	: $3.w_{\mathsf{b}} + 2.w_{\mathsf{c}} \leqslant 2.w_{\mathsf{j}}$
3	CCL(j/ab,b,j,a)	: $w_{\mathsf{a}} + 2.w_{\mathsf{b}} \leqslant w_{\mathsf{j}}$
4	CCL(b,j,c,a)	: $w_{\mathsf{j}} \leqslant w_{\mathsf{a}} + w_{\mathsf{b}} + w_{\mathsf{c}}$
5	CCL(a,j,b,c)	: $w_{\mathsf{j}} \leqslant w_{\mathsf{a}} + w_{\mathsf{b}} + w_{\mathsf{c}}$
6	CCL(a,b,c,j)	: $w_{\mathsf{j}} \leqslant w_{\mathsf{a}} + w_{\mathsf{b}} + w_{\mathsf{c}}$

(d)

$$w_{\mathsf{b}} \in [\, w_{\mathsf{a}} \ldots 2w_{\mathsf{a}} \,]$$
$$w_{\mathsf{c}} \in [\, w_{\mathsf{b}} \ldots 2w_{\mathsf{b}} - w_{\mathsf{a}} \,]$$
$$w_{\mathsf{j}} \in [\, \max\{w_{\mathsf{a}} + 2w_{\mathsf{b}}, (3w_{\mathsf{b}} + 2w_{\mathsf{c}})/2, (3w_{\mathsf{a}} + 3w_{\mathsf{c}})/2\} \ldots w_{\mathsf{a}} + w_{\mathsf{b}} + w_{\mathsf{c}} \,]$$

FIG. 5.5. *Triangulation du masque basé sur les points visibles* {a,b,c,j} *représentée en 3D (a) et en projection (b), contraintes associées (c) et domaine de recherche (d).*

$n_{3,4,5}$ sur les trois critères et est optimal pour $\mathrm{MAE}_{\mathsf{a}}$, même pour de plus grandes valeurs de w_{a}. Si la valeur de w_{a} importe peu, on peut aussi choisir $n_{24,34,42,\mathsf{e}=59}$ avec $s = 25{,}00961$ qui réalise un très bon taux de 4% d'erreur pour $\mathrm{MAE}_{\mathrm{opt}}$.

En suivant toujours la même tactique, on constate que le maximum d'erreur est mesuré vers le milieu de l'arête (a,b), et on construit en faisant intervenir le nouveau point visible d le masque de la figure 5.9 (a) et (b), noté masque {a,b,c,d,e}. Il y a cette fois-ci sept contraintes de convexité locale, données à la figure (c), et dont on tire le domaine de recherche (d).

Les résultats de l'optimisation (figure 5.11) sont une fois de plus meilleurs puisqu'on constate que le masque $n_{13,18,22,\mathsf{d}=29,\mathsf{e}=31}$ améliore sensiblement les valeurs trouvées pour les trois critères, et constitue un très bon choix pratique pour des applications.

On aimerait continuer sur le même principe à rajouter des points visibles dans le masque là où le maximum d'erreur est observé, en raffinant successivement une triangulation de Farey, tout en conservant une triangulation régulière qui soit naturelle ; cependant cela n'est plus possible à l'étape suivante, où il est nécessaire de faire intervenir le point f, qui se trouve au milieu de l'arête (a,c). En effet, alors que l'incorporation des points e (médian de a et c) et d (médian de a et b) a toujours permis de produire une triangulation de Farey naturelle, l'incorporation de f, médian de a et c, conduit à la triangulation de la figure 5.10 (a), qui n'est pas la triangulation naturelle du masque donnée à la figure 5.10 (b). Or, plus on choisit un w_{a} grand, plus les pondérations du masque se rapprochent (en proportion)

w_a	w_b	w_c	w_j	MaxDiff ($\times M\%$)	MAE$_\mathsf{a}$ (%) ($s = w_\mathsf{a}$)	MAE$_{\text{opt}}$ (% : s_{opt})
1	1	1	3	⋆ 73.20508	⋆ 42.26497	⋆ 26.79492 : 0.78868
1	2	2	5	⋆ 58.57864	⋆ 41.42136	⋆ 17.15729 : 1.20711
2	3	3	8	⋆ 23.20508	⋆ 13.39746	⋆ 12.70167 : 1.98406
2	3	4	9	29.28927	22.47449	⋆ 10.10205 : 2.22474
3	4	5	12	⋆ 11.80769	⋆ 10.55416	⋆ 7.94457 : 3.07253
4	6	7	17	17.08405	14.56439	⋆ 6.78789 : 4.29129
5	7	8	20	13.20508	⋆ 9.54451	8.50259 : 5.04801
7	9	11	27	16.06222	⋆ 9.27353	8.62534 : 6.95034
7	10	12	29	14.28571	12.48583	⋆ 6.38962 : 7.40111
8	11	13	32	⋆ 10.72846	9.68705	7.79712 : 8.14026
8	12	14	33	15.21739	13.19231	⋆ 6.18799 : 8.52769
10	13	16	39	13.20508	⋆ 8.62780	8.32899 : 10.02758
11	16	19	45	14.23522	12.44834	⋆ 5.99710 : 11.66949
12	16	19	47	14.87175	⋆ 8.58621	8.47048 : 11.98483
15	22	26	62	14.62169	12.73291	⋆ 5.98540 : 15.95497
16	21	26	63	⋆ 10.70508	9.32789	8.17279 : 16.17085
17	22	27	66	14.38155	⋆ 8.49206	8.40645 : 16.98411
17	23	27	66	14.38155	⋆ 8.30513	8.30408 : 17.00016
19	25	31	75	⋆ 10.52570	9.51879	8.13512 : 19.24312
26	37	45	107	14.42767	12.59272	⋆ 5.96028 : 27.62743
27	36	44	107	⋆ 10.49553	9.49350	7.56792 : 27.48333
30	43	52	123	14.49496	12.64432	⋆ 5.94623 : 31.89665

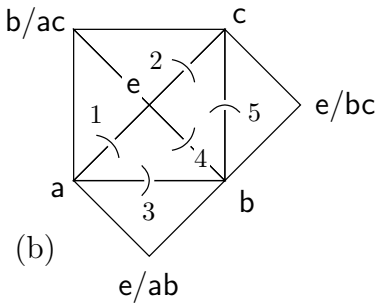
FIG. 5.6. *Résultat de l'énumération des masques* $\{\mathsf{a},\mathsf{b},\mathsf{c},\mathsf{j}\}$ *pour* $w_\mathsf{a} \in [1 \ldots 30]$, *une étoile* ($\star$) *indique que le masque est optimal pour le critère correspondant.*

de leur valeur euclidienne, et donc plus la triangulation se rapproche de la triangulation naturelle. On constate en pratique que si on choisit d'optimiser la triangulation de Farey (a), on obtient des taux d'erreur plus importants que si on choisit d'optimiser la triangulation (b), car il n'est pas possible de satisfaire correctement toutes les contraintes en même temps.

D'autre part, quand on cherche à raffiner la triangulation d'origine par la méthode des raffinement de Farey décrite au §4.4.2, on obtient des triangles de plus en plus étirés et longs, ce qui est souvent dommageable pour l'optimisation des normes : il vaux mieux que les facettes soient petites et uniformément réparties sur la surface de la boule, afin que le polyèdre formant la boule soit le plus « rond » possible.

Pour toutes ces raisons, nous choisissons d'orienter maintenant notre présentation sur le calcul des masques non-réguliers.

(a)



(b)

(c)

1	CCL(b/ac,a,b,e) : $w_{\mathsf{e}} \leqslant 2.w_{\mathsf{b}}$
2	CCL(b/ac,e,b,c) : $w_{\mathsf{e}} \leqslant 2.w_{\mathsf{b}}$
3	CCL(e/ab,b,e,a) : $w_{\mathsf{a}} + w_{\mathsf{b}} \leqslant w_{\mathsf{e}}$
4	CCL(a,b,c,e) : $w_{\mathsf{e}} \leqslant w_{\mathsf{a}} + w_{\mathsf{c}}$
5	CCL(e,b,e/bc,c) : $w_{\mathsf{b}} + 2.w_{\mathsf{c}} \leqslant 2.w_{\mathsf{e}}$

(d)

$$w_{\mathsf{b}} \in [\, w_{\mathsf{a}} \ldots 2w_{\mathsf{a}} \,]$$
$$w_{\mathsf{c}} \in [\, w_{\mathsf{b}} \ldots w_{\mathsf{a}} + w_{\mathsf{b}} \,]$$
$$w_{\mathsf{e}} \in [\, \max\{(w_{\mathsf{b}} + 2w_{\mathsf{c}})/2, w_{\mathsf{a}} + w_{\mathsf{b}}\} \ldots \min\{2w_{\mathsf{b}}, w_{\mathsf{a}} + w_{\mathsf{c}}\} \,]$$

FIG. 5.7. *Triangulation du masque basé sur les points visibles* {a,b,c,e} *représentée en 3D (a) et en projection (b), contraintes associées (c) et domaine de recherche (d).*

5.6 Application aux masques non-réguliers

La figure 5.12 reprend la triangulation déjà présentée à la figure 5.10 (b). Comme on l'a déjà dit, cette triangulation n'est pas régulière puisque $\Delta_{\mathsf{d,f,e}} = 2$; le point $\mathsf{j} = (3,2,1)$ appartient au parallélépipède défini par les vecteurs $\overrightarrow{v_{\mathsf{d}}}$, $\overrightarrow{v_{\mathsf{e}}}$ et $\overrightarrow{v_{\mathsf{f}}}$ sans être atteint par une combinaison linéaire entière de ces vecteurs. Il est donc nécessaire après la détermination des neuf contraintes de convexité locale (figure 5.12 c) et de l'espace de recherche (d) de procéder à la surcharge du masque en rajoutant quand cela est nécessaire (ici tout le temps) une nouvelle pondération $(\overrightarrow{v_{\mathsf{j}}}, w_{\mathsf{j}})$ où $w_{\mathsf{j}} = (w_{\mathsf{d}} + w_{\mathsf{e}} + w_{\mathsf{f}})/2$.

Le tableau de la figure 5.13 donne le résultat de l'énumération, de la surcharge (colonne w_{j}) et de l'estimation d'erreur pour tous les masques où w_{j} est entier. On constate une nouvelle amélioration de l'erreur, même si elle est moins importante pour MaxDiff et $\mathrm{MAE}_{\mathsf{a}}$ que pour $\mathrm{MAE}_{\mathrm{opt}}$, qui tire parti de l'aspect plus uniforme de la triangulation sur la surface de la boule.

À la figure 5.14 enfin, nous présentons une triangulation encore plus subdivisée ainsi que les douze contraintes de convexité locale correspondantes et le domaine de recherche qui en découle.

L'opération de surcharge devient ici conséquente et assez lourde puisqu'il existe dix-huit points à vérifier ; leur liste est donnée à la figure 5.15. Une fois ces points surchargés si nécessaire, il est possible d'effectuer le calcul d'erreur. Les masques optimaux obtenus sont donnés à la figure 5.16 tandis que la valeur des surcharges nécessaires pour chaque masque est donnée à la figure 5.17. Une fois de plus les taux d'erreur font un bond en avant puisqu'il sont presque divisés par deux.

w_a	w_b	w_c	w_e	MaxDiff ($\times M\%$)	MAE$_\mathsf{a}$ (%) ($s = w_\mathsf{a}$)	MAE$_{\text{opt}}$ (%: s_{opt})
1	1	1	2	$\star$ 73.20508	$\star$ 42.26497	$\star$ 26.79492 : 0.78868
1	2	2	3	$\star$ 58.57864	$\star$ 41.42136	$\star$ 17.15729 : 1.20711
2	2	3	4	$\star$ 41.42136	$\star$ 29.28932	17.15729 : 1.70711
2	3	3	5	$\star$ 23.20508	$\star$ 13.39746	$\star$ 12.70167 : 1.98406
2	3	4	6	29.28932	22.47449	$\star$ 10.10205 : 2.22474
3	4	5	7	$\star$ 8.08802	$\star$ 5.71910	$\star$ 5.57281 : 2.99535
5	7	9	13	10.71796	9.54451	$\star$ 5.05878 : 5.21349
7	10	12	17	9.64921	8.79676	$\star$ 4.72752 : 7.27199
7	10	13	18	15.08176	9.73065	$\star$ 4.63960 : 7.34057
10	14	18	25	10.38516	8.16654	$\star$ 4.42727 : 10.35807
11	15	19	27	$\star$ 7.29055	6.79400	5.10307 : 11.17697
12	17	21	30	9.47655	8.65265	$\star$ 4.14692 : 12.51916
14	19	24	34	$\star$ 6.86848	6.42617	5.16905 : 14.16735
17	23	29	41	$\star$ 6.61994	6.20865	5.22165 : 17.15946
17	24	30	42	9.05974	8.30560	$\star$ 4.07373 : 17.69126
20	27	34	48	$\star$ 6.45854	6.06566	5.26299 : 20.15251
24	34	42	59	9.18842	8.41286	$\star$ 4.03663 : 25.00954

FIG. 5.8. *Résultat de l'énumération des masques* $\{\mathsf{a},\mathsf{b},\mathsf{c},\mathsf{e}\}$ *pour* $w_\mathsf{a} \in [1 \ldots 30]$, *une étoile* ($\star$) *indique que le masque est optimal pour le critère correspondant.*

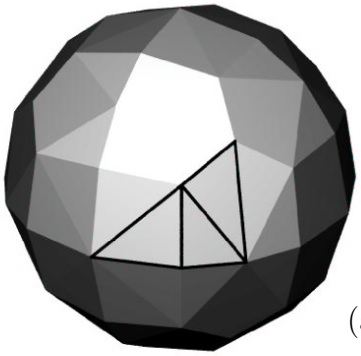
Il est tout à fait possible de continuer à subdiviser le masque sur le même principe afin de réaliser des masques de taille 9 ou plus pour diminuer encore l'erreur commise, cependant la quantité de surcharges devient vite très importante (plus les cônes sont fins, plus ils contiennent de points à surcharger), induisant un sur-coût important de calcul pour la transformation de distance.

5.7 Conclusion

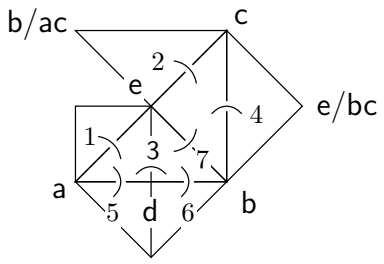
Dans ce chapitre, nous avons présenté la méthode d'optimisation de Borgefors ainsi que celle de Verwer et de Thiel. Nous avons également exposé trois critères de comparaison des distances de chanfrein avec la distance euclidienne. Puis nous avons détaillé notre méthode d'optimisation et justifié les choix qui nous poussaient à privilégier cette méthode plutôt que celles déjà existantes : nous choisissons un calcul systématique de l'erreur pour chaque masque afin de déterminer les meilleurs masques possibles *en pratique*.

Notre méthode s'articule sur trois étapes indépendantes : premièrement l'énumération des masques vérifiant les contraintes de convexité locale, ensuite pour les masques non-réguliers, la surcharge du masque, et enfin, le calcul effectif de l'erreur grâce à la carte de distance à l'origine CT^g.

Finalement en application de ce procédé, nous présentons une série de masques opti-



(a)



(b) e/ab

(c)

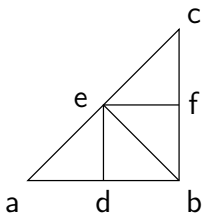
1	CCL(d/ac,a,d,e)	: $2.w_{\mathsf{a}} + w_{\mathsf{e}} \leqslant 2.w_{\mathsf{d}}$
2	CCL(b/ac,e,b,c)	: $w_{\mathsf{e}} \leqslant 2.w_{\mathsf{b}}$
3	CCL(a,d,b,e)	: $w_{\mathsf{d}} \leqslant w_{\mathsf{a}} + w_{\mathsf{b}}$
4	CCL(e/bc,c,e,b)	: $w_{\mathsf{b}} + 2.w_{\mathsf{c}} \leqslant 2.w_{\mathsf{e}}$
5	CCL(e/ab,d,e,d)	: $w_{\mathsf{d}} \leqslant w_{\mathsf{e}}$
6	CCL(e/ab,b,e,d)	: $w_{\mathsf{d}} \leqslant w_{\mathsf{e}}$
7	CCL(d,b,c,e)	: $w_{\mathsf{b}} + w_{\mathsf{e}} \leqslant w_{\mathsf{c}} + w_{\mathsf{d}}$

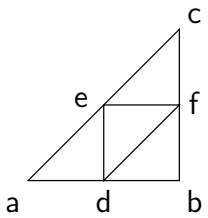
$$
\begin{aligned}
&w_{\mathsf{b}} \in [\, w_{\mathsf{a}} \ldots 2w_{\mathsf{a}} \,] \\
&w_{\mathsf{c}} \in [\, w_{\mathsf{b}} \ldots 5w_{\mathsf{b}}/2 \,] \\
&w_{\mathsf{d}} \in [\, 2w_{\mathsf{a}} \ldots \min\{w_{\mathsf{a}} + w_{\mathsf{b}}, 2w_{\mathsf{b}}\} \,] \\
&w_{\mathsf{e}} \in [\, \max\{(w_{\mathsf{b}} + 2w_{\mathsf{c}})/2, w_{\mathsf{d}}\} \\
&\qquad \ldots \min\{2w_{\mathsf{b}}, 2w_{\mathsf{d}} - 2w_{\mathsf{a}}, \\
&\qquad\quad w_{\mathsf{a}} + w_{\mathsf{c}}, w_{\mathsf{c}} + w_{\mathsf{d}} - w_{\mathsf{b}}\} \,]
\end{aligned}
$$

(d)

FIG. 5.9. *Triangulation du masque basé sur les points visibles* {a,b,c,d,e} *représentée en 3D (a) et en projection (b), contraintes associées (c) et domaine de recherche (d).*

(a)





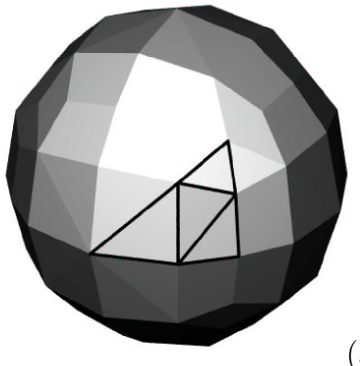
(b)

FIG. 5.10. *La triangulation de Farey raffinée (a) peut pas correspondre à la triangulation naturelle (b) recherchée.*

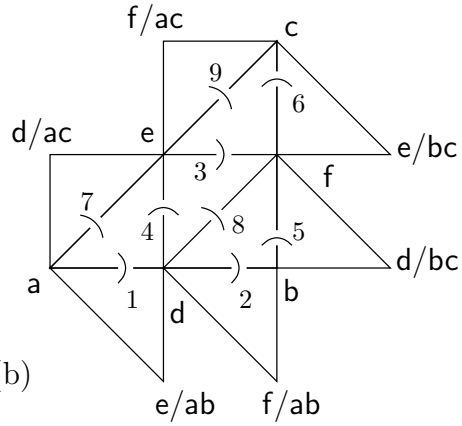
maux basés sur des triangulations régulières et non-régulières, ainsi que les raisons qui à nos yeux peuvent les rendre pertinents dans des applications : efficacité en coût de calcul et uniformité de la triangulation. Pour chaque masque optimal, les valeurs des distances locales w_i, du facteur d'échelle s et des taux d'erreur (MaxDiff, MAE) sont données. Les résultats concernant les masques {a,b,c,j}, {a,b,c,d,e}, {a,b,c,d,e,f} et {a,b,c,g,h,i,j,k,l,m} sont nouveaux.

w_{a}	w_{b}	w_{c}	w_{d}	w_{e}	MaxDiff ($\times M\%$)	$\mathrm{MAE}_{\mathsf{a}}$ (%) ($s = w_{\mathsf{a}}$)	$\mathrm{MAE}_{\mathrm{opt}}$ (% : s_{opt})
1	1	1	2	2	⋆ 73.20508	⋆ 42.26497	⋆ 26.79492 : 0.78868
1	2	2	3	3	⋆ 58.57864	⋆ 41.42136	⋆ 17.15729 : 1.20711
2	2	3	4	4	⋆ 41.42136	⋆ 29.28932	⋆ 17.15729 : 1.70711
2	3	3	5	5	⋆ 23.20508	⋆ 13.39746	⋆ 12.70167 : 1.98406
2	3	4	5	6	29.28927	22.47449	⋆ 10.10205 : 2.22474
3	4	5	7	7	⋆ 8.08802	⋆ 5.71910	⋆ 5.57281 : 2.99535
4	6	7	9	10	13.06929	8.97247	⋆ 4.29362 : 4.17945
6	8	10	13	14	⋆ 8.08802	⋆ 5.71910	⋆ 4.29362 : 5.91063
6	8	10	14	14	⋆ 8.08802	⋆ 5.71910	⋆ 5.57281 : 5.99070
6	8	10	14	15	⋆ 8.08802	6.71874	6.18799 : 6.02999
7	10	12	16	17	⋆ 7.33019	⋆ 4.97813	⋆ 2.94373 : 7.13834
8	11	14	18	20	⋆ 6.53437	6.06602	4.34783 : 8.13173
9	13	15	20	22	6.62588	⋆ 4.82201	⋆ 4.27610 : 9.04712
10	14	17	22	24	⋆ 5.09178	⋆ 3.92305	⋆ 2.94373 : 10.09513
12	17	21	27	30	8.33333	6.06602	⋆ 2.94373 : 12.36396
13	18	22	29	31	⋆ 3.97431	⋆ 3.76650	⋆ 3.18966 : 13.07267
16	22	27	35	38	4.45508	⋆ 3.45511	⋆ 3.24136 : 16.03313
17	24	29	38	41	6.01029	4.23113	⋆ 2.84731 : 17.22874
19	26	32	42	45	4.78403	⋆ 3.40074	⋆ 3.35365 : 19.00866
19	27	33	43	47	7.91962	5.52354	⋆ 2.68755 : 19.52474
22	31	38	49	54	6.83737	5.03836	⋆ 2.65421 : 22.51095
26	36	44	57	62	⋆ 3.97431	3.63196	3.12485 : 26.12785
26	36	44	58	62	⋆ 3.97431	3.76650	3.18966 : 26.14534
26	37	45	59	64	7.74723	5.32550	⋆ 2.63067 : 26.68270

FIG. 5.11. *Résultat de l'énumération des masques* $\{\mathsf{a},\mathsf{b},\mathsf{c},\mathsf{d},\mathsf{e}\}$ *pour* $w_{\mathsf{a}} \in [1 \ldots 30]$, *une étoile* ($\star$) *indique que le masque est optimal pour le critère correspondant.*



(a)



(b)

(c)

1	CCL(d/ac,a,d,e)	: $2.w_{\mathsf{a}} + w_{\mathsf{e}} \leqslant 2.w_{\mathsf{d}}$
2	CCL(f/ac,e,f,c)	: $2.w_{\mathsf{c}} + w_{\mathsf{e}} \leqslant 2.w_{\mathsf{f}}$
3	CCL(e,d,b,f)	: $w_{\mathsf{d}} + w_{\mathsf{f}} \leqslant 2.w_{\mathsf{b}} + w_{\mathsf{e}}$
4	CCL(a,d,f,e)	: $w_{\mathsf{d}} + w_{\mathsf{e}} \leqslant 2.w_{\mathsf{a}} + w_{\mathsf{f}}$
5	CCL(e/bc,c,e,f)	: $w_{\mathsf{c}} + w_{\mathsf{f}} \leqslant 2.w_{\mathsf{e}}$
6	CCL(d/bc,f,d,b)	: $3.w_{\mathsf{b}} \leqslant 2.w_{\mathsf{d}}$
7	CCL(e/ab,d,e,a)	: $w_{\mathsf{d}} \leqslant w_{\mathsf{e}}$
8	CCL(f/ab,b,f,d)	: $2.w_{\mathsf{b}} \leqslant w_{\mathsf{f}}$
9	CCL(d,f,c,e)	: $w_{\mathsf{e}} + w_{\mathsf{f}} \leqslant 2.w_{\mathsf{c}} + w_{\mathsf{d}}$

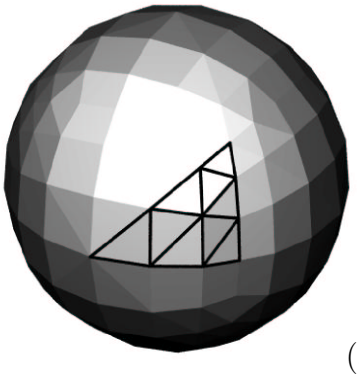
(d)

$$
\begin{aligned}
&w_{\mathsf{b}} \in [\, w_{\mathsf{a}} \ldots 2w_{\mathsf{a}} \,] \\
&w_{\mathsf{c}} \in [\, w_{\mathsf{b}} \ldots w_{\mathsf{a}} + w_{\mathsf{b}} \,] \\
&w_{\mathsf{d}} \in [\, 3w_{\mathsf{a}}/2 \ldots w_{\mathsf{a}} + w_{\mathsf{b}} \,] \\
&w_{\mathsf{e}} \in [\, w_{\mathsf{e}} \ldots 2(w_{\mathsf{c}} - w_{\mathsf{b}}) + w_{\mathsf{d}} \,] \\
&w_{\mathsf{f}} \in [\, \max\{(2w_{\mathsf{c}} + w_{\mathsf{e}})/2, w_{\mathsf{d}} + w_{\mathsf{e}} - 2w_{\mathsf{a}}, 2w_{\mathsf{b}}\} \\
&\qquad \ldots \min\{3w_{\mathsf{c}} + w_{\mathsf{d}} - w_{\mathsf{e}}, 2w_{\mathsf{e}} - w_{\mathsf{c}}, 2w_{\mathsf{b}} + w_{\mathsf{e}} - w_{\mathsf{d}}\} \,]
\end{aligned}
$$

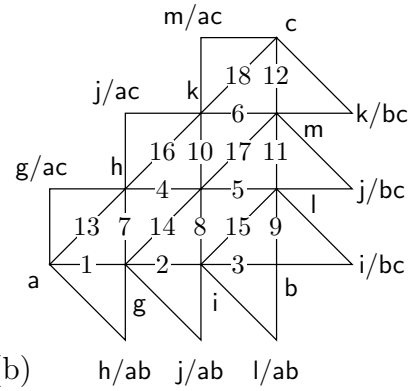
FIG. 5.12. *Triangulation du masque basé sur les points visibles* {a,b,c,d,e,f} *représentée en 3D (a) et en projection (b), contraintes associées (c) et domaine de recherche (d).*

w_{a}	w_{b}	w_{c}	w_{d}	w_{e}	w_{f}	w_{j}	MaxDiff ($\times M\%$)	MAE$_{\mathsf{a}}$ (%) ($s = w_{\mathsf{a}}$)	MAE$_{\text{opt}}$ (%: s_{opt})
1	1	1	2	2	2	3	⋆ 73.20508	⋆ 42.26497	⋆ 26.79492 : 0.78868
1	2	2	3	3	4	5	⋆ 58.57864	⋆ 41.42136	⋆ 17.15729 : 1.20711
2	3	3	5	5	6	8	⋆ 23.20508	⋆ 13.39746	⋆ 12.70167 : 1.98406
2	3	4	5	6	7	9	29.28932	22.47449	⋆ 10.10205 : 2.22474
3	4	5	7	8	9	12	⋆ 11.80826	⋆ 10.55416	⋆ 7.94457 : 3.07253
4	6	7	9	10	13	16	13.06936	⋆ 8.97247	⋆ 4.29362 : 4.17945
5	7	8	11	12	15	19	13.20508	⋆ 7.62396	5.88235 : 4.90748
5	7	9	11	12	15	19	⋆ 6.79492	⋆ 3.92305	⋆ 2.94373 : 5.04757
7	10	12	16	17	21	27	⋆ 5.23928	4.97813	⋆ 2.94373 : 7.13834
9	13	16	20	22	28	35	6.64717	4.82201	⋆ 2.66461 : 9.18913
10	14	17	22	24	30	38	⋆ 4.08329	⋆ 3.92305	2.94373 : 10.09513
11	15	19	24	26	32	41	5.05772	⋆ 3.57635	3.42015 : 10.98221
11	16	19	24	26	32	41	4.54545	⋆ 3.50495	3.38318 : 10.98614
11	16	20	25	27	34	43	8.74177	5.23488	⋆ 2.55068 : 11.28792
13	18	22	29	31	38	49	⋆ 3.97431	3.77490	3.19371 : 13.07322
16	22	27	35	38	47	60	4.45508	⋆ 3.45581	3.24170 : 16.03318
16	22	28	35	38	47	60	⋆ 3.92136	3.45581	3.24170 : 16.03318
16	23	28	35	38	47	60	⋆ 3.72449	3.45581	3.24170 : 16.03318
19	27	32	42	45	57	72	4.78403	⋆ 3.40465	3.35554 : 19.00903
20	29	35	45	49	60	77	5.26349	4.99941	⋆ 2.43875 : 20.49994
21	29	36	46	50	62	79	⋆ 3.69627	3.56453	3.16913 : 21.08048
27	37	46	59	64	79	101	4.38432	⋆ 3.37249	3.29891 : 27.01923
29	40	50	64	69	85	109	⋆ 3.69487	3.56315	3.20293 : 29.10122
29	41	51	65	71	88	112	5.18809	4.93189	⋆ 2.43137 : 29.70794

FIG. 5.13. *Résultat de l'énumération des masques* $\{\mathsf{a},\mathsf{b},\mathsf{c},\mathsf{d},\mathsf{e},\mathsf{f}\}$ *pour* $w_{\mathsf{a}} \in [1 \ldots 30]$, *une étoile* ($\star$) *indique que le masque est optimal pour le critère correspondant. La colonne* w_{j} *indique la valeur de la seule surcharge nécessaire pour ce masque au point* (3,2,1).



(a)



(b)

(c)

1	CCL(h/ab,g,h,a)	$: w_{\mathsf{g}} \leqslant w_{\mathsf{h}}$
2	CCL(j/ab,i,j,g)	$: w_{\mathsf{i}} \leqslant w_{\mathsf{j}}$
3	CCL(l/ab,b,l,i)	$: 3.w_{\mathsf{b}} \leqslant w_{\mathsf{l}}$
4	CCL(g,j,k,h)	$: w_{\mathsf{h}} + w_{\mathsf{j}} \leqslant w_{\mathsf{g}} + w_{\mathsf{k}}$
5	CCL(i,l,m,j)	$: w_{\mathsf{j}} + w_{\mathsf{l}} \leqslant w_{\mathsf{i}} + w_{\mathsf{m}}$
6	CCL(j,m,c,k)	$: w_{\mathsf{k}} + w_{\mathsf{m}} \leqslant 3.w_{\mathsf{c}} + w_{\mathsf{j}}$
7	CCL(a,g,j,h)	$: w_{\mathsf{g}} + w_{\mathsf{h}} \leqslant 3.w_{\mathsf{a}} + w_{\mathsf{j}}$
8	CCL(g,i,l,j)	$: w_{\mathsf{i}} + w_{\mathsf{j}} \leqslant w_{\mathsf{g}} + w_{\mathsf{l}}$
9	CCL(i/bc,l,i,b)	$: 5.w_{\mathsf{b}} \leqslant 2.w_{\mathsf{i}}$
10	CCL(h,j,m,k)	$: w_{\mathsf{j}} + w_{\mathsf{k}} \leqslant w_{\mathsf{h}} + w_{\mathsf{m}}$
11	CCL(j/bc,m,j,l)	$: 4.w_{\mathsf{l}} + w_{\mathsf{m}} \leqslant 6.w_{\mathsf{j}}$
12	CCL(k/bc,c,k,m)	$: 2.w_{\mathsf{c}} + w_{\mathsf{m}} \leqslant 2.w_{\mathsf{k}}$
13	CCL(g/ac,a,g,h)	$: 3.w_{\mathsf{a}} + w_{\mathsf{h}} \leqslant 2.w_{\mathsf{g}}$
14	CCL(h,g,i,j)	$: w_{\mathsf{g}} + w_{\mathsf{j}} \leqslant w_{\mathsf{h}} + w_{\mathsf{i}}$
15	CCL(j,i,b,l)	$: w_{\mathsf{i}} + w_{\mathsf{l}} \leqslant 3.w_{\mathsf{b}} + w_{\mathsf{j}}$
16	CCL(j/ac,h,j,k)	$: w_{\mathsf{h}} + w_{\mathsf{k}} \leqslant 2.w_{\mathsf{j}}$
17	CCL(k,j,l,m)	$: w_{\mathsf{j}} + w_{\mathsf{m}} \leqslant w_{\mathsf{k}} + w_{\mathsf{l}}$
18	CCL(m/ac,k,m,c)	$: 3.w_{\mathsf{c}} + w_{\mathsf{k}} \leqslant 2.w_{\mathsf{m}}$

(d)

$$
\begin{aligned}
w_{\mathsf{b}} &\in [\, w_{\mathsf{a}} \ldots 2w_{\mathsf{a}} \,] \\
w_{\mathsf{c}} &\in [\, w_{\mathsf{b}} \ldots 2w_{\mathsf{b}} - w_{\mathsf{a}} \,] \\
w_{\mathsf{g}} &\in [\, \max\{3w_{\mathsf{a}}, 2w_{\mathsf{b}}, w_{\mathsf{b}} - 6w_{\mathsf{a}}\} \ldots \min\{2w_{\mathsf{a}} + w_{\mathsf{b}}, 3w_{\mathsf{a}} + w_{\mathsf{i}} \,] \\
w_{\mathsf{h}} &\in [\, w_{\mathsf{g}} \ldots 2w_{\mathsf{g}} - 3w_{\mathsf{a}} \,] \\
w_{\mathsf{i}} &\in [\, \max\{2w_{\mathsf{b}} - 3w_{\mathsf{a}}, 5w_{\mathsf{b}}/2\} \ldots (3w_{\mathsf{b}} + w_{\mathsf{g}})/2 \,] \\
w_{\mathsf{j}} &\in [\, \max\{w_{\mathsf{i}}, 2w_{\mathsf{h}} - w_{\mathsf{g}}, w_{\mathsf{g}} + w_{\mathsf{h}} - 3w_{\mathsf{a}}\} \ldots w_{\mathsf{h}} + w_{\mathsf{i}} - w_{\mathsf{g}} \,] \\
w_{\mathsf{k}} &\in [\, \max\{7w_{\mathsf{c}}/3, w_{\mathsf{h}} + w_{\mathsf{j}} - w_{\mathsf{g}}\} \ldots \min\{2w_{\mathsf{j}} - w_{\mathsf{h}}, 2(w_{\mathsf{c}} + w_{\mathsf{j}})/3\} \,] \\
w_{\mathsf{l}} &\in [\, \max\{3w_{\mathsf{b}}, w_{\mathsf{i}} + w_{\mathsf{j}} - w_{\mathsf{g}} \ldots 3w_{\mathsf{b}} + w_{\mathsf{j}} - w_{\mathsf{i}} \,] \\
w_{\mathsf{m}} &\in [\, \max\{(3w_{\mathsf{c}} + w_{\mathsf{k}})/2, w_{\mathsf{j}} + w_{\mathsf{k}} - 2w_{\mathsf{h}}, w_{\mathsf{j}} + w_{\mathsf{l}} - w_{\mathsf{i}}\} \\
&\qquad \ldots \min\{3w_{\mathsf{c}} + w_{\mathsf{j}} - w_{\mathsf{k}}, 6w_{\mathsf{j}} - 4w_{\mathsf{l}}, 2w_{\mathsf{k}} - 2w_{\mathsf{c}}, w_{\mathsf{k}} + w_{\mathsf{l}} - w_{\mathsf{j}}\} \,]
\end{aligned}
$$

FIG. 5.14. *Triangulation du masque basé sur les points visibles* {a,b,c,g,h,i,j,k,l,m} *représentée en 3D (a) et en projection (b), contraintes associées (c) et domaine de recherche (d).*

triangle	point	valeur de la surcharge
(g,j,h)	(4,2,1)	$(w_{\mathsf{g}} + 2w_{\mathsf{j}} + w_{\mathsf{h}})/3$
	(5,2,1)	$(2w_{\mathsf{g}} + w_{\mathsf{j}} + 2w_{\mathsf{h}})/3$
(g,i,j)	(2,1,0)	$(w_{\mathsf{g}} + w_{\mathsf{i}})/3$
	(4,2,0)	$(2w_{\mathsf{g}} + 2w_{\mathsf{i}})/3$
	(5,3,1)	$(w_{\mathsf{g}} + 2w_{\mathsf{i}} + 3w_{\mathsf{j}})/3$
	(7,4,1)	$(2w_{\mathsf{g}} + 2w_{\mathsf{i}} + 3w_{\mathsf{j}})/3$
(i,l,j)	(4,3,1)	$(w_{\mathsf{i}} + w_{\mathsf{l}} + 2w_{\mathsf{j}})/3$
	(5,4,1)	$(2w_{\mathsf{i}} + 2w_{\mathsf{l}} + w_{\mathsf{j}})/3$
(h,j,k)	(2,1,1)	$(w_{\mathsf{h}} + w_{\mathsf{k}})/3$
	(4,2,2)	$(2w_{\mathsf{h}} + 2w_{\mathsf{k}})/3$
	(5,3,2)	$(w_{\mathsf{h}} + 3w_{\mathsf{j}} + w_{\mathsf{k}})/3$
	(7,4,3)	$(2w_{\mathsf{h}} + 3w_{\mathsf{j}} + 2w_{\mathsf{k}})/3$
(j,m,k)	(4,3,2)	$(2w_{\mathsf{j}} + w_{\mathsf{m}} + w_{\mathsf{k}})/3$
	(5,4,3)	$(w_{\mathsf{j}} + 2w_{\mathsf{m}} + 2w_{\mathsf{k}})/3$
(j,l,m)	(2,2,1)	$(w_{\mathsf{l}} + w_{\mathsf{m}})/3$
	(4,4,2)	$(2w_{\mathsf{l}} + 2w_{\mathsf{m}})/3$
	(5,4,2)	$(3w_{\mathsf{j}} + w_{\mathsf{l}} + w_{\mathsf{m}})/3$
	(7,6,3)	$(3w_{\mathsf{j}} + 2w_{\mathsf{l}} + 2w_{\mathsf{m}})/3$

FIG. 5.15. *Liste pour chaque triangle de la triangulation* $\triangle$ *des masques* $\{\mathsf{a,b,c,g,h,i,j,k,l,m}\}$, *des points de surcharge avec pour chacun d'eux, l'expression littérale de sa valeur de surcharge.*

w_{a}	w_{b}	w_{c}	w_{g}	w_{h}	w_{i}	w_{j}	w_{k}	w_{l}	w_{m}	MaxDiff ($\times M\%$)	$\mathrm{MAE}_{\mathsf{a}}$ (%) ($s = w_{\mathsf{a}}$)	$\mathrm{MAE}_{\mathrm{opt}}$ (% : s_{opt})
1	1	1	3	3	3	3	3	3	3	⋆ 73.20508	⋆ 42.26497	⋆ 26.79492 : 0.78868
1	2	2	4	4	5	5	5	6	6	⋆ 58.57864	⋆ 41.42136	⋆ 17.15729 : 1.20711
2	3	3	7	7	8	8	8	9	9	⋆ 23.20508	⋆ 13.39746	⋆ 12.70167 : 1.98406
2	3	4	7	8	8	9	10	10	11	29.28932	22.47449	⋆ 10.10205 : 2.22474
3	4	5	10	11	11	12	13	13	14	⋆ 11.80826	⋆ 10.55416	⋆ 7.94457 : 3.07253
4	6	7	14	15	16	17	18	19	20	17.08436	14.56439	⋆ 6.78789 : 4.29129
5	7	8	17	18	19	20	21	22	23	13.20508	⋆ 9.54451	8.50259 : 5.04801
5	7	9	15	15	18	18	21	21	24	⋆ 10.55416	9.54660	6.92967 : 4.85941
6	8	10	18	18	21	21	24	24	27	11.96330	9.54660	⋆ 5.01256 : 5.71360
6	9	10	19	20	23	24	25	28	29	10.68443	⋆ 8.01234	5.77137 : 6.12712
6	9	11	19	19	23	23	26	27	30	10.85878	⋆ 6.71874	5.55900 : 6.06592
7	10	12	22	22	26	26	29	30	33	⋆ 5.79225	⋆ 5.23929	⋆ 4.17393 : 6.92218
7	10	12	22	23	26	27	28	31	32	⋆ 4.90204	⋆ 4.00157	⋆ 3.47600 : 7.03555
8	11	14	25	26	29	30	34	34	38	⋆ 4.28932	⋆ 3.07764	⋆ 2.92078 : 8.01219
9	13	16	29	30	34	35	39	40	44	7.50491	5.40926	⋆ 2.63340 : 9.24342
10	14	17	32	33	37	38	42	43	47	⋆ 3.97613	3.44080	⋆ 2.62476 : 10.07952
11	16	19	35	37	40	42	47	50	52	6.39030	4.44659	⋆ 2.31299 : 11.22939
12	17	21	38	40	43	45	50	53	58	4.77293	⋆ 3.07764	⋆ 1.82447 : 12.14769
13	18	22	41	43	46	48	53	56	61	⋆ 3.97431	⋆ 2.33950	2.31651 : 13.00292
14	20	24	44	45	52	53	57	61	65	⋆ 3.96033	3.26241	3.17122 : 14.01237
14	20	24	45	45	51	51	57	60	66	⋆ 3.41130	3.08564	2.68886 : 13.94292
15	21	26	47	49	55	57	62	65	70	⋆ 3.13350	2.63203	2.05758 : 15.08443
15	21	26	48	50	54	56	61	65	70	⋆ 2.93182	2.84756	2.09273 : 15.11090
17	24	29	53	55	61	63	71	74	79	⋆ 2.71102	2.45221	1.92379 : 16.90841
19	27	33	60	62	69	71	79	83	91	3.31906	⋆ 2.19204	1.89656 : 19.05510
19	27	33	60	63	69	72	78	84	90	3.04095	2.46275	⋆ 1.43320 : 19.19285
20	28	34	63	65	72	74	82	86	94	3.20508	⋆ 2.00881	1.81379 : 19.96027
20	28	35	63	65	72	74	82	86	94	⋆ 2.22083	2.00881	1.81379 : 19.96027
22	31	38	69	72	78	81	90	96	102	⋆ 2.00322	⋆ 1.84247	1.75309 : 22.01933
22	32	39	70	73	80	83	92	96	105	4.19632	2.85190	⋆ 1.40590 : 22.31371
23	33	40	73	76	83	86	95	102	108	3.26905	2.33567	⋆ 1.33975 : 23.22603
27	38	46	85	88	98	101	110	117	126	2.83471	⋆ 1.72964	1.71518 : 26.99603
27	38	47	85	88	98	101	110	117	126	⋆ 1.95018	1.72964	1.71518 : 26.99603
30	42	52	94	98	107	111	124	130	140	⋆ 1.79386	1.76210	1.63211 : 30.03837

FIG. 5.16. *Résultat de l'énumération des masques* {a,b,c,g,h,i,j,k,l,m} *pour* $w_{\mathsf{a}} \in [1 \ldots 30]$, *une étoile* (⋆) *indique que le masque est optimal pour le critère correspondant.*

w_{a}	w_{b}	w_{c}	w_{g}	w_{h}	w_{i}	w_{j}	w_{k}	w_{l}	w_{m}	(2,1,0)	(2,1,1)	(2,2,1)	(4,2,1)	(4,3,1)	(4,3,2)	(5,4,1)	(5,4,3)	(7,4,1)	(7,4,3)	(7,6,3)
1	1	1	3	3	3	3	3	3	3									7	7	7
1	2	2	4	4	5	5	5	6	6									11	11	13
2	3	3	7	7	8	8	8	9	9									18	18	20
2	3	4	7	8	8	9	10	10	11									19	21	23
3	4	5	10	11	11	12	13	13	14									26	28	30
4	6	7	14	15	16	17	18	19	20									37	39	43
5	7	8	17	18	19	20	21	22	23									44	46	50
5	7	9	15	15	18	18	21	21	24	11	12	15						40	42	48
6	8	10	18	18	21	21	24	24	27	13	14	17						47	49	55
6	9	10	19	20	23	24	25	28	29	14	15							52	54	62
6	9	11	19	19	23	23	26	27	30	14	15	19						51	53	61
7	10	12	22	22	26	26	29	30	33	16	17	21						58	60	68
7	10	12	22	23	26	27	28	31	32	16	17	21						59	61	69
8	11	14	25	26	29	30	34	34	38	18	20	24						66	70	78
9	13	16	29	30	34	35	39	40	44	21	23	28						77	81	91
10	14	17	32	33	37	38	42	43	47	23	25	30						84	88	98
11	16	19	35	37	40	42	47	50	52	25	28	34	52					92	98	110
12	17	21	38	40	43	45	50	53	58	27	30	37	56					99	105	119
13	18	22	41	43	46	48	53	56	61	29	32	39	60					106	112	126
14	20	24	44	45	52	53	57	61	65	32	34	42						117	121	137
14	20	24	45	45	51	51	57	60	66	32	34	42	64					115	119	135
15	21	26	47	49	55	57	62	65	70	34	37	45						125	131	147
15	21	26	48	50	54	56	61	65	70	34	37	45	70		81			124	130	146
17	24	29	53	55	61	63	71	74	79	38	42	51	78					139	147	165
19	27	33	60	62	69	71	79	83	91	43	47	58	88					157	165	187
19	27	33	60	63	69	72	78	84	90	43	47	58	89		104			158	166	188
20	28	34	63	65	72	74	82	86	94	45	49	60	92					164	172	194
20	28	35	63	65	72	74	82	86	94	45	49	60	92		108			164	172	194
22	31	38	69	72	78	81	90	96	102	49	54	66	101		118		155	179	189	213
22	32	39	70	73	80	83	92	96	105	50	55	67	103	114	121	145		183	193	217
23	33	40	73	76	83	86	95	102	108	52	57	70	107		125		164	190	200	226
27	38	46	85	88	98	101	110	117	126	61	66	81	125		146			223	233	263
27	38	47	85	88	98	101	110	117	126	61	66	81	125		146			223	233	263
30	42	52	94	98	107	111	124	130	140	67	74	90	138		162		213	245	259	291

FIG. 5.17. *Liste des surcharges nécessaires pour les masques optimaux* {a,b,c,g,h,i,j,k,l,m} *(une case vide signifie que cette surcharge est inutile pour ce masque).*

Chapitre

6

Calcul de l'axe médian

6.1 Introduction

Dans ce chapitre nous présentons les notions de boules maximales et d'axe médian. Puis, nous présentons les trois méthodes existantes pour effectuer l'extraction de l'axe médian à partir de la carte de distance d'un objet discret : méthode des maxima locaux, méthode tenant compte des boules équivalentes, et utilisation des tables de correspondance. Nous présentons aussi les problèmes qui leurs sont liés, en particulier le problème ouvert qui consiste à calculer la table de correspondance.

Nous présentons ensuite notre méthode pour le calcul de la table de correspondance Lut. Nous montrons que ce problème est indissociable de la détermination du voisinage de test $\mathcal{M}_{Lut}$. Nous proposons un algorithme efficace permettant de déterminer le voisinage $\mathcal{M}_{Lut}$ pour toute fonction discrète, et de le valider, c'est-à-dire de vérifier que ce voisinage est nécessaire et suffisant pour que le calcul de l'axe médian soit correct.

6.2 Boules maximales et axe médian

On considère une image binaire discrète I contenant une forme $\mathcal{F}$ constituée par les points ayant une valeur non-nulle. Les points de $\mathcal{F}$ tels quels ne donnent aucune information globale sur la forme de $\mathcal{F}$.

On considère une famille $\mathcal{B}$ de boules définies par un rayon et un centre, telles que tout centre est inclus dans sa boule et que la boule de rayon 0 est égale à son centre. Un recouvrement de $\mathcal{F}$ par les boules de $\mathcal{B}$ est complètement défini par les positions des centres des boules et leurs rayons respectifs ; ces informations sont suffisantes pour reconstruire $\mathcal{F}$. Nous nous intéressons maintenant plus particulièrement au recouvrement par des boules maximales, définies comme suit :

Définition 17 (Boule maximale) *Une boule est* maximale *dans une forme si elle n'est incluse dans aucune autre boule incluse dans la forme.*

On note qu'une boule maximale peut être incluse dans l'union de plusieurs autres boules maximales ; en conséquence, un recouvrement par des boules maximales, qui est unique par construction, n'est que rarement un recouvrement minimal.

Définition 18 (Axe médian) *L'axe médian (medial axis, ou MA) d'une forme est l'ensemble des centres (et des rayons) des boules maximales incluses dans la forme.*

Historiquement, l'idée d'un axe de symétrie généralisé a été introduite par Blum en 1964, qui le baptisa plus tard axe médian [Blu67]. Par la suite Pfaltz et Rosenfeld [Pfa67] reprirent cette idée en l'adaptant à l'espace discret en introduisant la notion de boule maximale. L'axe médian constitue un codage réversible de la forme et une représentation globale ; il est centré dans la forme, et permet de faire de la description, de l'analyse, de la simplification ou de la compression de forme. L'axe médian est rarement fin et ses points sont souvent déconnectés, mais après un traitement approprié, il permet de très nombreuses opérations, comme par exemple le calcul du squelette [Thi94, San96], la reconstruction par surfaces implicites [Mar00], l'interpolation de coupes manquantes dans un volume numérique [Mon93, Muk00], ou encore l'amélioration de l'éclairage des volumes binaires pour leur visualisation [Muk99]. Il est enfin possible de calculer un recouvrement minimal d'une forme à partir de son axe médian (qui est recouvrement non-minimal), par exemple grâce à la méthode décrite par Borgefors et Nyström [Bor97], et ainsi disposer d'un codage efficace pour la compression ou le découpage d'objets en parties d'épaisseurs différentes.

L'aspect final de l'axe médian et son calcul dépendent de la géométrie de la famille de boules choisie (disques, sphères, polygones, etc). La figure 6.1 montre schématiquement le recouvrement d'une forme 2D par des disques, et l'axe médian obtenu (en pointillés).

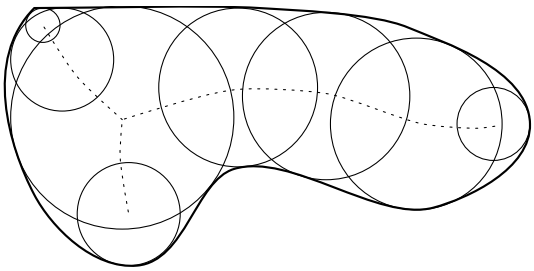


FIG. 6.1. *Axe médian et boules maximales.*

L'axe médian est particulièrement intéressant lorsque les boules utilisées sont des boules de distances, et plus particulièrement si la distance est une distance discrète. Nous présentons dans ce chapitre le calcul de l'axe médian à partir de la transformation de distance de la forme. Nous traitons dans un premier temps les distances de chanfrein, et dans un deuxième temps, le carré de la distance euclidienne.

6.3 Méthodes existantes pour le calcul de l'axe médian

Nous détaillons ici trois méthodes de calcul de l'axe médian à partir d'une carte de distance : la méthode des maxima locaux, sa variation qui tient compte des boules équivalentes et enfin celle utilisant des tables de correspondance.

6.3.1 Méthode des maxima locaux

Après la transformation de distance, chaque point p est étiqueté à sa distance $DT[p]$ au complémentaire ; $DT[p]$ est aussi le rayon de la plus grande boule dans la forme qui soit

centrée en p, qui est par (3.1) et la définition 6, la boule inverse $B_d^{-1}(p,DT[p])$.

Soit $(\overrightarrow{v},w)$ une pondération du masque de chanfrein $\mathcal{M}_C$. Le point $p+\overrightarrow{v}$ est plus profondément à l'intérieur de la forme que p si $DT[p+\overrightarrow{v}] > DT[p]$ (cf. figure 6.2). D'après la définition des distances de chanfrein, la plus grande valeur possible pour $DT[p+\overrightarrow{v}]$ est $DT[p]+w$, c'est-à-dire quand le point p a propagé l'information de distance locale au point $p+\overrightarrow{v}$ durant la transformation de distance. On en déduit que la boule centrée en $p+\overrightarrow{v}$ recouvre entièrement la boule centrée en p (cf. figure 6.2), et donc que $p \notin \mathrm{MA}$.

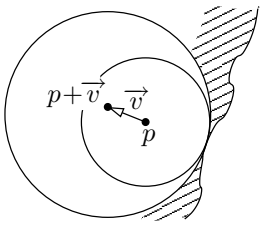


FIG. 6.2. *Maximum local.*

Au contraire, si p ne propage aucune pondération, alors p est appelé un *maximum local* et vérifie le critère

$$\text{LMC:} \qquad \forall(\overrightarrow{v},w) \in \mathcal{M}_C \ , \quad DT[p+\overrightarrow{v}] < DT[p]+w \ , \tag{6.1}$$

qu'on appelle Critère de Maximum Local (Local Maximum Criterion, ou LMC). L'ensemble des points détectés par le LMC, inclut par construction les points de MA. Rosenfeld et Pfaltz ont montré dans [Ros66] que pour les distances comme d_4 et d_8 en 2D, ou d_6 et d_{26} en 3D, ou plus généralement, toute distance de chanfrein où $w_{\mathsf{a}} = 1$, l'ensemble des points vérifiant le LMC est *exactement* l'axe médian.

Le critère de maximum local n'est malheureusement plus correct dès que $w_{\mathsf{a}} > 1$, car le LMC détecte les points de l'axe médian *plus* des points qui ne sont pas des centres de boules maximales. L'ensemble des points qui vérifient le LMC est certes toujours réversible, mais les points détectés par erreur sont en général nombreux, particulièrement près du bord de la forme, et ils rendent cet l'ensemble inutilisable pour la plupart des applications. Ces points supplémentaires sont détectés à cause de l'existence des *boules équivalentes.*

6.3.2 Boules équivalentes

Deux boules de rayon r et r' sont *équivalentes* si les ensembles de points $B_d(O,r)$ et $B_d(O,r')$ sont identiques (même si les valeurs des points sur la DT sont en général différentes). La *classe d'équivalence* d'une boule est l'intervalle de rayon pour lesquels les boules sont équivalentes. Le *conducteur* χ est le plus petit rayon pour lequel le cardinal de toutes les classes d'équivalence est 1, c'est-à-dire le rayon à partir duquel toutes les boules sont différentes. Le calcul du conducteur χ est lié au problème de Frobenius (voir [Syl84] et [Huj87]). Il est évident que si $w_{\mathsf{a}} = 1$ alors on a $\chi = 1$. Inversement, la classe d'équivalence de la boule inverse d'un pixel est $[1..w_{\mathsf{a}}]$, et donc $\chi \geqslant w_{\mathsf{a}}$ (des boules équivalentes apparaissent dès que $w_{\mathsf{a}} > 1$).

Le LMC teste dans (6.1) la différence entre des rayons de boules. Or, ce test est perturbé par l'apparition de classes d'équivalence, donc le LMC est incorrect si un des rayons au moins est inférieur ou égal à χ, ce qui arrive sur la DT dès que $w_{\mathsf{a}} > 1$.

Arcelli et Sanniti di Baja [Arc88] ont montré en 2D et pour des masques de taille 3, que le LMC est exact si on abaisse chaque valeur de DT au plus petit représentant de sa classe d'équivalence. Par exemple, pour $d_{3,4}$, il suffit de descendre les 3 à 1 et les 6 à 5. Leur méthode est malheureusement inappropriée pour des masques de taille supérieure à 3 en 2D ou 3D, à cause des interactions entre les cônes de la distance de chanfrein (cf. [Bor93] et [Thi94]). Nacken a montré dans [Nac94] comment calculer l'axe médian avec la distance 2D $d_{5,7,11}$ en ayant recours à la morphologie mathématique, mais son approche est plutôt complexe et ne semble pas facilement extensible à de plus grand masques ni à de plus grandes dimensions. Arcelli et Frucci utilisent dans [Arc92] un axe médian 2D calculé avec $d_{5,7,11}$, mais ils ne détaillent pas la méthode utilisée pour l'obtenir.

6.3.3 Tables de correspondances

La solution la plus générale et la plus efficace est la méthode des *tables de correspondance* (Look-Up Tables, ou LUT), qui mémorisent les corrections apportées au critère des centres maximaux.

Un point p dans une forme est un point de l'axe médian s'il n'existe aucun point q tel que la boule $B_d(q,DT[q])$ recouvre entièrement la boule $B_d(p,DT[p])$: on dit que la présence de q *interdit* à p de faire partie de l'axe médian. Supposons que (i) il est suffisant de chercher q dans un voisinage local de p et (ii) qu'on connaisse pour toute valeur de rayon $DT[p]$ la valeur minimale de rayon $DT[q]$, stockée dans une table Lut, qui interdise p dans la direction $\overrightarrow{v} = \overrightarrow{pq}$.

(i) Le voisinage de test autour de p est noté $\mathcal{M}_{Lut}$ et est 48-symétrique au même titre que nos masques de chanfrein. Le générateur de $\mathcal{M}_{Lut}$ est noté $\mathcal{M}^g_{Lut}$. Étant donné $\overrightarrow{v} \in \mathcal{M}_{Lut}$, on appelle $\overrightarrow{v}^g$ le vecteur de $\mathcal{M}^g_{Lut}$ correspondant à $\overrightarrow{v}$ par symétrie.

(ii) La valeur minimale pour p et $\overrightarrow{v}$ est stockée dans $Lut[\overrightarrow{v}][DT[p]]$. Du fait de la 48-symétrie, il est suffisant de ne stocker que les valeurs pour les vecteurs de $\mathcal{M}^g_C$, et donc la valeur minimale pour p et $\overrightarrow{v}$ est trouvée dans $Lut[\overrightarrow{v}^g][DT[p]]$.

Finalement, on a le critère suivant :

$$p \in \mathrm{MA} \iff \forall \overrightarrow{v} \in \mathcal{M}_{Lut}\,,\ DT[p+\overrightarrow{v}] < Lut[\overrightarrow{v}^g][DT[p]]\ . \tag{6.2}$$

L'application de ce critère se fait en un seul balayage de l'image, et avec au plus pour chaque point, un parcours de tout le masque $\mathcal{M}_{Lut}$. Soit L la longueur du côté de l'image, n sa dimension et m_{Lut} le nombre de directions formant le voisinage $\mathcal{M}_{Lut}$, la complexité en pire cas de l'algorithme est la même que celle de la transformation de distance : $O(m_{Lut}.L^n)$.

La première apparition des tables de correspondance est due à Borgefors et al. [Bor91b], pour d_E en 2D. La table de correspondance est calculée par une recherche exhaustive ; la complexité est énorme mais le calcul n'a besoin d'être fait qu'une fois. La table est donnée dans l'article jusqu'à un rayon de $\sqrt{80}$. Plus tard, Borgefors a donné la table de

correspondance pour la distance 2D $d_{5,7,11}$, dont les valeurs diffèrent du LMC pour des rayons inférieurs à $\chi = 60$; mais elle ne généralise sa méthode de calcul ni à d'autres distances ni à d'autres dimensions.

Dans [Thi94], Thiel a proposé un algorithme efficace pour calculer ces tables de correspondance pour toute distance de chanfrein 2D, sous l'hypothèse que $\mathcal{M}_{Lut} = \mathcal{M}_C$, mais il montre que pour de grands masques, des points supplémentaires sont souvent détectés (cf. figure 6.3), ce qui remet en cause la validité de toute sa méthode.

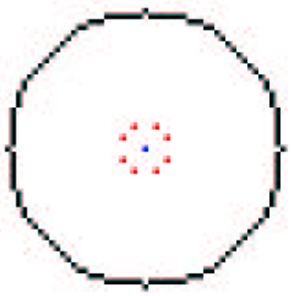


FIG. 6.3. *L'extraction de l'axe médian de la boule de rayon $r = 351$ obtenue en 2D avec les masques $\mathcal{M}_C^g = \mathcal{M}_{Lut}^g = \{(\mathsf{a},14),(\mathsf{b},20),(\mathsf{c},31),(\mathsf{d},44)\}$ conduit à conserver par erreur, en plus du centre (point bleu), huit points de trops (en rouge).*

Nous avons découvert que c'est l'hypothèse que $\mathcal{M}_{Lut} = \mathcal{M}_C$ qui est fausse. En fait, les deux masques sont généralement complètement différents. Nous proposons dans la suite, un algorithme correct et efficace qui calcule aussi bien le masque $\mathcal{M}_{Lut}$ que la table de correspondance Lut en 3D. Notre méthode est directement applicable en 2D simplement en éliminant toute référence à z (ou en 4D en ajoutant une quatrième coordonnée t). Ces résultats ont été publiés dans [Rem00a] et dans [Rem01].

6.4 Calcul de la Look-Up Table

Le calcul d'une valeur $Lut[\overrightarrow{v}][r]$ de la table de correspondance pour un rayon $r = DT[p]$ et une direction $\overrightarrow{v}$, consiste à trouver le plus petit rayon R de la boule $B_d^{-1}(p + \overrightarrow{v},R)$ qui recouvre entièrement la boule $B_d^{-1}(p,r)$ (cf. figure 6.2). On peut trouver R, comme illustré sur la figure 6.4, en faisant décroître R_+ tant que la boule $B_d^{-1}(q,R_+)$ recouvre $B_d^{-1}(p,r)$, où $q = p + \overrightarrow{v} = p - \overrightarrow{v}^g$ par symétrie. Malheureusement, pour chaque rayon R_+ il est nécessaire de faire une coûteuse transformation de distance inverse (*reverse distance transform*, ou RDT). Pour éviter cela, nous utilisons le lemme 1 du §3.1.5 qui lie les résultats de la DT et de la RDT, c'est-à-dire les boules et les boules inverses.

Nous allons montrer qu'il est suffisant de calculer la DT dans $\frac{1}{48}\mathbb{Z}^3$ une seule fois au début et ensuite de tester tous les recouvrements sur l'image résultante.

Pour ce qui est des distances de chanfrein, nous nous limitons aux distances associées à des normes (cf. §2.3.4). En conséquence, toutes les boules considérées auront les propriétés

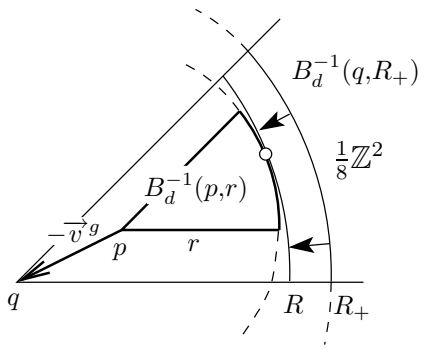


FIG. 6.4. *Test de recouvrement de deux boules dans* $\frac{1}{8}\mathbb{Z}^2$.

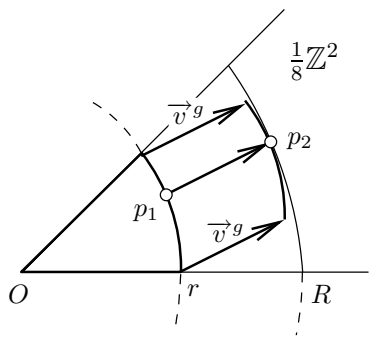


FIG. 6.5. *Test de recouvrement translaté dans* CT^g.

désirées : elles seront toutes

(i) convexes,

(ii) 48-symétriques, et

(iii) telles que si $r_1 < r_2$ alors $B_d(O,r_1) \subseteq B_d(O,r_2)$.

Nous pouvons donc limiter le test de recouvrement en limitant les deux boules à $\frac{1}{48}\mathbb{Z}^3$ (ou $\frac{1}{8}\mathbb{Z}^2$ en 2D), ce qui donne la figure 6.4. Dans la suite, nous notons CT^g l'image résultant d'une transformation de distance à l'origine par l'algorithme de la figure 5.1 tel qu'il est décrit au §5.4.2.

6.4.1 Calcul de la table Lut

Le recouvrement de la boule $B_d^{-1}(p,r)$ par $B_d^{-1}(q,R_+)$ peut être testé simplement en parcourant CT^g ; de plus, le plus petit rayon R peut être cherché pendant le balayage de CT^g. Nous proposons de translater les boules $B_d^{-1}(p,r)$ et $B_d^{-1}(q,R)$ à l'origine, comme indiqué sur la figure 6.5. Le point p_1 parcourt tous les points de $B_d^{-1}(O,r) \cap \frac{1}{48}\mathbb{Z}^3$, et donne par translation de vecteur $\overrightarrow{v}^g$ le point p_2. Les valeurs de $d(O,p_1)$ et $d(O,p_2)$ sont lues sur CT^g. On a

$$R = \max \left\{ d(O,p_2) \,:\, p_2 = p_1 + \overrightarrow{v}^g \,,\, p_1 \in B_d^{-1}(O,r) \cap \tfrac{1}{48}\mathbb{Z}^3 \right\} \text{ , et donc} \tag{6.3}$$

$$R = \max \left\{ d(O,p_1 + \overrightarrow{v}^g) \,:\, p_1 \in B_d^{-1}(O,r) \cap \tfrac{1}{48}\mathbb{Z}^3 \right\} . \tag{6.4}$$

Cette méthode peut être efficacement implémentée (cf. figure 6.6), car toutes les relations de recouvrement (r,R) dans une direction $\overrightarrow{v}^g$ peuvent être détectées durant le même balayage (lignes 2–7). Afin de rester dans les limites de l'image CT^g, le parcours en x est borné par $L - v_x^g - 1$ (où v_x^g est la composante sur x de $\overrightarrow{v}^g$). Pour chaque point p_1, on cherche le rayon r_1 correspondant, qui est $CT^g[p_1] + 1$ d'après le lemme 1. Puis on cherche le rayon r_2 de la boule passant par le point p_2. Sa valeur est $CT^g[p_2]+1 = CT^g[p_1+\overrightarrow{v}^g]+1$, d'après le lemme 1. Durant le balayage, on conserve dans $Lut[\overrightarrow{v}^g][r_1]$ la plus grande valeur trouvée pour r_2, qui à la fin du processus, est R d'après (6.4).

```
    Procedure CalculeLutCol ( CT^g, L, v^g, R_max, Lut[v^g] ) ;
 1  for r = 0 to R_max do Lut[v^g][r] = 0 ;                    // Initialise Lut[v^g] à 0.
 2  for x = 0 to L − v_x^g − 1 , for y = 0 to x , for z = 0 to y do
 3   {
 4     r_1 = CT^g[(x,y,z)] + 1 ;                    // Rayon de la boule où p_1 se trouve,
 5     r_2 = CT^g[(x,y,z) + v^g] + 1 ;                                  // idem pour p_2.
 6     if r_1 ≤ R_max and r_2 > Lut[v^g][r_1] then Lut[v^g][r_1] = r_2;
 7   }
 8  r_b = 0 ;
 9  for r_a = 0 to R_max do
10     if Lut[v^g][r_a] > r_b then r_b = Lut[v^g][r_a] else Lut[v^g][r_a] = r_b ;
```

FIG. 6.6. *Algorithme de calcul d'une colonne de Lut.* ***Entrées*** *: CT^g le cône, L la longueur du côté, $\overrightarrow{v}^g$ la direction de recherche, R_{max} la plus grande valeur de rayon devant être validée dans Lut.* ***Sortie*** *: la colonne $Lut[\overrightarrow{v}^g]$ est remplie avec les valeurs correctes.*

```
    Procedure CalculeEtVérifieLut(M_C^△,L,M_Lut^g,R_Connu,R_Cible,Lut);
 1  CalculeCT^g(L, M_C^△, CT^g) ;
 2  for each v^g in M_Lut^g do CalculeLutCol(CT^g, v^g, R_Cible, Lut ) ;
 3  for R = R_Connu + 1 to R_Cible do
 4   {
 5     for x = 0 to L − 1 , for y = 0 to x , for z = 0 to y do
 6        if CT^g[(x,y,z)] ≤ R
 7           then DT^g[(x,y,z)] = 1
 8           else  DT^g[(x,y,z)] = 0 ;                 // Copie de B_d(R) ∩ 1/48 Z^3 dans DT^g.
 9     CalculeDT^g(M_C^△, L, DT^g) ;
10     for x = 1 to L − 1 , for y = 0 to x , for z = 0 to y do
11        if DT^g[(x,y,z)] ≠ 0 and EstMA^g( (x,y,z), M_Lut^g, Lut, DT^g) then
12          {
13           M_Lut^g = M_Lut^g ∪ (x,y,z,CT^g[(x,y,z)]) ;                 // Extension de M_Lut^g
14           CalculeLutCol(CT^g, L, (x,y,z), R_Cible, Lut[(x,y,z)] ) ;          // et de Lut.
15           if EstMA^g ( (x,y,z), M_Lut^g, Lut, DT^g ) then error ;
16          }
17   }
```

FIG. 6.7. *Algorithme complet de calcul de Lut avec détermination de $\mathcal{M}^g_{Lut}$.* ***Entrées*** *: $\mathcal{M}^{\triangle}_C$ le générateur complété du masque de chanfrein, L la longueur du côté des images, $\mathcal{M}^g_{Lut}$ le générateur du voisinage de Lut, R_{Connu} le dernier rayon vérifié et validé, R_{Cible} le rayon maximum à vérifier.* ***Sorties*** *: Lut la table de correspondance (Look-Up Table), R_{Cible} le nouveau rayon vérifié. Au premier appel, $\mathcal{M}^g_{Lut}$ et R_{Connu} doivent être initialisés à $\emptyset$ et 0 respectivement. Après la terminaison de cet algorithme, R_{Connu} doit prendre la valeur R_{Cible} pour mémoriser que le nouveau $\mathcal{M}^g_{Lut}$ est valide jusqu'à R_{Cible}.*

À ce point, notre algorithme produit un ensemble de relations de recouvrement local, qui constituent un ordre partiel sur les recouvrements des boules. On peut observer dans

Lut, des cas où $r_a < r_b$ alors que $Lut[\overrightarrow{v}^g][r_a] > Lut[\overrightarrow{v}^g][r_b]$, ce qui signifie que la boule recouvrant $B_d^{-1}(O,r_a)$ est plus grande que la boule recouvrant $B_d^{-1}(O,r_b)$. Or, comme on l'a vu au §6.4, on ne considère que des distances, ou tout au moins des fonctions, telles que si $r_1 < r_2$ alors $B_d(O,r_1) \subseteq B_d(O,r_2)$. On corrige donc en conséquence la table en supposant que dans ce cas, $Lut[\overrightarrow{v}^g][r_b]$ est au moins égal à $Lut[\overrightarrow{v}^g][r_a]$; ce faisant, nous construisons un ordre total compatible (figure 6.6 lignes 8–10).

6.4.2 Calcul du voisinage de test $\mathcal{M}_{Lut}$

Nous nous concentrons maintenant sur le calcul de l'ensemble de pondérations $\mathcal{M}_{Lut}^g$, qui par symétrie donne $\mathcal{M}_{Lut}$. Ce calcul est décrit par l'algorithme de la figure 6.7.

Supposons qu'un voisinage $\mathcal{M}_{Lut}$ soit suffisant pour extraire correctement l'axe médian de n'importe quelle image DT dans laquelle les valeurs n'excèdent pas R_{Connu}. Cela signifie que $\mathcal{M}_{Lut}^g$ permet d'extraire de n'importe quelle boule $B_d(O,R)$ où $R \leqslant R_{Connu}$, un axe médian qui est, par la définition 18, le seul point O (le centre). Au début, $\mathcal{M}_{Lut}^g$ est vide et $R_{Connu} = 0$.

Afin d'augmenter R_{Connu} jusqu'à atteindre R_{Cible}, nous proposons de tester chaque boule $B_d(O,R)$ pour laquelle $R > R_{Connu}$, en extrayant à chaque fois sa transformation de distance et ensuite son axe médian, jusqu'à ce que R atteigne R_{Cible} ou qu'un point différent de O soit détecté dans l'axe médian de $B_d(O,R)$. Si R atteint R_{Cible}, alors nous savons que $\mathcal{M}_{Lut}^g$ permet l'extraction correcte de l'axe médian pour toute image DT contenant des valeurs inférieures ou égales à R_{Cible}, et donc que cette valeur R_{Cible} doit être mémorisée comme la nouvelle valeur de R_{Connu}.

Si au contraire, un point supplémentaire p est détecté dans l'axe médian durant le parcours, alors on sait que $\mathcal{M}_{Lut}^g$ ne suffit pas à extraire correctement l'axe médian, puisque par construction $B_d(O,R)$ recouvre $B_d^{-1}(p,DT^g[p])$. Dans ce cas, nous utilisons le lemme suivant :

Lemme 3 *Rajouter la pondération $(p,CT^g[p])$ dans $\mathcal{M}_{Lut}^g$ est nécessaire et suffisant pour enlever p de l'axe médian de la boule $B_d(O,R)$.*

Preuve : Le voisinage $\mathcal{M}_{Lut}^g$ est validé jusqu'à $R-1$, donc il permet de trouver toutes les boules directes, qui recouvrent la boule inverse $B_d^{-1}(p,DT^g[p])$, de rayon supérieur ou égal à $R-1$. Donc la seule boule qui n'est pas testée est la seule boule de rayon R : la boule $B_d(O,R)$ elle-même. La boule est dans la direction $\overrightarrow{pO}$ depuis p et doit être cherchée par $\mathcal{M}_{Lut}^g$ pour pouvoir enlever p de l'axe médian. Or puisque $\mathcal{M}_{Lut}$ est symétrique, $B_d(O,R)$ est détectée en rajoutant $\overrightarrow{Op}$ dans son générateur, c'est-à-dire en rajoutant la nouvelle pondération $(\overrightarrow{Op},CT^g[p])$ dans $\mathcal{M}_{Lut}^g$. □

Après avoir rajouté la nouvelle pondération, on calcule la nouvelle colonne correspondante qui se rajoute au tableau *Lut*. Ensuite, on s'assure que ce nouveau masque est désormais suffisant pour enlever p. Ceci est un test de consistance de l'algorithme de calcul de la colonne de *Lut* de la figure 6.6, car on sait que le nouveau masque est correct. Faillir à ce test ne peut qu'être une conséquence de l'impossibilité de construire une colonne de *Lut* qui soit assez restrictive pour enlever p. Dans ce cas, il ne peut exister de voisinage $\mathcal{M}_{Lut}$ et de table *Lut* pour cette distance.

Une fois p enlevé, on reprend le balayage pour le rayon R courant. D'autres points supplémentaires p peuvent être détectés séquentiellement, donnant à chaque fois une nouvelle pondération dans $\mathcal{M}^g_{Lut}$ et une nouvelle colonne dans Lut. Le calcul de $\mathcal{M}^g_{Lut}$ est terminé lorsque R atteint R_{Cible}.

Finalement, si la fonction de distance choisie satisfait bien (i-iii) du §6.4, et si un masque suffisant $\mathcal{M}_{Lut}$ existe pour tout rayon R_{Cible} donné, alors notre algorithme le calcule, ainsi que la table Lut correspondante. Sinon, cela signifie que toute la méthode des tables de correspondance est incorrecte pour cette distance, ce qui est aussi détecté par notre algorithme. Notons cependant que si une au moins des assertions (i-iii) n'est pas vérifiée, alors le résultat n'est plus prévisible.

L'algorithme complet, présenté à la figure 6.7, utilise deux autres algorithmes donnés aux figures 6.8 et 6.9. Ce sont des versions adaptées de la transformation de distance et de l'extraction de l'axe médian, dont le temps de calcul est 48^2 fois plus rapide que l'original : elles agissent sur $\frac{1}{48}\mathbb{Z}^3$ en utilisant les générateurs $\mathcal{M}^{\triangle}_C$ et $\mathcal{M}^g_{Lut}$, le tout en un seul balayage. Tout comme pour le calcul de CT^g, ces algorithmes ne sont valides que si les assertions (i-iii) du §6.4 sont vérifiées.

On remarque que le calcul de DT^g (figure 6.7, ligne 9) est obligatoire, car l'axe médian est extrait à partir de la carte de distance aux bords. Or un simple seuillage de l'image CT^g au rayon R donne seulement l'ensemble $B_d(O,R) \cap \frac{1}{48}\mathbb{Z}^3$, mais pas les valeurs correctes de DT^g (cf. figure 6.10, où les valeurs diffèrent de (a) à (b)).

6.5 Application

Nous donnons maintenant, deux applications de notre méthode. La première est la suite directe de ce qui a déjà été présenté puisqu'elle concerne les distances de chanfrein. La deuxième, en revanche, concerne l'application à des cartes de distances calculées avec le carré de la distance euclidienne, noté d^2_E, qui est une fonction discrète et dispose des propriétés nécessaires à la validité du calcul, bien que n'étant pas une distance (l'inégalité triangulaire n'est pas vérifiée).

6.5.1 Application aux distances de chanfrein

Nous donnons de la figure 6.11 à la figure 6.14 quelques tables de correspondance calculées avec notre algorithme en 2D et 3D. Un exemple 2D d'utilisation de ces tables, utilisant la formule (6.2) et la première ligne du tableau 6.11, est : un point de valeur 5 sur la DT n'est pas un point de l'axe médian, s'il a au moins un a-voisin supérieur ou égal à 6, ou un b-voisin supérieur ou égal à 8, ou un c-voisin supérieur ou égal à 12. Nous ne montrons dans les tables que les valeurs qui diffèrent du Critère des Centres Maximaux (c'est-à-dire $Lut[\overrightarrow{v}^g][r] \neq r + w$, cf. (6.1) et (6.2)), et donc qui représentent des irrégularités. Les tables sont aussi compressées en ne donnant que les rayons r qu'il est possible d'observer dans une carte de distance DT ; elles peuvent être détectées en un simple balayage sur CT^g.

Nous donnons à la figure 6.11, le résultat de notre algorithme pour la distance 2D très

```
    Function EstMA^g ( p, M^g_Lut, Lut, DT^g );
1   for each v^g in M^g_Lut do
2     if p − v^g ∈ (1/48)Z^3 then                    // Test seulement dans (1/48)Z^3.
3       if DT^g[p − v^g] ≥ Lut[v^g][DT^g[p]] then return false;
4   return true;
```

FIG. 6.8. *Extraction rapide des points de l'axe médian de B_d dans $\frac{1}{48}\mathbb{Z}^3$.* ***Entrées :*** *p le point à tester, $\mathcal{M}^g_{Lut}$ le générateur du voisinage de Lut, Lut la table de correspondance (Look-Up Table), DT^g la transformation de distance de la section de boule.* ***Sortie :*** *renvoie* ***true*** *si le point p est détecté comme un point de* MA *dans DT^g.*

```
    Procedure CalculeDT^g ( M^△_C, L, DT^g );
1   for z = L − 1 to 0 , for y = L − 1 to z , for x = L − 1 to y do
2     if DT^g[(x,y,z)] ≠ 0 then
3       {
4         min = +∞;
5         for each (v,w) in M^△_C do
6           {
7             (x′,y′,z′) = (x,y,z) + v;
8             if x′ < L and DT^g[(x′,y′,z′)] + w < min
9               then min = DT^g[(x′,y′,z′)] + w;
10          }
11        DT^g[(x,y,z)] = min;
12      }
```

FIG. 6.9. *Algorithme rapide de transformée de distance dans le cône.* ***Entrées :*** *L la largeur de l'image, $\mathcal{M}^{\triangle}_C$ le générateur complété du masque de d_C.* ***Sortie :*** *DT^g l'image de distance au bord, de taille L^3.*

		21	25		
	14	18	22	27	
7	11	16	21	26	
0	5	10	15	20	25

(a)

		7	5		
	14	11	7	5	
21	18	14	10	5	
28	25	20	15	10	5

(b)

FIG. 6.10. *Différence entre (a) CT^g et (b) DT^g calculées sur $B_{5,7,\mathsf{c}=11}(27) \cap \frac{1}{8}\mathbb{Z}^2$.*

employée $d_{5,7,\mathsf{c}=11}$. Les valeurs diffèrent de celles de Borgefors [Bor93] car nous calculons le plus petit rayon dans chaque classe de boules équivalentes, au lieu du plus grand. Puisque toutes les boules sont équivalentes entre ces deux rayons, les deux tables doivent être interprétées comme étant identiques.

La figure 6.12 montre la différence entre $\mathcal{M}^g_C$ et $\mathcal{M}^g_{Lut}$ qui cause le problème observé dans [Thi94, p. 81] quand on utilise la distance $d_{14,20,\mathsf{c}=31,\mathsf{d}=44}$ provenant de [Bor86]. On doit noter la présence dans $\mathcal{M}^g_{Lut}$ du point $2\mathsf{c}$, qui n'est pas un point visible, et donc n'aurait

$$\mathcal{M}_C^g = \mathcal{M}_{Lut}^g = \left\{ \begin{array}{l} \mathsf{a} = (1,0,5) \\ \mathsf{b} = (1,1,7) \\ \mathsf{c} = (2,1,11) \end{array} \right\}$$

R	a	b	c	R	a	b	c	R	a	b	c
5	6	8	12	21		27		38		44	
7	11	12	17	25	28	30	34	39		45	
10	12	15	19	27		33		40	44		
11		17		28		34		42		48	
14	17	19	23	29	33			46		52	
15	19			30	34			49		55	
16		22		31		37		53		59	
18	22	23	28	32		38		60		66	
20	23	26	30	35	39	41	45				

FIG. 6.11. *$\mathcal{M}_C^g$, $\mathcal{M}_{Lut}^g$ et Lut pour la distance 2D $d_{5,7,\mathsf{c}=11}$.*

$$\mathcal{M}_C^g = \left\{ \begin{array}{l} \mathsf{a} = (1,0,14) \\ \mathsf{b} = (1,1,20) \\ \mathsf{c} = (2,1,31) \\ \mathsf{d} = (3,1,44) \end{array} \right\}$$

$$\mathcal{M}_{Lut}^g = \left\{ \begin{array}{r} \mathsf{a} = (1,0,14) \\ \mathsf{b} = (1,1,20) \\ \mathsf{c} = (2,1,31) \\ \mathsf{d} = (3,1,44) \\ 2\mathsf{c} = (4,2,62) \\ \mathsf{i} = (5,2,75) \end{array} \right\}$$

FIG. 6.12. *$\mathcal{M}_C^g$ et $\mathcal{M}_{Lut}^g$ pour la distance 2D $d_{14,20,\mathsf{c}=31,\mathsf{d}=44}$.*

pas sa place dans $\mathcal{M}_C^g$, qui n'est constitué que de points visibles (cf. §3.1.6).

Les distances utilisées dans les figures 6.13 et 6.14 sont des normes optimales pour l'approximation de d_E qu'on a vues au §5.5. La figure 6.13 montre quelques exemples de tableaux *Lut* pour des distances 3D pour lesquelles $\mathcal{M}_C^g = \mathcal{M}_{Lut}^g$. On donne à la figure 6.14 un exemple complet, comprenant aussi bien le masque calculé $\mathcal{M}_{Lut}^g$ que le tableau *Lut* pour la distance $d_{11,16,19,\mathsf{j}=45}$. On note la différence entre $\mathcal{M}_C^g$ et $\mathcal{M}_{Lut}^g$, et comme à la figure 6.12, la présence de points non-visibles (3e).

Alors que le calcul de la table *Lut* tel qu'il est donné à la figure 6.6 est très rapide (moins d'une seconde sur un PC/Linux-2.2.16 PentiumIII 650MHz), le calcul donné à la figure 6.7, qui inclut le calcul et la vérification de $\mathcal{M}_{Lut}^g$, est lui beaucoup plus lourd, et son résultat doit donc impérativement être conservé pour tout usage ultérieur. Le calcul de $\mathcal{M}_{Lut}^g$ prend 41 secondes pour la distance 3D $d_{11,16,19,\mathsf{j}=45}$ pour $L = 100$ et pour $R_{Connu} = 0$ à $R_{Cible} = 1066$. Ce coût s'explique par le test systématique de 1066 boules $B_d(O,r)$, chacun d'entre eux nécessitant des calculs en $O(r^3)$ (calcul de CT^g et extraction de l'axe médian). Il est donc bien plus intéressant d'utiliser des distances de chanfrein avec un faible coefficient d'échelle puisque cela donne moins de boules à tester et donc un résultat beaucoup plus rapide. Dans cette optique, dans la plupart des cas, il est plus intéressant de privilégier l'ajout de nouvelles pondérations dans $\mathcal{M}_C^g$ pour pallier la perte de qualité dans l'approximation de d_E. Par exemple, l'extraction de $\mathcal{M}_{Lut}^g$ pour la distance $d_{7,10,12,\mathsf{d}=16,\mathsf{e}=17}$ pour $L = 100$ de $R_{Connu} = 0$ à $R_{Cible} = 685$ est plus rapide (29s), tout en conduisant à une meilleure approximation de d_E. Une autre façon de réduire le coût de calcul est de ne tester qu'une fois chaque classe de boules équivalentes, mais pour la plupart des distances de chanfrein, le gain n'est pas significatif, puisque les boules équivalentes n'apparaissent que pour de faibles rayons, là où la vérification est la plus rapide.

$d_{3,4,5}$

R	a	b	c
3	4	5	6

$d_{4,6,7,\mathsf{d}=9,\mathsf{e}=10}$

R	a	b	c	d	e
4	5	7	8	10	11
6	9	10	11	14	15
7	10				
8	11				
9		14	15		
12	15	17	18	20	21
16	19				

$d_{19,27,33}$

R	a	b	c
19	20	28	34
27	39	47	53
33	47	55	61
38	53	61	67
46	58	66	72
52	66	74	80
54	72	80	86
57	74	82	88
60	77	85	91
65	80	88	94
71	86	94	100
73	91	99	105

R	a	b	c
76	93	101	107
79	96	104	110
81	99	107	113
84	101	109	115
87	105	113	119
90	107	115	121
92	110	118	124
95	113	121	127
98	115	123	129
103	120	128	134
106	124	132	138
108	126	134	140

R	a	b	c
111	129	137	143
114	132	140	146
117	134	142	148
122	140	148	154
125	143	151	157
130	148	156	162
135	153	161	167
141	159	167	173
144	162	170	176
149	167	175	181
168	186	194	200

$d_{3,4,5,\mathsf{e}=7}$

R	a	b	c	e
3	4	5	6	8
4			8	
6	8			
7			11	

$d_{7,10,13,\mathsf{e}=18}$

R	a	b	c	e
7	8	11	14	19
10	15	18	19	26
13	18	21	24	29
14	19			
17	22	25	27	33
18			29	

R	a	b	c	e
20	26	29	32	37
23	29	32	34	40
24	29			
25			37	
26			37	
27	33			

R	a	b	c	e
28			40	
30	36	39	42	47
33			45	
34	40			
35			47	
36			47	

R	c
38	50
43	55
46	58
48	60
56	68

FIG. 6.13. *Exemples de Lut pour des distances 3D pour lesquelles* $\mathcal{M}_C^g = \mathcal{M}_{Lut}^g$.

Dans tous les cas étudiés, on constate expérimentalement que quelle que soit la norme de chanfrein utilisée pour le calcul de la carte de distance, le nombre de vecteurs formant le voisinage de test $\mathcal{M}_{Lut}$ est toujours borné, et donc que :

Conjecture 1 *Pour toute carte de distance calculée avec une norme de chanfrein* n_C*, l'extraction de l'axe médian peut toujours être effectuée par un test uniquement local dans le voisinage* $\mathcal{M}_{Lut}$ *calculé par notre algorithme.*

6.5.2 Application à la distance euclidienne

Notre méthode de calcul est suffisamment générale pour être appliquée en dehors du cadre des distances de chanfrein. Nous choisissons ici de présenter rapidement comment l'appliquer au cas d'une autre fonction discrète : le carré d_E^2 de la distance euclidienne. En effet, bien que cette fonction ne soit pas à proprement parler une distance (cf. contre-exemple de la figure 6.15), elle dispose néanmoins de toutes les propriétés nécessaires, vues au §6.4 (i-iii). Il est bien entendu possible d'utiliser notre algorithme sur d'autres fonctions discrètes du moment qu'elles vérifient toujours ces trois propriétés, comme c'est le cas pour l'arrondi de la distance euclidienne, ou sa troncature, mais cela n'est pas présenté dans ce

$$\mathcal{M}^g_C = \left\{ \begin{array}{c} \mathsf{a} = (1,0,0,11) \\ \mathsf{b} = (1,1,0,16) \\ \mathsf{c} = (1,1,1,19) \\ \mathsf{j} = (3,2,1,45) \end{array} \right\} \qquad \mathcal{M}^g_{Lut} = \left\{ \begin{array}{cc} \mathsf{a} = (1,0,0,11) & \mathsf{k} = (3,2,2,49) \\ \mathsf{b} = (1,1,0,16) & \mathsf{d} = (2,1,0,27) \\ \mathsf{c} = (1,1,1,19) & \mathsf{3e} = (6,3,3,90) \\ \mathsf{f} = (2,2,1,35) & \mathsf{j} = (3,2,1,45) \end{array} \right\}$$

R	a	b	c	f	k	d	3e	j
11	12	17	20	36	50	28	91	46
16	23	28	31	46	61	39	102	57
19	28	33	36	52	65	44	106	62
22	31	36	39	55	69	46	110	65
27	34	39	42	57	72	50	113	68
30	39	44	46	62	76	55	117	73
32	42	46	50	65	80	57	121	76
33					81		121	
35	45	50	53	68	83	61	124	79
38	46	52	55	71	84	62	125	81
41	50	55	58	74	88	66	129	84
43	53	57	61	76	91	68	132	87
44			62	78	91		132	
45				79				
48	57	62	65	81	95	73	136	91
49					97		136	
51	61	66	69	84	99	77	140	95
52	62				100	78	140	
54	64	68	72	87	102	79	143	98
55					103		144	
56				90				
57				91				
59	69	74	77	93	107	84	148	103
60			78	94	107		148	
61				95				
63	73	78	81	97	110	89	151	107
64		79		98		90		

R	a	b	c	f	k	d	3e	j
66			84	100	114		155	
67				101				
70	80	84	88	103	118	95	159	114
71					119		159	
72				106				
73			91	107	121		162	
74	84				122	100	163	
75		90		109		101		
76				110				
79				113				
80		95		114		106		
81					129		170	
82			100	116	129		170	
83				117				
85	95	100	103	119	133	111	174	129
86		101		120		112		
88				122				
89				123				
92			110	126	140		181	
93					141		182	
95				129				
96						122		
98				132				
99				133				
101				135				
102				136				
104			122	138	152		193	

R	k
105	139
108	142
111	145
114	148
117	151
118	152
120	154
121	155
124	158
127	161
130	164
133	167
137	171
140	174
143	177
146	180
149	183
156	190
159	193
162	196
165	199
175	209
178	212
194	228

FIG. 6.14. *$\mathcal{M}^g_{Lut}$ et Lut pour la distance 3D $d_{11,16,19,\mathrm{j}=45}$.*

document.

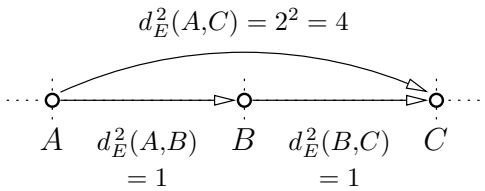


FIG. 6.15. *La fonction d_E^2 ne respecte pas l'inégalité triangulaire* (2.3) *: on n'a pas $d_E^2(A,C) \leqslant d_E^2(A,B) + d_E^2(B,C)$.*

Pour le calcul effectif de d_E^2, nous nous basons sur la transformation de distance proposée par Saito et Toriwaki [Sai94], car bien que leur algorithme ne soit pas le seul, il est simple à décrire de façon concise, et donc les modifications nécessaires pour notre objectif y sont aisément compréhensibles.

Afin d'adapter notre méthode, il est nécessaire de pouvoir calculer une carte de distance à l'origine dans $\frac{1}{48}\mathbb{Z}^3$; c'est ce que fait l'algorithme trivial de la figure 6.16.

	Procedure CalculeCT^g (L, CT^g) ;
1	**for** $x = 0$ **to** $L - 1$, **for** $y = 0$ **to** x , **for** $z = 0$ **to** y **do**
2	$CT^g[(x,y,z)] = x^2 + y^2 + z^2$;

FIG. 6.16. *Algorithme de transformation de distance à l'origine dans le cône pour d_E^2.* ***Entrée*** *: L la largeur de l'image,* ***Sortie*** *: CT^g l'image de distance a l'origine, de taille L^3.*

Il est par contre moins facile d'exhiber un algorithme pour le calcul de la distance au bord dans $\frac{1}{48}\mathbb{Z}^3$. Nous proposons à la figure 6.17, un algorithme très général (en 3D) qui fonctionne simplement, mais avec un coût de calcul assez important du fait qu'il fonctionne dans $\frac{1}{8}\mathbb{Z}^3$ au lieu de fonctionner seulement dans $\frac{1}{48}\mathbb{Z}^3$, et ne tire donc pas parti de toutes les symétries existantes. Cependant, on y reconnaît l'algorithme original de Saito et Toriwaki, à ceci près qu'on évite pour chaque balayage x, y et z, le balayage retour qui n'est pas nécessaire puisque la boule est symétrique.

Dans le cas 2D, nous proposons à la figure 6.18, un algorithme de remplacement qui fonctionne dans $\frac{1}{8}\mathbb{Z}^2$ mais dont le principe n'est pas facilement transposable en 3D.

En pratique, le calcul et la validation du masque $\mathcal{M}_{Lut}^g$ en utilisant cette méthode est possible, mais très lourd pour les raisons suivantes.

Conjecture 2 *On conjecture que le voisinage de test $\mathcal{M}_{Lut}$ est formé uniquement de points visibles, et que quand R grandit le nombre de vecteurs dans $\mathcal{M}_{Lut}$ augmente indéfiniment. On conjecture donc qu'il n'existe pas un masque $\mathcal{M}_{Lut}$ suffisant pour permettre l'extraction de l'axe médian de façon locale sur une image de distance quelconque calculée avec d_E^2 : l'extraction de l'axe médian est possible mais reste une opération globale sur l'image.*

Ces constatations ne sont pas surprenantes car la transformation de distance pour d_E^2 est une opération globale, et il n'y a donc pas de raisons pour que l'extraction d'axe médian

```
    Procedure CalculeDT^g ( L, DT^g ) ;
 1  for x = 0 to L − 1 , for y = 0 to x , for z = 0 to y do
 2     DT^g[(z,y,x)] = DT^g[(z,x,y)] = DT^g[(y,z,x)] =          // Calcul de B_{d_E^2} ∩ (1/8)Z^3
 3     DT^g[(y,x,z)] = DT^g[(x,z,y)] = DT^g[(x,y,z)] ;          // à partir de B_{d_E^2} ∩ (1/48)Z^3.
 4  df = 0 ;                                                    // Balayage en x.
 5  for z = L − 1 to 0 , for y = L − 1 to 0 , for x = L − 1 to 0 do
 6   {
 7     if DT^g[(x,y,z)] ≠ 0 then df = df + 1 ; else df = 0 ;
 8     DT^g[(x,y,z)] = df^2 ;
 9   }
10  for z = 0 to L − 1 , for x = 0 to L − 1 do                  // Balayage en y.
11   {
12     buff[L] = 0 ;
13     for y = 0 to L − 1 do buff[y] = CT^g[(x,y,z)] ;
14     for y = 0 to L do
15       {
16         d = buff[y] ;
17         if d ≠ 0 then
18            for n = 0 to min{√d, L − y} do
19               { w = buff[y + n] + n^2 ; if w < d then d = w ; }
20         if y ≠ L then DT^g[(x,y,z)] = d ;
21       }
22   }
23  for z = 0 to L − 1 , for x = 0 to L − 1 do                  // Balayage en z.
24     { ... }                  // idem lignes 11–22 en permutant les rôles de y et z.
```

FIG. 6.17. *Algorithme de transformée de distance pour d_E^2 dans le cas général (ici 3D).* ***Entrée*** *: L la largeur de l'image,* ***Sortie*** *: DT^g l'image de distance au bord, de taille L^3.*

soit une opération locale (avec un voisinage borné) plutôt qu'une opération globale (avec un voisinage qui est fonction de R).

Nous ne donnons pas ici d'exemple de masques $\mathcal{M}^g_{Lut}$ ni de tables de correspondance *Lut*, car leurs intérêts seraient très limités puisqu'ils ne seraient valables que jusqu'à un rayon donné.

6.6 Conclusion

Après avoir présenté un rappel des différentes méthodes d'extraction de l'axe médian d'une forme discrète à partir de sa carte de distance (maxima locaux, boules équivalentes et tables de correspondance), nous avons montré que la méthode plus générale des tables de correspondance (ou Look-Up Tables) repose sur un élément dont le calcul n'a jamais à notre connaissance été détaillé : celui de la table *Lut* elle-même.

```
    Procedure CalculeDT^g ( L, DT^g ) ;
 1  for x = 0 to L − 1 do                                   // Balayage en y.
 2      {
 3        k = 0 ;
 4        if DT^g[(x,x)] ≠ 0 then
 5           for i = x + 1 to L − 1 do
 6              if DT^g[(i,x)] ≠ 0 then k = k + 1 ; else break ;
 7        for y = x to 0 do
 8           if DT^g[(x,y)] ≠ 0 then {  k = k + 1; DT^g[(x,y)] = k^2 ;  }
 9      }
10  for y = 0 to L − 1 , for x = y to L − 1 do              // Balayage en x.
11     if DT^g[(x,y)] ≠ 0 then
12      {
13        value = DT^g[(x,y)] ;
14        for i = 0 to L − x − 1 do value = min{value,DT^g[(x + i,y)] + i^2} ;
15        DT^g[(x,y)] = value ;
16      }
```

FIG. 6.18. *Algorithme rapide de transformée de distance pour d_E^2 dans $\frac{1}{8}\mathbb{Z}^2$.* ***Entrée*** *: L la largeur de l'image,* ***Sortie*** *: DT^g l'image de distance au bord, de taille L^2.*

Nous apportons une solution à ce problème et nous montrons que le calcul des valeurs de la table est indissociable de celui du voisinage de test $\mathcal{M}_{Lut}$, et que celui-ci n'est pas, dans le cas général, égal au masque de chanfrein $\mathcal{M}_C$. Nous donnons un deuxième algorithme permettant la détermination et la validation de ce masque $\mathcal{M}_{Lut}$. Nos deux algorithmes tiennent compte des propriétés de symétrie des boules afin de réduire leurs coûts de calcul.

Nous donnons ensuite une série d'exemples de résultats de ces calculs. Nous donnons pour certaines distances de chanfrein optimales présentées au chapitre 5, le voisinage et la table de correspondance nécessaires et suffisants pour l'extraction de l'axe médian. Ces résultats peuvent être utilisés tels quels pour extraire des axes médians en 2D et 3D, et permettent ainsi d'éviter pour ces exemples, d'implémenter nos algorithmes.

Les algorithmes sont publiés dans [Rem01] et les sources en dimension 2, 3 et 4 sont disponibles sur `http://www.lim.univ-mrs.fr/~eremy/Recherche/PRL2001/index.html` avec d'autres exemples.

Afin de montrer la généralité et la robustesse de notre méthode, nous détaillons comment l'appliquer à une autre fonction discrète (qui n'est cependant pas une distance) : le carré de la distance euclidienne.

Finalement, on trouvera au chapitre 7 des exemples de visualisation tridimensionnelles d'objets 3D, de leur carte de distance et de leurs axes médians pour différentes distances de chanfrein, ainsi que pour le carré de la distance euclidienne.

Chapitre 7

Application

Dans ce chapitre nous décrivons les outils spécifiques que nous avons dû mettre au point pour mener à bien ce travail de de recherche. Nous donnons quelques exemples de transformations de distance (cf. §3.1.4) et d'extraction de l'axe médian (cf. §6) avec différentes normes de chanfrein ainsi qu'avec le carré de la distance euclidienne (cf. §6.5.2). Nous concluons en présentant l'application de nos travaux dans le cadre de l'action incitative «Cœur battant».

7.1 Développement logiciel

La manipulation et la visualisation des volumes numériques est un art peu commode pour qui n'est pas doté des outils *ad hoc*. Lorsqu'on travaille sur des images numériques 2D, il est toujours possible d'utiliser l'un des nombreux outils de visualisation déjà existants, mais en 3D les outils sont très rares et souvent inadaptés à notre problématique. En DEA, nous avons développé une solution simple de visualisation reposant sur le logiciel de lancer de rayon POV-Ray[1], mais cette solution bien que permettant de fournir des rendus de grande qualité ne nous permettait pas d'avoir l'interactivité souhaitée. Nous avons donc entrepris en début de thèse, l'écriture d'un logiciel de visualisation et de manipulation aussi bien des volumes numériques, que des distances de chanfrein.

Ce travail de développement représente environ 15 000 lignes de programme en langage C, et se répartit en trois sous-parties : une bibliothèque, un outil interactif de visualisation et de manipulation, et un interprète Scheme étendu.

La bibliothèque contient toutes les fonctions C qui traitent de la définition et de la création des volumes numériques ou des distances de chanfrein, de leur lecture et écriture sur disque, de leurs modifications, etc. Cette bibliothèque contient les algorithmes présentés dans ce document, ainsi que des outils classiques (fonctions de remplissage, transformations affines, seuillages, symétries, etc).

Nous présentons à la figure 7.1 une capture d'écran de notre outil interactif de visualisation (GtkDv). L'interface graphique est réalisée grâce à la bibliothèque Gtk+[2]. Le rendu 3D est effectué grâce au toolkit OpenGL[3] dans un widget dédié[4] compatible avec Gtk+.

Notre programme est actuellement prévu pour ne réaliser que des manipulations simples, c'est-à-dire ne faisant intervenir qu'un seul volume numérique en tant qu'argument. Pour effectuer une opération sur un volume numérique, il suffit d'utiliser un menu contextuel attaché au bouton droit sur la fenêtre contenant l'opérande choisi. Le résultat apparaît

1. Disponible librement sur `http://www.povray.org`.
2. Disponible librement sur `http://www.gtk.org` (licence GPL).
3. Voir `http://www.opengl.org`.
4. Disponible librement sur `http://www.student.oulu.fi/~jlof/gtkglarea` (licence GPL).

dans une nouvelle fenêtre. Ce menu contextuel est visible sur la figure 7.1 au-dessus de la fenêtre qui contient les lettres E et R sous forme de volume numérique.

Les trois fenêtres du bas représentent les trois étapes de calcul détaillées dans ce document. La première fenêtre (de gauche à droite) contient un objet binaire rappelant la forme d'un dé, défini géométriquement comme étant l'intersection entre une sphère et un cube. L'image est de côté 150 et représente donc environ 3,4 millions de voxels soit environ 6.8 Mo de mémoire (16bits/voxels). À partir de cet objet, on calcule sa carte de distance dont les lignes de niveaux internes ne sont pas visibles puisqu'elles sont cachées par la surface de l'objet. On a donc préféré montrer un écorché de cette carte de distance où ces lignes de niveaux sont visibles. À partir de la carte de distance, on calcule ensuite l'axe médian du dé (fenêtre de gauche). Ses points forment pour chaque «face» du dé une portion conique puis une portion pyramidale qui atteint le centre du dé.

Le rendu dans ces quatre fenêtres est interactif : il est possible de faire tourner l'objet, de zoomer ou encore de modifier les positions des deux lampes servant à l'éclairage de la scène.

Dans la fenêtre située en haut à gauche, on peut voir les différentes informations relatives à la fonction de distance choisie pour réaliser les calculs précédents : il s'agit de la norme de chanfrein $n_{23,33,40,\mathsf{g}=73,\mathsf{h}=76,\mathsf{j}=83,\mathsf{i}=86,\mathsf{k}=95,\mathsf{l}=102,\mathsf{m}=108}$. L'interface contient un calepin dont le premier onglet décrit le masque de chanfrein $\mathcal{M}_C$, tandis que le deuxième (invisible ici) décrit le masque $\mathcal{M}_{Lut}$ et le tableau Lut. On peut voir les 16 vecteurs définissant le générateur surchargé $\mathcal{M}_C^g$, les 217 vecteurs du demi-masque surchargé $\mathcal{M}_C^{1/2}$ obtenus par symétrie (en-dessous), ainsi que la liste des 9 triangles choisis pour former le générateur $\triangle$ de la triangulation. Pour chaque triangle, on peut lire la valeur des déterminants et du gradient discret qui leur sont associés. L'estimation d'erreur pour ce masque est disponible en bas à gauche : on peut lire les taux d'erreur suivant les critères MaxDiff, $\mathrm{MAE}_{\mathsf{a}}$ et $\mathrm{MAE}_{\mathrm{opt}}$ (ainsi que le s_{opt} associé). Toutes ces informations sont rassemblées dans le diagramme en projection (à gauche) où figurent les points visibles du masque (en rose), les triangles, les points rajoutés lors de la surcharge (en bleu), ainsi que la localisation du minimum (en vert) et du maximum (en rouge) d'erreur pour les critères MaxDiff (croix) et MAE (carré).

Nous avons développé parallèlement à notre logiciel graphique, un interprète permettant d'automatiser certaines tâches à l'aide de scripts. Après avoir exploré différentes voies, il nous est apparu comme la meilleure solution, d'utiliser Guile[5] qui est un interprète Scheme développé par le groupe GNU sous forme d'un bibliothèque, précisément dans le but de fournir un langage de script puissant. L'écriture de macros ou le prototypage de calculs impliquant les volumes numériques, les distances de chanfrein ou encore l'extraction d'axe médian s'en trouve ainsi très grandement facilité et à la portée de personnes qui ne sont pas familières avec l'implémentation des algorithmes.

5. Disponible librement sur `http://www.gnu.org/software/guile`.

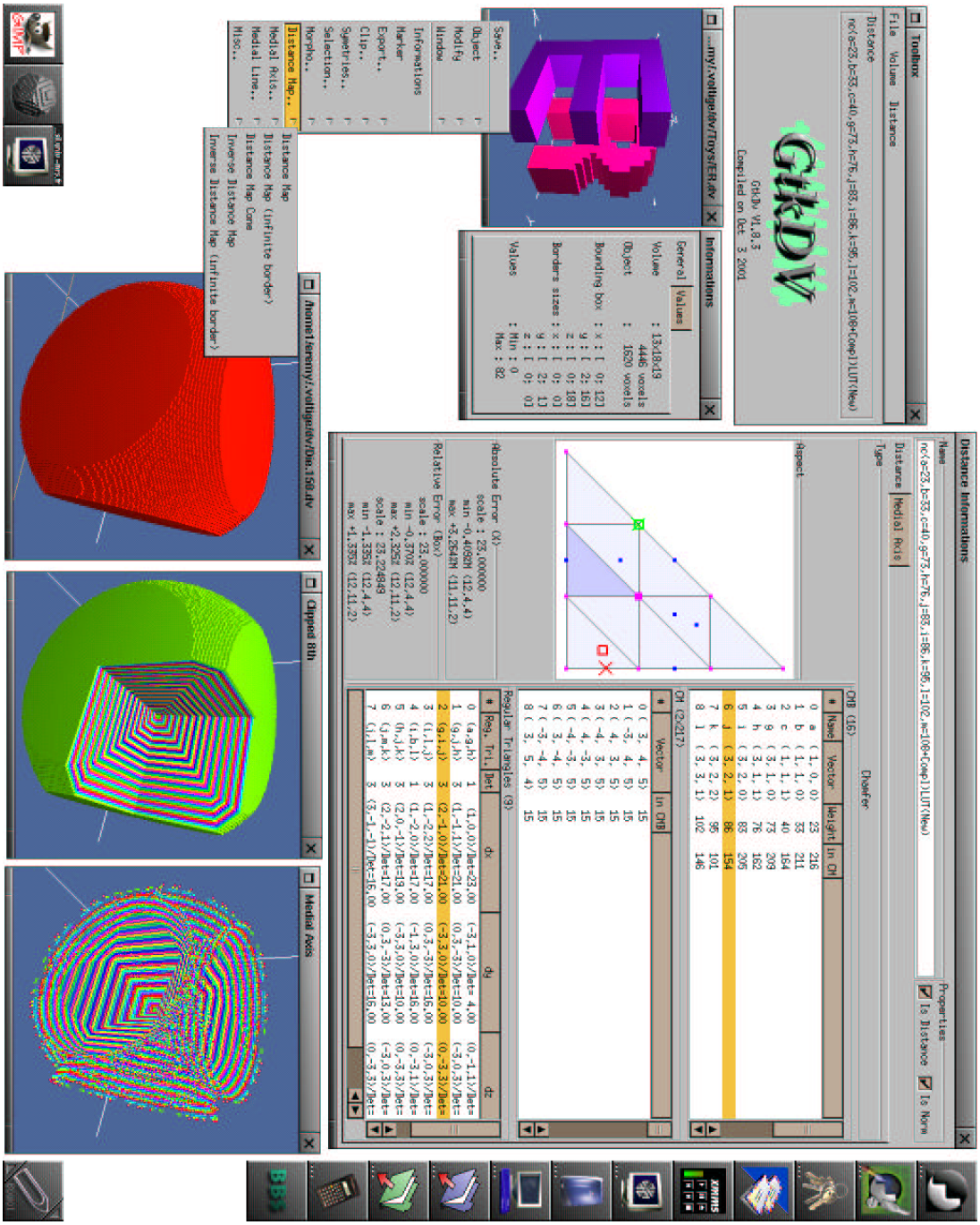


FIG. 7.1. *Capture d'écran de GtkDv.*

7.2 Images d'exemple

Une des principales qualité de notre système est la possibilité de travailler interactivement sur des images volumiques discrètes (rendu en OpenGL) et ensuite de pouvoir exporter les résultats obtenus vers le logiciel de lancé de rayon POV-Ray *à l'identique* (taille, orientation, éclairage, etc) afin d'obtenir des rendus de haute qualité.

Nous donnons à la figure 7.2 un premier exemple d'un tel rendu : l'objet (a) est formé par l'union de quatre sphères euclidiennes. L'image est un cube de côté 100 (soit 1 million de voxels), dont 222 693 voxels sont dans la forme. Grâce à la norme de chanfrein $n_{3,4,5}$, on calcule la carte de distance (b) et l'axe médian (c) de l'objet. On peut aussi calculer la carte de distance puis l'axe médian avec la norme de chanfrein $n_{29,41,51,\mathsf{d}=65,\mathsf{e}=71,\mathsf{f}=88}$ (d) ou avec le carré de la distance euclidienne d_E^2 (e).

La figure 7.3 représente un volume binaire contenant une vertèbre dont la numérisation a été effectuée au laboratoire TIMC-IMAG de Grenoble. Sur le même principe, on calcule avec la norme de chanfrein $n_{26,37,45,\mathsf{d}=59,\mathsf{e}=64}$, la carte de distance (b) et l'axe médian (c) à partir de l'image de côté 179 dans laquelle la vertèbre représente 217 668 voxels. Afin de permettre une meilleure «lecture» de cet axe médian, nous faisons figurer en (d) un axe médian simplifié qui est plus représentatif de la notion classique de squelette. Cette simplification est effectuée uniquement pour améliorer l'aspect visuel, en retirant tous les points de (c) qui ont moins de 5 points dans leur 26-voisinage.

Nous avons participé, avec d'autres membres de l'équipe LXAO, à l'action incitative inter GDR-PRC «Cœur battant»[6]. Ce projet qui implique des cardiologues, des radiologues, des informaticiens et des mécaniciens, vise à produire un modèle numérique réaliste du cœur humain. Ce modèle doit à terme permettre la simulation numérique de différents paramètres cardiaques (comme le flux sanguin) pour un cœur sain, mais aussi permettre de simuler efficacement les conséquences physiologiques de pathologies cardiaques (comme une ischémie). L'équipe LXAO participe à la modélisation géométrique du cœur à partir de données volumiques discrètes segmentées par d'autres membres de l'action incitative. Dans le cadre de ce projet, notre travail d'analyse morphologique dans le domaine discret s'inscrit en aval des travaux de Jean-Luc Mari qui permettent de synthétiser un modèle géométrique volumique continu par surface implicite [Mar01].

La figure 7.4 montre un volume binaire représentant un cœur de fœtus. Les données sont obtenues en photographiant des coupes après autopsie. Cette méthode d'acquisition des données donne une bonne résolution horizontale (256×256 dans le plan de coupe) et une piètre résolution verticale (60 coupes), mais a l'avantage de ne nécessiter aucune segmentation. Comme tous les calculs présentés dans ce document reposent sur l'utilisation de la maille cubique, il est nécessaire de rééchantillonner les données, ce qui a été fait ici en inférant les coupes manquantes pour obtenir le volume de taille $256 \times 256 \times 295$ qui est représenté en (a). Ces deux opérations (saisie et rééchantillonnage) ont été réalisées par Yves Usson et Annick Montanvert du laboratoire TIMC-IMAG de Grenoble et sont décrites dans [Mon93]. La figure (b) représente (comme pour la vertèbre) une partie de la carte de distance, tandis que les figures (c) et (d) représentent respectivement l'axe médian et l'axe

6. cf. `http://www-isis.enst.fr/Documents/PresentationGDR/Actions/Coeur.html`

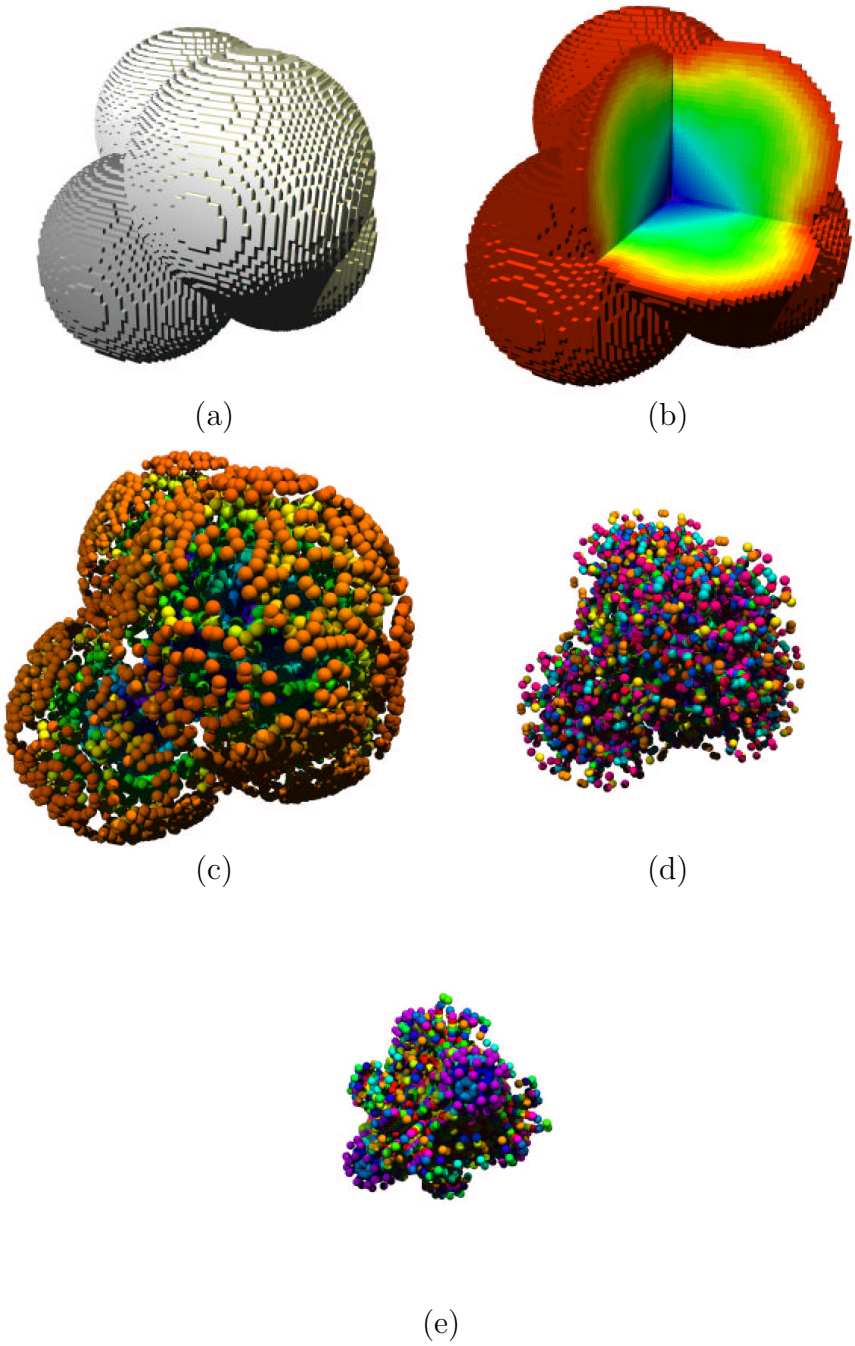


FIG. 7.2. *Exemple : objet «4 boules» (a), carte de distance avec* $n_{3,4,5}$ *(b), axe médian avec* $n_{3,4,5}$ *(c), axe médian avec* $n_{29,41,51,\mathsf{d}=65,\mathsf{e}=71,\mathsf{f}=88}$ *(c) et axe médian avec* d_E^2 *(d).*

médian simplifié.

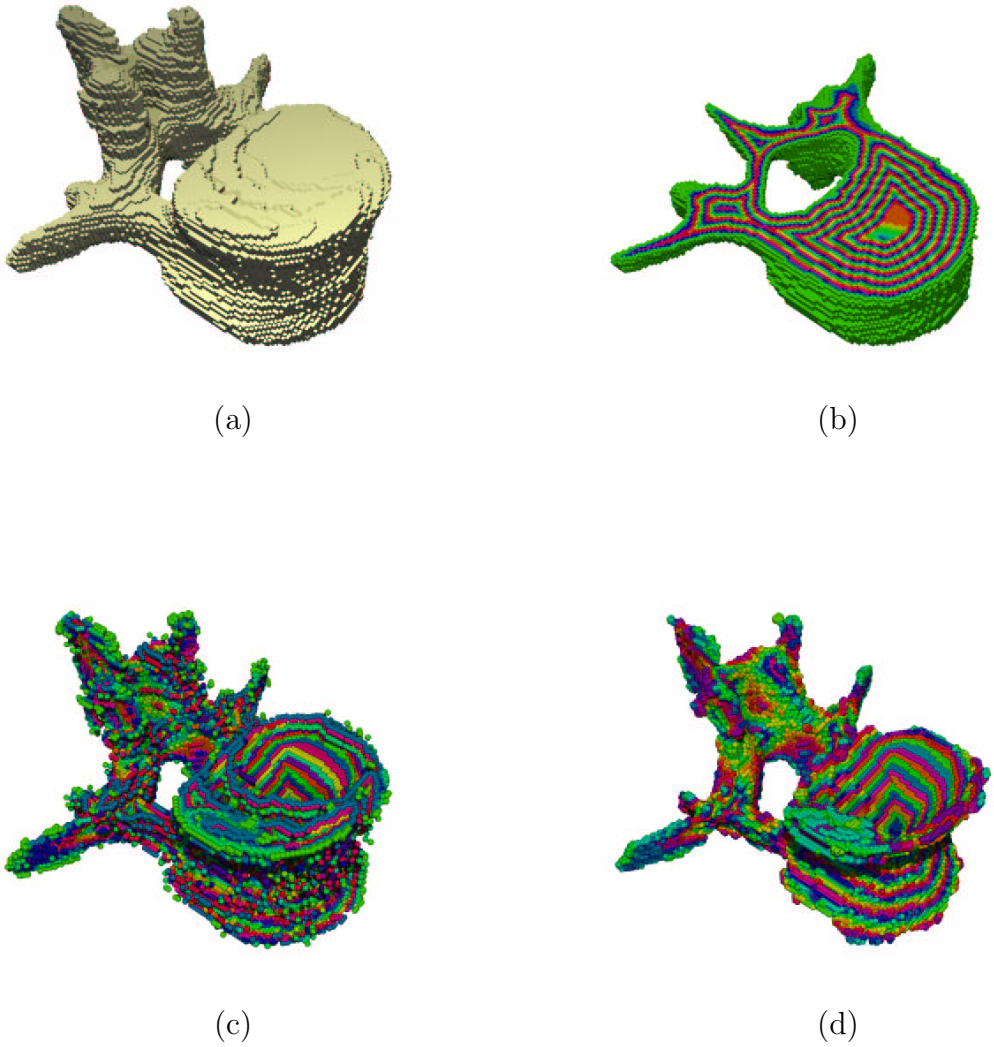


FIG. 7.3. *Exemple : objet vertèbre (a), carte de distance avec* $n_{26,37,45,\mathsf{d}=59,\mathsf{e}=64}$ *(b), axe médian (c) et axe médian simplifié (d).*

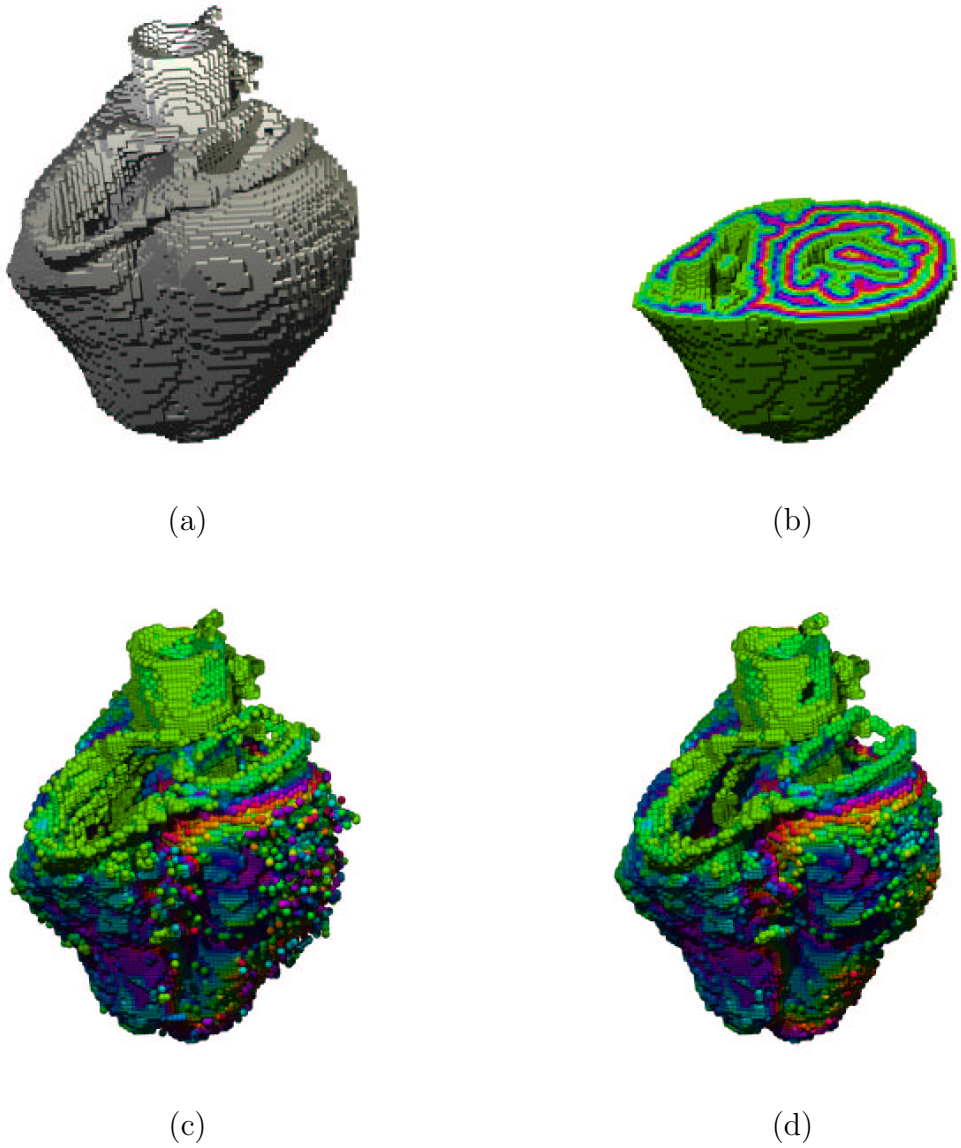


FIG. 7.4. *Exemple : cœur (a), coupe de la carte de distance (b), axe médian (c) et axe médian simplifié (d).*

Chapitre

8

Conclusion et perspectives

Nous avons tout d'abord défini au premier chapitre les distances et les normes discrètes, puis nous avons présenté, dans le deuxième chapitre, une définition des masques de chanfrein et des distances de chanfrein. Nous avons défini le générateur d'un masque, ainsi que le demi-masque, puis nous avons rappelé l'algorithme séquentiel permettant de calculer la carte de distance. Après avoir repris la démonstration de Verwer prouvant que tout masque de chanfrein induit bien une distance discrète, et après avoir présenté la notion de régularité au sens de Borgefors, nous avons montré, au §3.2.4, que la propriété de régularité n'est pas suffisante, à elle seule, pour assurer qu'une distance de chanfrein ait la propriété d'homogénéité indispensable à toute norme discrète.

Au chapitre trois, nous avons détaillé une démarche différente de celle de Borgefors et Kiselman ayant cependant le même but : produire des normes de chanfrein. Nous avons montré que tout masque de chanfrein induit bien une norme dans l'espace continu, et que son expression est celle d'une fonction linéaire par morceaux. Nous avons montré que chacun des morceaux correspond à une facette de l'enveloppe convexe du masque rationnel équivalent. Nous avons présenté une méthode qui, après avoir choisi un masque de chanfrein et une triangulation de ses points, permet de s'assurer que la triangulation est convexe, et ainsi par conséquent de construire une norme de chanfrein. Nous avons insisté sur le rôle de la triangulation, et montré qu'une opération de surcharge peut être utilisée pour rendre valides des masques non-réguliers.

Au chapitre quatre, nous avons montré comment obtenir des normes de chanfrein optimales en fonction des critères d'erreur MaxDiff, $\mathrm{MAE}_{\mathsf{a}}$ et $\mathrm{MAE}_{\mathrm{opt}}$. Nous avons explicité le rôle du facteur d'échelle et son calcul. Nous avons donné, pour une série de masques réguliers et non-réguliers, les résultats de l'optimisation ; les masques optimaux 3D ainsi proposés sont nouveaux et bien plus performants que le classique $d_{3,4,5}$ (jusqu'à 2% d'erreur au lieu de 8%).

Au chapitre cinq, nous avons détaillé la méthode de calcul de l'axe médian par les tables de correspondance développée par Borgefors, et nous avons souligné l'importance de savoir comment calculer ces tables pour toute distance. Nous avons repris et généralisé l'algorithme de Thiel effectuant ce calcul. Nous avons introduit la notion de voisinage de test, puis nous avons montré qu'il est, dans le cas général, différent du masque de chanfrein. Nous avons donné un algorithme performant de calcul du voisinage de test qui procède à la validation de toute la méthode pour n'importe quelle fonction discrète du moment que ses boules sont (i) convexes, (ii) 48-symétriques et (iii) incluses les unes dans les autres à rayon croissant. Nous avons donné des exemples de tables de correspondance et de voisinage de test pour différentes normes de chanfrein optimales présentées dans ce document. Enfin, pour illustrer la robustesse et la généralité de notre méthode, nous présentons son adaptation à une fonction discrète qui n'est ni une norme, ni une distance : le carré de la distance euclidienne.

Le sixième et dernier chapitre décrit le développement logiciel effectué, et termine cet exposé par quelques exemples pratiques de calcul de cartes de distance et d'axes médians en 3D.

Toutes les méthodes exposées dans ce document sont *a priori* généralisables à des dimensions supérieures à trois, cependant des notions comme la triangulation du masque de chanfrein deviennent alors inévitablement difficiles à maîtriser, et il serait sans doute intéressant d'approfondir la question, car les applications des cartes de distance 4D pour la planification des mouvements en robotique sont nombreuses.

Enfin, un travail important reste à produire pour amener le calcul à partir de l'axe médian du squelette discret 3D, au niveau de maîtrise déjà obtenu en 2D. En effet, la tâche s'annonce ardue car il sera alors nécessaire de savoir reconnaître les plans et les surfaces discrètes le formant, problème actuellement encore ouvert.

Bibliographie

[Arc88] C. Arcelli and G. Sanniti di Baja. Finding local maxima in a pseudo-euclidean distance transform. *Computer Vision, Graphics and Image Processing*, 43:361–367, 1988.

[Arc92] C. Arcelli and M. Frucci. Reversible skeletonization by (5-7-11)-erosion. In C. Arcelli et al., editors, *Visual Form Analysis and Recognition*, pages 21–28. Plenum, New York, 1992.

[Bar77] H.G. Barrow, J.M. Tenenbaum, R.C. Bolles, and H.C. Wolf. Parametric correspondence and chamfer matching: two new techniques for image matching. In *proc. 5th Int. Joint Conf. on Artif. Intell.*, pages 659–663, 1977.

[Bec90] A.L.D. Beckers and A.W.M. Smeulders. The probability of a random straight line in two and three dimensions. *Pattern Recognition Letters*, 11:233–240, 1990.

[Blu67] H. Blum. A transformation for extracting new descriptors of shape. In W. Wathendunn, editor, *Models for the Perception of Speech and Visual Form*, pages 362–380, Cambridge, 1967. MIT Press.

[Bor84] G. Borgefors. Distance transformations in arbitrary dimensions. *Computer Vision, Graphics and Image Processing*, 27:321–345, 1984.

[Bor86] G. Borgefors. Distance transformations in digital images. *Computer Vision, Graphics and Image Processing*, 34:344–371, 1986.

[Bor91a] G. Borgefors. Another comment on ‘a note on ‘distance transformation in digital images”. *Computer Vision, Graphics and Image Processing*, 54:301–306, 1991.

[Bor91b] G. Borgefors, I. Ragnemalm, and G. Sanniti di Baja. The Euclidean Distance Transform : finding the local maxima and reconstructing the shape. In *7th Scandinavian Conf. on Image Analysis*, volume 2, pages 974–981, Aalborg, Denmark, 1991.

[Bor93] G. Borgefors. Centres of maximal disks in the 5-7-11 distance transform. In *8th Scandinavian Conf. on Image Analysis*, pages 105–111, Tromsø, Norway, 1993.

[Bor96] G. Borgefors. On digital distance transforms in three dimensions. *Computer Vision and Image Understanding*, 64(3):368–376, 1996.

[Bor97] G. Borgefors and I. Nyström. Efficient shape representation by minimizing the set of centres of maximal discs/spheres. *Pattern Recognition Letters*, 18:465–472, 1997.

[Bor00] G. Borgefors. Some weighted distance transforms in four dimensions. In *9th Discrete Geometry for Computer Image*, volume 1953 of *Lectures Notes in Computer Science*, pages 325–336, Uppsala, Sweden, Dec 2000.

[Bor01] G. Borgefors and S. Svensson. Optimal local distances for distance transforms in 3D using an extended neighbourhood. In *9th Discrete Geometry for Computer Image*, volume 2059 of *Lectures Notes in Computer Science*, pages 113–122, May 2001.

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

[Har78] G.H. Hardy and E.M. Wright. *An introduction to the theory of numbers.* Oxford University Press, fifth edition, October 1978. §3.1.

[Hil69] J. Hilditch and D. Rutovitz. Chromosome recognition. *Ann. New York Acad. Sci.*, 157:339–364, 1969.

[Huj87] M. Hujter and B. Vizvari. The exact solutions to the Frobenius problem with three variables. *Ramanujan Math. Soc.*, 2(2):117–143, 1987.

[Kis96] C.O. Kiselman. Regularity properties of distance transformations in image analysis. *Computer Vision and Image Understanding*, 64(3):390–398, Nov 1996.

[Mar00] J.L. Mari and J. Sequeira. Using implicit surfaces to characterize shapes within digital volumes. In *RECPAD'00*, pages 285–289, Porto, Portugal, May 2000.

[Mar01] J.L. Mari, L. Astart, and J. Sequeira. Geometrical modeling of the heart and its main vessels. In *First International Workshop on Funcional Imaging and Modeling of the Heart, FIMH'2001*, Helsinki, Finland, Novembre 2001. Accepted.

[Mon68] U. Montanari. A method for obtaining skeletons using a quasi-euclidean distance. *Journal of ACM*, 15:600–624, 1968.

[Mon93] A. Montanvert and Y. Usson. Discrete distances applied to 2D granulometry and to 3D reconstruction. In *Proc. of the 8th Scandinavian Conf. on Image Analysis, Tromsø, Norway*, pages 1153–1160, May 1993.

[Muk99] J. Mukherjee, M. Aswatha Kumar, B.N. Chatterji, and P.P. Das. Discrete shading of three-dimensional objects from medial axis transform. *Pattern Recognition Letters*, 20:1533–1544, 1999.

[Muk00] J. Mukherjee, M. Aswatha Kumar, P.P. Das, and B.N. Chatterji. Fast computation of cross-sections of 3d objects from their medial axis transforms. *Pattern Recognition Letters*, 21:605–613, 2000.

[Nac94] P.F.M. Nacken. *Image analysis methods based on hierarchies of graphs and multiscale mathematical morphology.* PhD thesis, (Unknown school), Amsterdam, 1994.

[Pfa67] J.L. Pfaltz and A. Rosenfeld. Computer representation of planar regions by their skeletons. *Comm. of ACM*, 10:119–125, feb 1967.

[Rem99] E. Remy and E. Thiel. Triangulations des boules de chanfrein. In 12èmes *journées de l'AFIG*, pages 126–135, Reims, Nov 1999.

[Rem00a] E. Remy and E. Thiel. Computing 3D medial axis for chamfer distances. In 9th *DGCI, Discrete Geometry for Computer Image*, volume 1953 of *Lectures Notes in Computer Science*, pages 418–430, Uppsala, Sweden, Dec 2000.

[Rem00b] E. Remy and E. Thiel. Optimizing 3D chamfer masks with norm constraints. In 7th *IWCIA, Int. Workshop on Combinatorial Image Analysis*, pages 39–56, Caen, July 2000.

[Rem00c] E. Remy and E. Thiel. Structures dans les sphères de chanfrein. In 12ème *RFIA, Rec. des Formes et I.A*, volume 1, pages 483–492, Paris, Fev 2000.

[Rem01] E. Remy and E. Thiel. Medial axis for chamfer distances: computing look-up tables and neighbourhoods in 2D or 3D. *Pattern Recognition Letters*, 23:649–661, 2001.

[Ros66] A. Rosenfeld and J.L. Pfaltz. Sequential operations in digital picture processing. *Journal of ACM*, 13(4):471–494, 1966.

[Ros68] A. Rosenfeld and J.L. Pfaltz. Distance functions on digital pictures. *Pattern Recognition*, 1:33–61, 1968.

[Sai94] T. Saito and J.I. Toriwaki. New algorithms for euclidean distance transformation of an n-dimensional digitized picture with applications. *Pattern Recognition*, 27(11):1551–1565, 1994.

[San96] G. Sanniti di Baja and E. Thiel. A skeletonization algorithm running on path-based distance maps. *Image and Vision Computing*, 14(1):47–57, Feb 1996.

[Syl84] J. Sylvester. Mathematicals questions with their solutions. *Educational Times*, 41:21, 1884.

[Thi94] E. Thiel. *Les distances de chanfrein en analyse d'images : fondements et applications*. PhD thesis, Université Joseph Fourier, Grenoble 1, Sept 1994. `http://www.lim.univ-mrs.fr/~thiel/these`.

[Ver91a] J.H. Verwer. *Distance transforms: metrics, algorithms and applications*. PhD thesis, Technische Universiteit, Delft, 1991.

[Ver91b] J.H. Verwer. Local distance for distance transformations in two and three dimensions. *Pattern Recognition Letters*, 12:671–682, 1991.

[Vos88] A.M. Vossepoel. A note on "Distance transformations in digital images". *Computer Vision, Graphics and Image Processing*, 43:88–97, 1988.

Résumé

Dans un premier temps, nous présentons une classe de distances discrètes : *les distances de chanfrein*. Nous montrons que les propriétés de ces fonctions dépendent de la géométrie de l'enveloppe convexe des points du *masque de chanfrein* servant à les définir. Nous présentons une méthode permettant de construire des masques de chanfrein *réguliers* qui définissent des *normes discrètes* (vérifiant la propriété d'homogénéité). Nous effectuons, sur la base des contraintes ainsi déterminées, une optimisation afin de trouver des exemples pratiques dans les cas 3D de normes de chanfrein *optimales* (minimisant l'erreur par rapport à la distance euclidienne).

Dans un deuxième temps, nous définissons l'*axe médian* d'une forme, qui est l'ensemble des centres des boules maximales inscrites dans cette forme. Nous détaillons son calcul à partir de la carte de distance de la forme, par la méthode des tables de correspondance. Nous présentons une méthode de détermination des valeurs de cette table pour toute distance discrète en 2D ou 3D, puis nous présentons une méthode permettant de calculer, ainsi que de valider le voisinage de test, dont dépend le calcul local de l'axe médian. Nous donnons enfin plusieurs exemples de tables et de voisinages obtenus dans le cas des normes de chanfrein 3D, ainsi que dans celui du carré de la distance euclidienne.

Mots clefs : Distance de Chanfrein, Normes de Chanfrein, Axe Médian, Centres des Boules Maximales, Tables de Correspondances, Representation de Formes.
Discipline : Informatique.
Laboratoire : Laboratoire d'Informatique de Marseille.

Summary

Chamfer Norms and Medial Axis in the Discrete Volume

First, we present a class of discrete distances: the *chamfer distances*. We show that the properties of these functions depend on the geometry of the convex hull of the *chamfer mask points* used to define them. We present a method to construct *regular* chamfer masks which define *discrete norms* (verifying the homogeneity property). We obtain constraints and then optimize them, in order to find practical 3D examples of *optimal* chamfer norms (minimizing the error between the chamfer distance and the euclidean distance).

Then, we define the *medial axis* of a shape, which is the set of centres of maximal balls inside this shape. We detail its computation from the distance map of the shape, by the method of the look-up tables. We present a method to determine the values of these look-up tables for any discrete distance in both 2D and 3D. Then, we present a method to compute and validate the test neighbourhood on which depends the local computation of the medial axis. Finally, we give several examples of look-up tables and neighbourhood obtained in the case of 3D chamfer norms, and in the case of the square of the euclidean distance.

Key words: Chamfer Distances, Chamfer Norms, Medial Axis, Centres of Maximal Balls, Look-Up Tables, Shape Representation.