

THÈSE

pour obtenir le grade de

DOCTEUR

en

INFORMATIQUE

présentée et soutenue publiquement par

David COEURJOLLY

le 18 DÉCEMBRE 2002

**Algorithmique et géométrie
discrète pour la caractérisation des
courbes et des surfaces**

préparée au sein du laboratoire ERIC

sous la direction de

Serge Miguet et Laure Tougne

COMPOSITION DU JURY

M.	Jean-Pierre Braquelaire	Rapporteur	(Professeur)
M.	Jean-Marc Chassery	Rapporteur	(Directeur de Recherche CNRS)
M.	Jean-Michel Jolion	Examineur	(Professeur)
M.	Jean-Pierre Reveillès	Examineur	(Professeur)
M.	Serge Miguet	Directeur de thèse	(Professeur)
Mme.	Laure Tougne	Directeur de thèse	(Maître de conférences)

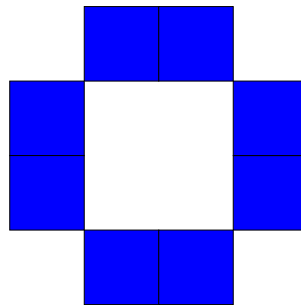
Table des matières

Introduction	1
I Géométrie discrète et objets discrets	5
1 Notions de base	7
1.1 Introduction	8
1.2 Espace discret et connexité	8
1.3 Objets, courbes, surfaces et hyper-surfaces	10
1.4 Codage et propriétés du codage des courbes discrètes	13
1.5 Processus de discrétisation	15
1.6 Conclusion	17
2 Droites et Plans discrets	19
2.1 Introduction	20
2.2 Droites discrètes 2D	20
2.3 Droites discrètes 3D	44
2.4 Plans Discrets	54
2.5 Statistique pour la reconnaissance de droites et plans discrets	73
2.6 Conclusion	77
3 Le cercle discret	79
3.1 Introduction	80
3.2 Synthèse et analyse du cercle discret	80
3.3 Reconnaissance et segmentation	83
3.4 Conclusion	101
II Analyse d'objets et mesures	103
4 Métriques discrètes	105
4.1 Introduction	107
4.2 Transformée en distance	107
4.3 Squelette de forme et axe médian	133
4.4 Visibilité et géodésiques discrètes	142

4.5	Conclusion sur ce chapitre	157
5	Mesures sur des objets discrets	159
5.1	Introduction	161
5.2	Contexte théorique : la convergence asymptotique	161
5.3	Tangentes et normales	163
5.4	Longueur et aire	176
5.5	Courbures d'objets discrets	197
5.6	Conclusion	218
	Conclusions et perspectives	221
III	Annexes	225
A	Applications	229
A.1	Introduction	230
A.2	Extraction de primitives pour la classification de profils de stèles funéraires	230
A.3	Analyse microscopique d'échantillon de neige	234
A.4	Modèle déformable discret	237
A.5	Conclusion	238
B	Preuves de convergence asymptotique	239
B.1	Introduction	240
B.2	Courbes en dimension d : approche par <i>tube</i>	240
B.3	Surfaces : aire et normales	245
C	Programmation linéaire : algorithme de PREPARATA et SHAMOS	251
D	Index des auteurs cités	255

Deuxième partie

Analyse d'objets et mesures



longueur : 9.658543
courbure moyenne : 0.02324
variance de la courbure : 0.0003422
mmhhh... ça doit être un cercle...

à moins que ce ne soit un carré ?

Chapitre 4

Métriques discrètes

Sommaire

4.1	Introduction	107
4.2	Transformée en distance	107
4.2.1	Métriques de chanfrein	107
4.2.2	Transformée en distance euclidienne	108
4.2.2.1	Approche par déplacements signés	108
4.2.2.2	Approche multi-dimensionnelle de SAITO et TORIWAKI	109
4.2.3	Diagramme de Voronoi et transformée en distance euclidienne	110
4.2.3.1	Diagramme de Voronoi et triangulation de Delaunay en géométrie algorithmique	111
4.2.3.2	Diagramme de Voronoi, triangulation de Delaunay en géométrie discrète et EDT	112
4.2.4	EDT en dimension 2 basée sur le diagramme de Voronoi discret	113
4.2.5	EDT en dimension 3 basée sur le diagramme de Voronoi discret	116
4.2.6	EDT en dimension n basée sur le diagramme de Voronoi discret	119
4.2.6.1	Analyse théorique	120
4.2.6.2	Description de l'algorithme	123
4.2.7	Analyse expérimentale	124
4.2.8	Diagramme de Voronoi discret et transformée en distance sur des grilles non-cubiques	129
4.2.9	Conclusion et perspectives	133
4.3	Squelette de forme et axe médian	133
4.3.1	Axe médian et squelette en géométrie discrète	133
4.3.2	Squelette n -dimensionnel basé sur le diagramme de Voronoi discret	135
4.3.3	Conclusion et discussion	137
4.4	Visibilité et géodésiques discrètes	142
4.4.1	Visibilité dans des domaines discrets	143
4.4.1.1	Notations et définitions	143
4.4.1.2	Domaine de visibilité	144
4.4.1.3	Algorithme de calcul de visibilité	147
4.4.2	Plus court chemin discret et métrique géodésique discrète	150

4.4.2.1	Définitions et état de l'art	150
4.4.2.2	Algorithme d'étiquetage en distance géodésique . .	151
4.4.3	Analyse expérimentale et discussion	152
4.4.4	Vers une extension en 3D	154
4.4.5	Conclusion et perspectives	157
4.5	Conclusion sur ce chapitre	157

4.1 Introduction

4.2 Transformée en distance

En analyse de formes binaires, la notion de transformée en distance est un outil classique qui est apparu avec les tous premiers algorithmes (ROSENFELD et PFALTZ, 1966, 1968). L'énoncé de cette transformation est la suivante : on étiquette chaque point de l'objet par la distance entre celui-ci et le point le plus proche du complémentaire. Bien évidemment, cette notion est liée à la métrique utilisée. Pour des raisons algorithmiques, les premières études considéraient des métriques approchées basées sur des masques de poids appelés masques de chanfrein. Il est à noter que plus la métrique utilisée se rapproche de la métrique euclidienne, plus la description de la forme sera invariante en rotation. En d'autres termes, l'utilisation de la métrique euclidienne permet de palier l'anisotropie intrinsèque de la grille discrète.

Pour la caractérisation de forme, on extrait de cette transformée en distance un objet géométrique appelé *squelette* ou *axe médian* qui correspond au plus petit sous-ensemble de points de l'objet qui permet de reconstruire la forme en appliquant le processus de transformée en distance inverse (voir ROSENFELD et PFALTZ 1966). En géométrie discrète, le terme *squelette* pour décrit généralement un objet topologique homotope à la forme (nous reviendrons sur ce point plus tard). Dans ce qui suit et quand nous ne précisons pas, nous parlerons toujours de *squelette* géométrique.

Ce *squelette* ou *axe médian* d'une forme correspond à la notion d'axe de symétrie généralisée (BLUM, 1967). Nous reviendrons sur la construction de cet objet dans le paragraphe 4.3.

Nous commençons cette partie par quelques rappels sur la transformée en distance basée sur des métriques de Chanfrein et sur la métrique euclidienne. Puis nous présentons un algorithme de calcul de transformée en distance euclidienne exacte optimal en temps pour des images n -dimensionnelles basé sur le calcul du diagramme de Voronoi discret.

4.2.1 Métriques de chanfrein

Dans un premier temps, l'idée est d'approximer la métrique euclidienne par des estimations locales de manière à obtenir des algorithmes de transformée en distance très rapides. Par exemple, ROSENFELD et PFALTZ (1966) considéraient des distances d_4 et d_8 sur la grille, plus tard ROSENFELD et PFALTZ (1968) proposaient de combiner ces distances pour se rapprocher de la distance euclidienne.

La distance d_4 (resp. d_8) est caractérisée par des poids affectés à 1 pour chaque 4-voisin (resp. pour chaque 8-voisin). Ainsi, la *boule* (ensemble de points à une distance inférieure à un certain seuil) construite sur cette métrique est un carré penché (respectivement un carré). La figure 4.1 représente les boules de rayon 100 associées à chaque métrique avec une palette de niveaux de gris cyclique.

Le cadre théorique et pratique pour le calcul des poids d'un masque de Chanfrein est le suivant : on commence par identifier un ensemble de déplacements élémentaires et on essaye d'optimiser les poids pour minimiser l'erreur par rapport à la distance euclidienne et pour que ceux-ci forment une distance (inégalité triangulaire ...). La description de ce processus général a été montré par BORGEFORS (1986); VERWER (1991); THIEL (1994, 2001); REMY (2001). Ce processus général est ensuite appliqué sur les grilles par exemple tridimensionnelle (KIRYATI et KUBLER, 1995; REMY et THIEL, 2000; BORGEFORS et SVENSSON, 2001) ou pour des grilles anisotropiques

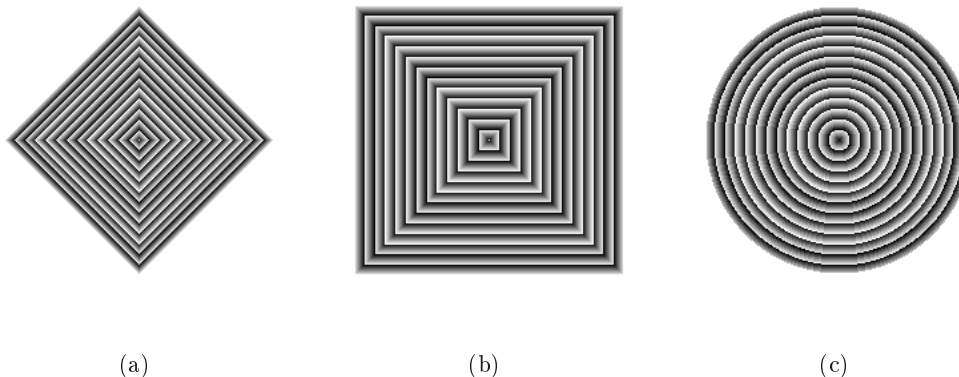


FIG. 4.1 – : Boules de rayon 100 pour les métriques : (a) d_4 (ou distance de Mahanatan), (b) d_8 (ou distance de l'échiquier) et (c) distance euclidienne.

(SINTORN et BORGEFORS, 2002).

L'intérêt majeur de ces approches est la simplicité de l'algorithme pour le calcul de la transformée en distance (notée WDT pour *Weighted Distance Transform*).

4.2.2 Transformée en distance euclidienne

Nous nous intéressons maintenant à la transformée en distance avec une métrique euclidienne (notée EDT pour *Euclidean Distance Transform*). Cette partie s'inscrivant dans le cadre d'une approche discrète, nous souhaitons ne manipuler que des nombres entiers et donc soit uniquement le vecteur à coordonnées entières entre le point de l'objet considéré et son plus proche point dans le complémentaire, soit la distance euclidienne au carré.

4.2.2.1 Approche par déplacements signés

Dans cette approche, DANIELSSON (1980) propose un balayage de l'image effectué en deux passes avec des opérations locales pour mettre à jour le vecteur entre le point considéré et le point du complémentaire le plus proche (on parle alors de VDT pour *Vector Distance Transform*). Cet algorithme considère donc un certain voisinage (4- ou 8-voisins) mais ce choix introduit des erreurs dans l'étiquetage (voir figure 4.2 pour le cas 4-connexe, un schéma équivalent existe pour le cas 8-connexe). Ainsi, cet algorithme est linéaire en le nombre de pixels de la grille ($O(n^2)$ pour une image $n \times n$) mais l'étiquetage n'est pas sans erreur.

YAMADA (1984) a proposé une version parallèle de cet algorithme qui permet d'éviter ces erreurs. Par la suite, RAGNEMALM (1990) a proposé un algorithme séquentiel pour la transformée en distance euclidienne exacte mais avec une complexité qui, dans le pire cas, est en $O(n^3)$ pour une image $n \times n$. Dans cet algorithme, RAGNEMALM propose un parcours de pixels, non plus par balayage de l'image mais par propagation de fronts d'ondes. Dans le même cadre, CUISENAIRE (1999) propose un algorithme rapide basé sur une propagation en distance euclidienne. Cette méthode n'est pas sans erreur mais l'utilisation de plusieurs voisinages dans la résolution permet de réduire l'erreur d'approximation.

Une version corrigée de l'algorithme de DANIELSSON, optimale en temps a été

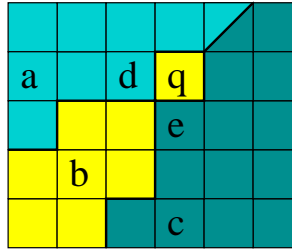


FIG. 4.2 —: *Erreur de la transformée de DANIELSSON sur un voisinage en 4-connexité (algorithme 4SED de DANIELSSON 1980.)* : les points a , b et c sont des points du complémentaire et q , d et e des points de l'objet. Lors de la propagation en 4-connexité, d est plus proche de a que de b , c et e est plus proche de c , q sera noté "plus proche de a ou de c " alors que le résultat exact serait b .

proposé par CUISENAIRE et MACQ (1999). Dans cette approche, on effectue un post-traitement sur la transformée en distance de DANIELSSON pour corriger les erreurs de la figure 4.2 qui sont facilement détectables dans la transformée (de manière intuitive, ce sont des *coins* dans le diagramme de Voronoi).

Des extensions pour des images 3D existent, par exemple MULLIKIN (1992) pour la version 3D de l'approche de DANIELSSON et CUISENAIRE a proposé une approche par propagation de l'algorithme précédent.

4.2.2.2 Approche multi-dimensionnelle de SAITO et TORIWAKI

SAITO et TORIWAKI (1994) ont proposé une approche basée sur la manipulation de distance au carré qui offre l'avantage d'être simple et valide quelle que soit la dimension de l'image.

Nous illustrons cet algorithme dans le cas 3D. Pour cela, nous notons \mathcal{P} l'ensemble des points composant le complémentaire de l'objet dans une image $[1 \dots m_1] \times [1 \dots m_2] \times [1 \dots m_3]$. L'objectif est de calculer la transformée en distance $S = \{s_{ijk}\}$ telle que s_{ijk} est la distance au carré entre le point (i, j, k) et le point de \mathcal{P} le plus proche. L'écriture de cette transformée est la suivante, pour tout point q de coordonnées (i, j, k) :

$$\begin{aligned} s_q &= \min_{p \in \mathcal{P}} \{dist^2(p, q)\} \\ &= \min_{p(x,y,z) \in \mathcal{P}} \{(i-x)^2 + (j-y)^2 + (k-z)^2\} \end{aligned}$$

Cette simple écriture ne nous permet pas de résoudre le problème mais nous laisse envisager une résolution dimension par dimension, à savoir une minimisation sur x , puis y et enfin z . Pour le cas 3D, nous obtenons l'algorithme suivant :

1. Construction, à partir de l'image source et de \mathcal{P} , d'une transformée en distance au carré dans selon l'axe des x . Nous notons cette transformée $G = \{g_{ijk}\}$, donnée par :

$$g_{ijk} = \min_{p(x,y,z) \in \mathcal{P}} \{(i-x)^2\}$$

2. A partir de G , nous construisons la transformée notée $H = \{h_{ijk}\}$ en utilisant une procédure de minimisation selon l'axe des y . G est donnée par :

$$h_{ijk} = \min_y \{g_{iyk} + (j-y)^2, 1 \leq y < m_2\}$$

- Enfin, nous construisons la transformée en distance au carré finale en appliquant une autre procédure de minimisation selon l'axe des z :

$$s_{ijk} = \min_z \{h_{ijz} + (k - z)^2, 1 \leq z < m_3\}$$

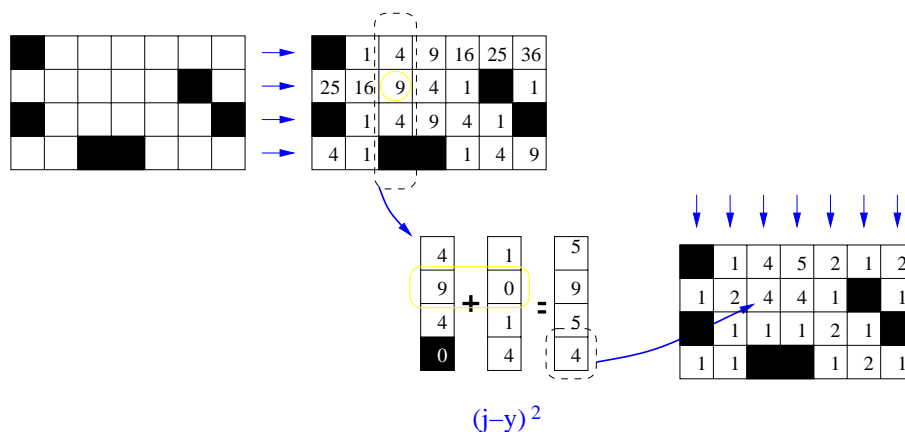


FIG. 4.3 –: Illustration de l'algorithme de SAITO et TORIWAKI en 2D : de gauche à droite, l'image source où les pixels noirs sont les pixels du complémentaire, la transformée G après la première passe, et la transformée finale après minimisation selon l'axe des y

Cet algorithme très simple est directement extensible aux images de dimension n . Dans ce cas, on applique la première phase calculant une transformée en distance au carré unidimensionnelle (étape 1 dans l'algorithme ci-dessus), puis pour chaque dimension supérieure, on minimise la transformée partielle calculée à l'étape précédente sommée avec des *paraboles* (fonctionnelles $(j - y)^2$ et $(k - z)^2$) dans la dimension actuelle (étape 2 et 3 de l'algorithme ci-dessus). La figure 4.3 illustre cet algorithme pour le cas 2D.

En terme de complexité, nous pouvons remarquer que pour une image 3D, la première étape s'effectue en temps linéaire en le nombre de points de l'objet. Cependant, l'opération de minimisation pour chaque dimension supérieure implique un coût supplémentaire. Ainsi, les étapes 2 et 3 ont chacune une complexité en $O(n^4)$ pour une image $n \times n \times n$. Le coût final d'une transformée en distance pour une image en dimension d est de $O(n^{d+1})$. SAITO et TORIWAKI (1994) ont cependant proposé une optimisation qui réduit la complexité en $O(Av(\sqrt{s_{ijk}})n^3)$ où $Av(\sqrt{s_{ijk}})$ représente la distance moyenne observée dans la transformée en distance de la forme.

Par la suite, HIRATA (1996) et MEIJSTER ET AL. (2000) vont indépendamment optimiser l'opération de minimisation rendant le calcul optimal, c'est-à-dire linéaire en le nombre de points de la grille et donc en $O(n^d)$ pour une image d -dimensionnelle.

4.2.3 Diagramme de Voronoi et transformée en distance euclidienne

Dans cette partie, nous proposons une autre approche pour le calcul de la transformée en distance euclidienne basée sur la construction du diagramme de Voronoi discret. Nous présentons un algorithme qui est aussi optimal en temps pour des images n -dimensionnelles. Cet algorithme construit une structure géométrique plus complexe qui permet de proposer des algorithmes de transformée en distance sur des

grilles anisotropiques ainsi que sur des grilles d'hexagones. De plus, nous verrons dans la partie 4.3 comment cette structure permet un calcul direct de l'axe médian de la forme.

4.2.3.1 Diagramme de Voronoi et triangulation de Delaunay en géométrie algorithmique

En géométrie algorithmique (non discrète), le diagramme de Voronoi d'un ensemble de sites $\mathcal{P} = \{p_i\} \subset \mathbb{R}^n$ est une partition de l'espace en cellules, chacune étant associée à un site avec la propriété suivante : un point q appartient à la cellule \mathcal{C}_{p_i} associée au site p_i si et seulement si $\text{dist}(q, p_i) < \text{dist}(q, p_j)$ ¹ pour tous les points $p_j \in \mathcal{P}$ et $i \neq j$ (voir figure 4.4-(a) pour un diagramme dans \mathbb{R}^2). Pour être plus précis, le diagramme de Voronoi est une partition de \mathbb{R}^n en cellules au sens des complexes simpliciaux (décomposition en points, arêtes, faces, hyper-faces, . . . , cellules de dimension n).

Ce diagramme a été très étudié et des bornes sur la complexité des algorithmes de construction existent. Par exemple en 2D, la borne inférieure est en $\Omega(|\mathcal{P}|\log|\mathcal{P}|)$ (voir PREPARATA et SHAMOS 1985).

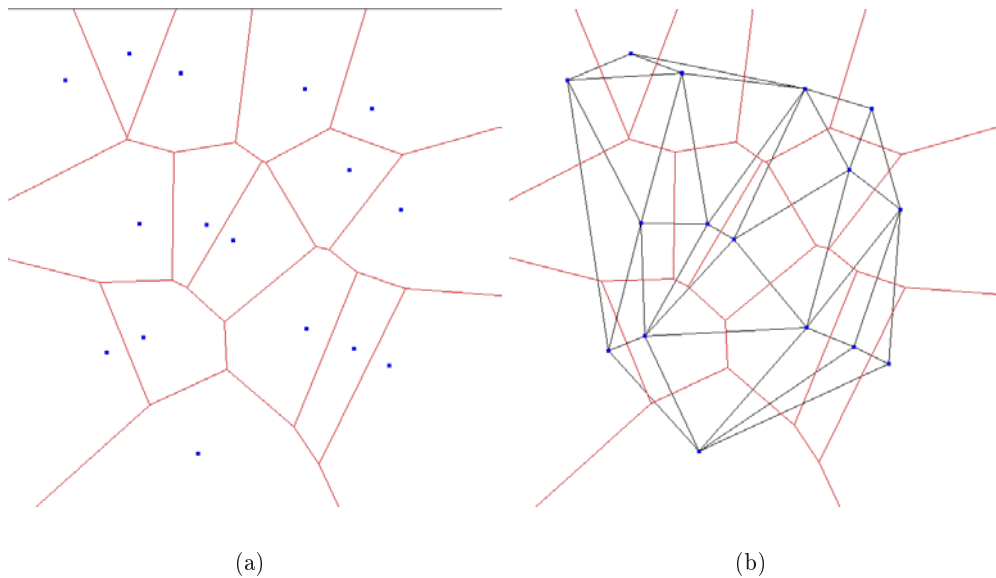


FIG. 4.4 –: (a) diagramme de Voronoi dans le plan et (b) triangulation de Delaunay (capture d'écran de VoroGlide, ICKING ET AL. 2001).

Si l'on considère un ensemble de sites en *position générale*, c'est à dire tels qu'il n'existe aucun quadruplet de sites cocycliques, de nombreuses propriétés sur ce graphe peuvent être prouvées (les lecteurs pourront se référer par exemple à PREPARATA et SHAMOS 1985 , BOISSONNAT et YVINEC 1995 ou encore à DE BERG ET AL. 2000). Par exemple, chaque sommet du diagramme est incident à exactement trois cellules.

Si nous considérons le graphe dual de ce diagramme, c'est à dire le graphe tel que les sommets sont des sites, et les arêtes, des adjacences entre cellules du diagramme, nous construisons alors une triangulation des sites qui est appelée *triangulation de*

1. $\text{dist}(p, q)$ correspond à la distance euclidienne.

Delaunay (voir figure 4.4-(b)). Cette triangulation a aussi de nombreuses propriétés en terme *d'aspect* des triangles générés. Plus formellement, si on trie tous les angles des triangles d'une triangulation \mathcal{T} des sites, dans un tableau $\mathcal{A}(\mathcal{T})$, la triangulation de Delaunay noté \mathcal{D} est minimale, c'est à dire $\mathcal{A}(\mathcal{D}) \preceq \mathcal{A}(\mathcal{T})$ en considérant l'ordre lexicographique pour la relation \preceq .

4.2.3.2 Diagramme de Voronoi, triangulation de Delaunay en géométrie discrète et EDT

Nous définissons une version discrète très simple du diagramme de Voronoi et de la triangulation de Delaunay qui correspond au plongement de ces structures géométriques euclidiennes dans l'espace discret.

Définition 4.1 (Diagramme de Voronoi discret) *Soit $P = \{p_i\}$ un ensemble de sites de l'espace discret, le diagramme de Voronoi discret est une partition de l'espace discret en sous-ensembles $\{c_i\}$, tels que chaque point de c_i soit plus proche de p_i que de chacun des $\{p_j\}$ avec $j \neq i$.*

Dans ce qui suit, nous ne nous intéressons pas à la topologie associée au diagramme de Voronoi mais uniquement à sa construction. Cette définition ne donne pas un diagramme unique, en effet, dans la cas où un point est équidistant de deux sites, il est arbitrairement associé à un des deux sites. Par contre, lorsqu'un point discret est associé à un site p_i , nous savons que ce point appartient à l'adhérence de la cellule c_i dans le diagramme euclidien des sites P .

A partir de la définition 4.1, nous pouvons de la définir une triangulation de Delaunay discrète basée sur les notions de droites discrètes.

Définition 4.2 (Triangulation de Delaunay discrète) *Soit $P = \{p_i\}$ un ensemble de points discrets, la triangulation de Delaunay discrète est telle que ses sommets sont les points $\{p_i\}$ et un segment discret existe, dans l'espace considéré, entre deux sommets si ces deux sommets engendrent des cellules adjacentes dans le diagramme de Voronoi discret associé à P .*

Les liens entre le diagramme de Voronoi et la transformée en distance ont été montrés par BREU ET AL. (1995). Si nous considérons comme ensemble de sites P , l'ensemble des points discrets du complémentaire de l'objet discret considéré, le diagramme de Voronoi discret de P nous donne, en tout point de l'objet, le point de P qui lui est le plus proche. Ainsi, pour obtenir la transformée en distance, il suffit d'effectuer un post-traitement sur le diagramme en étiquetant tous les points discrets par la distance au point de P le plus proche donné par le diagramme. Ce post-traitement se fait en temps linéaire en le nombre de points discret de l'objet.

Résoudre le problème du calcul du diagramme de Voronoi permet donc de résoudre le problème du calcul de la transformée en distance euclidienne. Nous présentons dans le paragraphe suivant le calcul de transformée en distance basé sur le diagramme de Voronoi proposé par BREU ET AL. pour le cas d'une grille régulière en dimension 2 pour ensuite l'étendre en nD et sur des grille non nécessairement cubiques.

4.2.4 EDT en dimension 2 basée sur le diagramme de Voronoi discret

BREU ET AL. (1995) ont proposé un algorithme très élégant et très efficace pour le calcul du diagramme de Voronoi discret et donc pour le calcul de la transformée en distance euclidienne sans erreur dans le cas d'une image binaire 2D notée $[0 \dots m_1] \times [0 \dots m_2]$.

Dans un premier temps, cet algorithme effectue un balayage des lignes de l'image de $y = 0$ à $y = m_2$. A chaque étape $y = r$ de l'algorithme, BREU ET AL. maintiennent une structure de données qui contient le diagramme de Voronoi discret \mathcal{V}_{P_r} des sites P_r . Ces sites correspondent aux points de P dont l'ordonnée est inférieure ou égale à r (voir figure 4.5 pour les notations de BREU ET AL.).

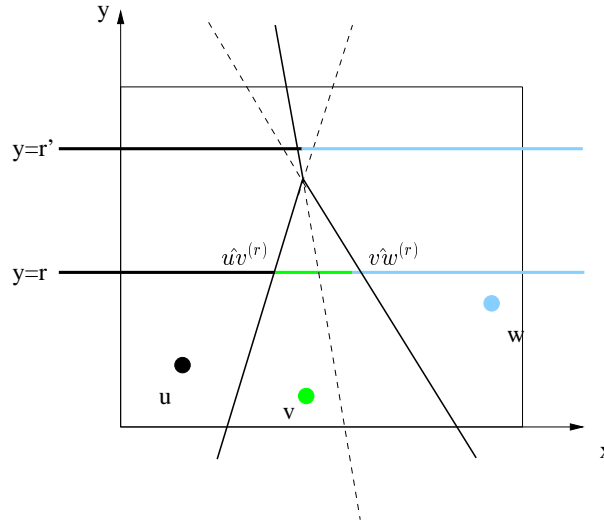


FIG. 4.5 – : Notations de BREU ET AL. : u , v et w sont des sites appartenant à P_r et $P_{r'}$, $\hat{u}v^{(r)}$ et $\hat{v}w^{(r)}$ sont les points intersections de la médiatrice de $[uw]$ et $[vw]$ respectivement, et de la droite $y = r$. Dans cet exemple, nous avons $\mathcal{L}_r = \{u, v, w\}$ et $\mathcal{L}_{r'} = \{u, w\}$ (voir le texte pour les explications).

A la fin de ce parcours, un autre balayage de $y = m_2$ à $y = 0$ est effectué pour fusionner les résultats et ainsi construire le diagramme de Voronoi à partir de diagrammes partiels (voir figure 4.6).

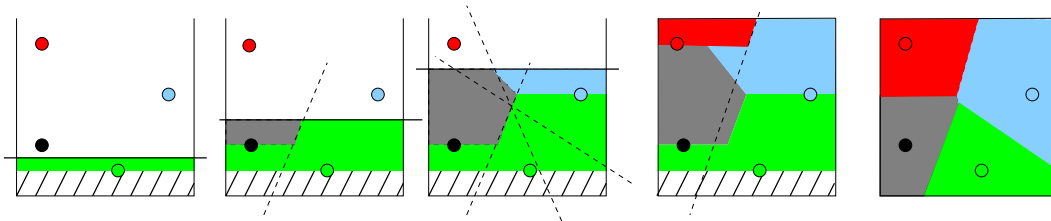


FIG. 4.6 – : Balayage de l'image en y croissant (4 premières images) avec les diagrammes de Voronoi partiels. Chaque cellule a la couleur du site auquel elle correspond, la partie hachurée correspond aux pixels non-étiquetés et les droites en pointillés sont des médiatrices entre sites.

Dans la suite, nous réutilisons les notations de BREU ET AL. : on note \mathcal{L}_r le sous-ensemble de sites P_r dont les cellules intersectent la droite $y = r$ (voir figure

4.5). Ainsi, le coeur de l'algorithme réside en un calcul efficace des sites de \mathcal{L}_r à partir des sites de \mathcal{L}_{r-1} et des nouveaux sites présents dans la ligne $y = r$ de l'image.

Cette propagation efficace est basée sur deux observations : la première est qu'il ne peut y avoir dans \mathcal{L}_r qu'un seul site par abscisse. Dans le cas contraire, la médiatrice issue des deux sites de même abscisse, soit n'intersectera pas la droite $y = r$, soit sera confondue avec celle-ci. La seconde observation est que si les sites de P_r sont triés selon leur abscisse, cet ordre est alors préservé sur la droite $y = r$ pour les cellules issues des sites de \mathcal{L}_r .

La première observation a pour conséquence que la liste \mathcal{L}_r est telle que :

$$|\mathcal{L}_r| \leq m_1$$

La seconde nous dit que les cellules de \mathcal{L}_{r-1} apparaîtront (si elles sont conservées) dans le même ordre dans \mathcal{L}_r .

Avant de détailler l'algorithme, nous présentons les conditions pour qu'un site de \mathcal{L}_{r-1} ne soit plus présent dans la liste \mathcal{L}_r . Considérons pour cela trois sites $u, v, w \in \mathbb{Z}^2$ de l'image tels que² $u_x < v_x < w_x$. On dira que le site v est *caché par u et w le long de la droite $y = r$* si la cellule associée à v n'intersecte pas la droite $y = r$ alors que c'est le cas pour les cellules associées à u et w . Cette propriété est illustrée dans la figure 4.5 pour la droite $y = r'$.

En se basant sur cette définition, nous pouvons formaliser l'approche de BREU ET AL. avec la proposition suivante :

Proposition 4.1 *Étant donné un site v d'un ensemble \mathcal{P} de sites et une droite l , v n'appartient pas à la liste \mathcal{L}_l si et seulement si il existe au moins une paire de sites u et w de $\mathcal{P} \setminus \{v\}$ tels que u et w cachent v le long de l .*

Preuve : Cette proposition est illustrée par la figure 4.7. L'idée est la suivante : si la cellule associée à v n'intersecte pas l , alors on trouve une paire de sites qui cachent v sur l en translatant l jusqu'au sommet de C_v le plus proche (droite l' dans la figure 4.7). Ainsi, si les sites sont en position générale (pas quatre points cocycliques), il n'existe que deux sites adjacents à ce sommet de C_v , ici u et w , et ces deux sites cachent v sur l .

Si plusieurs sites sont adjacents au sommet de C_v considéré, il suffit d'en choisir deux et ceux-ci vérifient la propriété de masquage de v .

La preuve de la proposition dans l'autre sens de l'équivalence logique est triviale : si u et w cachent v , alors $C_v \cap l = \emptyset$ par définition. \square

En d'autres termes, chaque *disparition* de sites dans les listes \mathcal{L}_r est dirigée par le fait qu'un site est caché par deux autres sites. Il nous faut donc un prédicat permettant de détecter ces cas.

Si on note $\hat{u}v^{(r)}$ l'intersection entre la médiatrice de $[uv]$ et la droite $y = r$, v est *caché* par u et w (avec toujours $u_x < v_x < w_x$) si et seulement si l'abscisse de $\hat{u}v^{(r)}$ est supérieure à l'abscisse de $\hat{v}w^{(r)}$. Quand il n'y a pas d'ambiguïté, et pour simplifier les notations, on écrira $\hat{u}v$ plutôt que $\hat{u}v^{(r)}$. Formellement, on a donc :

$$\hat{u}v_x \geq \hat{v}w_x \tag{4.1}$$

2. dans la suite, les coordonnées d'un point p sont notées (p_x, p_y) si $p \in \mathbb{R}^2$ et (p_x, p_y, p_z) si $p \in \mathbb{R}^3$

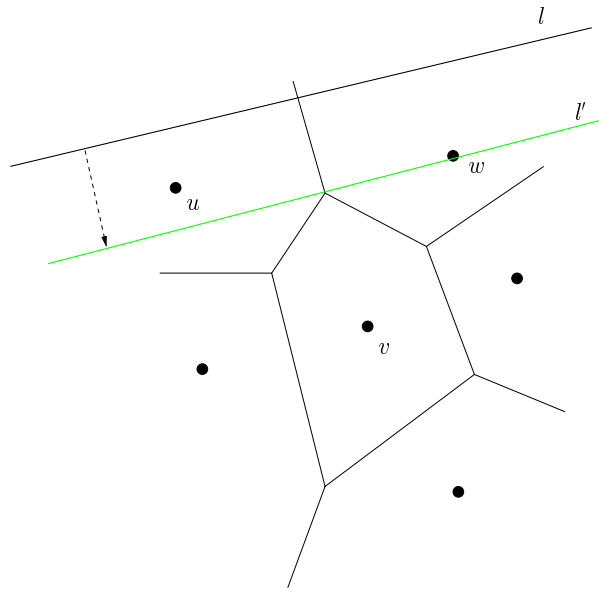


FIG. 4.7 – : *Illustration de la preuve : C_v n'intersecte pas l et on calcule une paire de sites cachant v en translatant la droite l vers un sommet de C_v . Les sites adjacents à ce sommet du diagramme sont solutions du problème.*

Si nous détaillons les coordonnées de chaque élément, $\hat{u}v$ est donné par :

$$\|u - \hat{u}v\|^2 = (u_x - \hat{u}v_x)^2 + (u_y - r)^2 \quad (4.2)$$

$$\|v - \hat{u}v\|^2 = (v_x - \hat{u}v_x)^2 + (v_y - r)^2 \quad (4.3)$$

Finalement, l'équation 4.1 est équivalente à (BREU ET AL., 1995) :

$$(w_x - v_x)(v_x^2 - 2r(v_y - u_y) + v_y^2 - u_y^2) \geq (v_x - u_x)(w_x^2 - v_x^2 - 2r(w_y - v_y) + w_y^2 - v_y^2) \quad (4.4)$$

Nous pouvons donc écrire maintenant un prédicat $\text{CACHÉ_PAR}(U, V, W, R)$ qui permet de décider si u et w cachent v sur $y = r$ (voir algorithme 16).

Algorithme 16 Prédicat CACHÉ_PAR de BREU ET AL. (1995)

$\text{CACHÉ_PAR}(u, v, w, r)$ { v est-il caché par u et w sur la droite $y = r$?}

1: **retourne** $((w_x - v_x)(v_x^2 - u_x^2 - 2r(v_y - u_y) + v_y^2 - u_y^2) \geq (v_x - u_x)(w_x^2 - v_x^2 - 2r(w_y - v_y) + w_y^2 - v_y^2))$

Avec ce prédicat et la seconde observation de BREU ET AL., nous pouvons maintenant présenter le coeur de l'algorithme qui est la procédure de suppression des sites d'une liste de sites candidats susceptibles d'appartenir à la liste \mathcal{L}_r étant donnée une droite $y = r$. Dans la suite nous notons Candidat_r cette liste de sites possibles. En pratique, cette liste est composée des sites de \mathcal{L}_{r-1} et des sites présents sur la droite $y = r$. Autrement dit :

$$\text{Candidat}_r = \mathcal{L}_{r-1} \cup \{p \in P \mid p_y = r\}$$

Si les sites de Candidat_r sont triés par abscisse croissante, il suffit d'appliquer le prédicat CACHÉ_PAR de manière gloutonne avec une structure de pile nous permettant de dépiler les sites qui seraient cachés par d'autres (voir algorithme 17). Nous obtenons ainsi un algorithme dont le coût est linéaire en le nombre de sites dans Candidat_r .

Remarquons que cet algorithme est similaire à celui utilisé par GRAHAM pour le calcul d'enveloppe convexe (voir CORMEN ET AL. 1990; DE BERG ET AL. 2000).

Algorithme 17 Procédure de suppression de sites de BREU ET AL. (1995)

```

SUPPRESSION_SITES(Candidatr, r)
1:  $\mathcal{L}_r[1] = \text{Candidat}_r[1]$ 
2:  $\mathcal{L}_r[2] = \text{Candidat}_r[2]$ 
3:  $k=2, l=3, c=|\text{Candidat}_r|$ 
4: tant que  $l \leq c$  faire
5:    $w = \text{Candidat}_r[l]$ 
6:   tant que  $k \geq 2$  et CACHÉ_PAR( $\mathcal{L}_r[k-1], \mathcal{L}_r[k], w, r$ ) faire
7:      $k = k - 1$ 
8:   fin du tant que
9:    $k = k + 1, l = l + 1$ 
10:   $\mathcal{L}_r[k] = w$ 
11: fin du tant que

```

Or, d'après la première observation, la liste \mathcal{L}_r a une taille bornée par m_1 et par conséquent c'est aussi le cas pour la liste Candidat_r . De plus, la construction de cette liste à partir de \mathcal{L}_{r-1} se fait en $O(m_1)$.

Le passage de \mathcal{L}_{r-1} vers \mathcal{L}_r se fait donc en $O(m_1)$. Comme uniquement deux balayages sont nécessaires, nous obtenons une complexité finale pour la construction du diagramme de Voronoi discret et de l'EDT en $O(m_1 m_2)$, c'est-à-dire linéaire en le nombre de pixels de l'image ; ce qui est optimal pour le problème de la transformée en distance.

Récemment, GUAN et MA (1998) ont présenté une optimisation de cet algorithme en considérant une approche basée sur des listes, et ramène la complexité en $O(\sqrt{m_1 m_2 t})$ où t représente le nombre de sites mais l'algorithme nécessite toujours une étape d'initialisation en $O(m_1 m_2)$.

Dans le paragraphe suivant, nous détaillons une version tridimensionnelle de cet algorithme avant de l'étendre finalement à des images en nD .

4.2.5 EDT en dimension 3 basée sur le diagramme de Voronoi discret

Dans la suite, une image \mathcal{I} en n dimensions sera un sous-ensemble de \mathbb{Z}^n noté $[1, m_1] \times \dots \times [1, m_n]$. Nous formalisons tout d'abord la notion de *tube* dans une grille de \mathbb{Z}^n . La notion de tube est une extension en nD des lignes, colonnes et hauteurs en 2D/3D.

Définition 4.3 (tube selon la $k^{\text{ème}}$ dimension dans une image en dimension n)

Un tube est un ensemble de m_k points du domaine $[1, m_1] \times \dots \times [1, m_n]$ tel que les coordonnées des points ne diffèrent que dans la k dimension. Ce tube est noté :

$$T_{i_1, \dots, i_{k-1}, \bullet, i_{k+1}, \dots, i_m} = \{(i_1, \dots, i_{k-1}, i_k, i_{k+1}, \dots, i_n) \in \mathcal{I} \mid 1 \leq i_k \leq m_k\}$$

Dans l'approche multidimensionnelle de SAITO et TORIWAKI, les fusions de distances sont calculées le long de *tubes* orthogonaux.

L'idée de la version 3D de l'algorithme de transformée en distance basée sur le diagramme de Voronoi discret est la suivante : utiliser la décomposition en tubes et construire les diagrammes partiels dans chaque dimension.

Dans SAITO et TORIWAKI (1994), les auteurs ont présenté un algorithme dérivé de leur calcul de transformée en distance pour construire un tel diagramme. Dans notre approche, nous proposons une construction basée sur l'approche de BREU ET AL. permettant ainsi une construction en temps optimal ainsi qu'une extensibilité pour des grilles non-cubiques.

Cette approche se décompose donc en trois étapes, une par dimension :

Étape 1 : lors de cette première étape, nous calculons un diagramme de Voronoi de dimension 1 dans les tubes parallèles à l'axe des x . En d'autres termes, nous étiquetons tous les pixels de l'objet par le plus proche pixel du complémentaire ayant les mêmes coordonnées en y et en z . Si un point de l'objet est équidistant de deux sites, nous choisissons arbitrairement un représentant, ce choix n'aura aucune importance pour la transformée en distance.

Ce calcul de diagramme de Voronoi unidimensionnel se fait en deux parcours de chaque tube et donc le coût total de cette première étape est en $O(m_1 m_2 m_3)$ (voir figure 4.8-(b)). Dans le cas particulier où il n'y a pas de site de même y et de même z , nous utilisons une étiquette spéciale notée \emptyset .

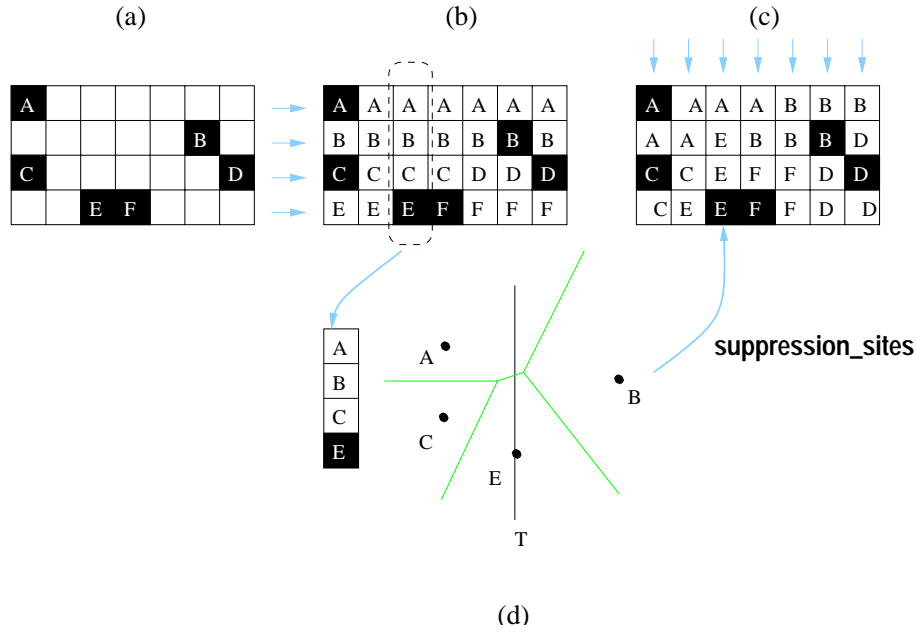


FIG. 4.8 — *Illustration du calcul 2D : les sites sont étiquetés par une lettre et on souhaite étiqueter les pixels de l'objet par le site le plus proche. (a) l'image binaire (les cases noires représentent les sites), (b) résultat après les deux premiers balayages en ligne, (c) résultat après suppression des sites dans les colonnes. (d) détail de la procédure de suppression : C est caché par A et E, B est caché par A et E le long de la colonne considérée.*

Étape 2 : l'objectif est ici de construire des diagrammes de Voronoi discrets dans chaque coupe (x, y) de l'image 3D. Pour cela, nous considérons un tube T défini par :

$$T_{r,\bullet,t} = \{(x, y, z) \in \mathcal{I} \mid x = r, z = t, y \in [1 \dots m_2]\}$$

Chaque case (r, y, t) de ce tube contient les coordonnées du site le plus proche dans chaque ligne (au sens axe des x). Nous appellerons dans la suite, les sites

présents dans ce tube des *candidats*. En d'autres termes, ces sites sont utilisés dans les diagrammes 1D et sont candidats pour les diagrammes 2D définis dans les coupes.

Ainsi, nous avons une liste de candidats telle que $|Candidat| \leq m_2$ et nous pouvons faire les mêmes remarques que BREU ET AL. :

- nous n'avons qu'un seul candidat par y (par construction à l'étape 1) ;
- l'ordre selon les ordonnées des sites dont les cellules intersectent le tube dans le diagramme final est le même que celui des candidats du tube.

Nous pouvons donc appliquer la procédure de suppression des sites de BREU ET AL. donné par l'algorithme 17, de manière à supprimer de la liste *Candidat* les sites dont les cellules vont disparaître du tube T (voir figure 4.8). Il nous faut cependant présenter un nouveau prédicat CACHÉ_PAR_2D étant donné que la suppression se fait verticalement et non horizontalement comme dans l'algorithme initial (voir algorithme 18).

Algorithme 18 Nouveau prédicat CACHÉ_PAR_2D

CACHÉ_PAR_2D(u, v, w, r) $\{v \text{ est-il caché par } u \text{ et } w \text{ sur la droite } x = r ?\}$
 1: **retourne** $((w_y - v_y)(v_y^2 - u_y^2 - 2r(v_x - u_x) + v_x^2 - u_x^2) \geq (v_y - u_y)(w_y^2 - v_y^2 - 2r(w_x - v_x) + w_x^2 - v_x^2))$

Nous appliquons donc l'algorithme 17 avec ce prédicat et nous construisons ainsi des diagrammes de Voronoi discrets dans chaque coupe de l'image. Étant donné que cette suppression sur un tube se fait en temps linéaire, c'est-à-dire en $O(m_2)$ selon nos notations, le coût global de cette étape est donc en $O(m_1 m_2 m_3)$.

Étape 3 : l'objectif de cette étape est de fusionner les différents diagrammes de Voronoi discrets 2D de chaque coupe pour construire le diagramme 3D.

Nous considérons donc le tube T' dans l'image défini par :

$$T'_{r,t,\bullet} = \{(x, y, z) \in \mathcal{I} \mid x = r, y = t, z \in [1 \dots m_3]\} \quad (4.5)$$

A chaque point (r, t, z) de ce tube, nous avons les coordonnées du site le plus proche du voxel (r, t, z) parmi les sites appartenant au plan parallèle à Oxy et de hauteur z . Les sites du tube T' définissent alors une liste de candidats pour le diagramme partiel 3D le long de T' .

Nous pouvons faire les mêmes observations que BREU ET AL. pour le cas 3D :

- nous n'avons qu'un seul candidat par z (c'est à dire un par coupe) ;
- l'ordre selon les coordonnées en "z" des sites dont les cellules intersectent le tube dans le diagramme final est le même que celui des candidats du tube.

L'idée est encore ici d'appliquer une suppression de sites sur ce tube (voir figure 4.9). Avant cela, il nous faut détailler le prédicat CACHÉ_PAR_3D qui sera utilisé. Celui-ci prend en arguments trois points $u, v, w \in \mathbb{Z}^3$ et le paramètre du tube 3D ($x = r, y = t$). Il décide si le site v est *caché* par u et w sur le long de la droite ($x = r, y = t$).

De la même façon que les équations (4.2), (4.3) et (4.4), nous avons :

$$\|u - \hat{u}v\|^2 = (u_x - r)^2 + (u_y - t)^2 + (u_z - \hat{u}v_z)^2 \quad (4.6)$$

$$\|v - \hat{u}v\|^2 = (v_x - r)^2 + (v_y - t)^2 + (v_z - \hat{u}v_z)^2 \quad (4.7)$$

Et donc :

$$\hat{u}v_z = \frac{v_z^2 - u_z^2 + (v_x - r)^2 + (v_y - t)^2 - (u_x - r)^2 - (u_y - t)^2}{2(v_z - u_z)} \quad (4.8)$$

Nous pouvons maintenant écrire le prédicat qui sera utilisé pour la suppression 3D (voir algorithme 19).

Algorithme 19 Prédicat CACHÉ_PAR_3D

CACHÉ_PAR_3D(u, v, w, r, t)

{ v est-il caché par u et w sur la droite ($x = r, y = t$) ?}

1: **retourne** $((w_z - v_z)(v_z^2 - u_z^2 + (v_x - r)^2 + (v_y - t)^2 - (u_x - r)^2 - (u_y - t)^2) \geq (v_z - u_z)(w_z^2 - v_z^2 + (w_x - r)^2 + (w_y - t)^2 - (v_x - r)^2 - (v_y - t)^2))$

En changeant uniquement le prédicat de l'algorithme 17, nous réutilisons celui-ci et obtenons ainsi un algorithme linéaire en la taille du tube, c'est-à-dire en $O(m_3)$, pour supprimer les sites non présents dans le diagramme final.

En appliquant ce processus pour tous les tubes, nous avons une complexité en $O(m_1 m_2 m_3)$ pour cette étape.

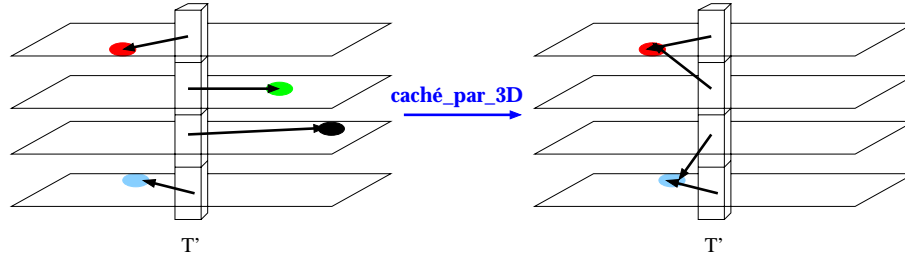


FIG. 4.9 – : *Illustration de l'étape 3 : dans chaque coupe, les voxels du tube sont étiquetés par le site le plus proche dans la coupe et on utilise la suppression de sites 3D pour fusionner les diagrammes.*

Coût global Étant donné que chaque étape de l'algorithme se fait en $O(m_1 m_3 m_3)$, le coût global du calcul du diagramme de Voronoi discret, et donc de la transformée en distance euclidienne 3D, se fait en $O(m_1 m_3 m_3)$ ce qui est donc linéaire en le nombre de voxels et optimal pour le problème.

L'algorithme global pour le cas 3D est présenté dans l'algorithme 20.

4.2.6 EDT en dimension n basée sur le diagramme de Voronoi discret

Dans ce paragraphe, nous généralisons l'approche proposée en 3D pour des images n -dimensionnelles permettant le calcul du diagramme de Voronoi discret et de la transformée en distance euclidienne en temps optimal. Nous prouvons aussi les différentes observations faites précédemment dans le cas nD .

Algorithme 20 Diagramme de Voronoi discret en 3DDIAGRAMME_VORONOI_DISCRET_3D(\mathcal{I})

```

    {Étape 1 : selon les lignes (axe  $Ox$ )}
1: pour  $j$  de 1 à  $m_2$  faire
2:   pour  $k$  de 1 à  $m_3$  faire
3:     Calcul du diagramme de Voronoi unidimensionnel dans le tube  $T_{\bullet,j,k}$ 
4:   fin du pour
5: fin du pour

    {Étape 2 : selon les colonnes (axe  $Oy$ )}
6: pour  $i$  de 1 à  $m_1$  faire
7:   pour  $k$  de 1 à  $m_3$  faire
8:     Appliquer la suppression de sites dans le tube  $T_{i,\bullet,k}$ 
       avec le prédicat CACHÉ_PAR_2D
9:   fin du pour
10: fin du pour

    {Étape 3 : selon les hauteurs (axe  $Oz$ )}
11: pour  $i$  de 1 à  $m_1$  faire
12:   pour  $j$  de 1 à  $m_2$  faire
13:     Appliquer la suppression de sites dans le tube  $T_{i,j,\bullet}$ 
       avec le prédicat CACHÉ_PAR_3D
14:   fin du pour
15: fin du pour

```

4.2.6.1 Analyse théorique

Introduisons tout d'abord quelques notations. Nous considérons par la suite un diagramme de Voronoi noté $\mathcal{V}_{\mathcal{P}}$ en dimension n , c'est à dire $\mathcal{P} \subset \mathbb{R}^n$. Nous considérons aussi une droite l dans \mathbb{R}^n et on note Π_l la projection orthogonale de points de \mathbb{R}^n sur l . Sur cette droite, nous allons considérer une origine, notée O , et un vecteur unitaire de manière à introduire un ordre des points de l . Nous noterons aussi $\{p_i\}_{1..n}$ les sites de \mathcal{P} et $\mathcal{Q} = \{q_i\} \subseteq \mathcal{P}$ l'ensemble des sites de \mathcal{P} dont la cellule associée intersecte la droite l . Étant données les propriétés de convexité des cellules de Voronoi, on dispose également d'un ordre des cellules de \mathcal{Q} le long de la droite l : l'intersection d'un convexe avec une droite est un segment. A part dans les cas *dégénérés*, ces segments sont totalement ordonnés.

Pour prouver les observations faites en 2D et en 3D (dans les étapes 2 et 3), il nous faut montrer les deux points suivants :

- si deux sites se projettent orthogonalement sur l au même point, l'un de ces deux sites peut être supprimé pour résoudre le problème de l'intersection de $\mathcal{V}_{\mathcal{P}}$ avec l (première observation de BREU ET AL.) ;
- l'ordre des cellules sur l des sites de \mathcal{Q} est compatible avec l'ordre des projections des sites $\{p_i\}$ sur l , notées $\Pi_l(p_i)$ (seconde observation de BREU ET AL.).

De manière intuitive, ces deux propriétés sont presque triviales mais il apparaît qu'une preuve formelle de celles-ci n'est pas évidente.

Tout d'abord, considérons deux points A et B de \mathcal{P} , notons H l'hyper-plan médiateur du segment $[AB]$ et \vec{l} un vecteur directeur de l . Le premier lemme nous permet d'enlever de \mathcal{P} les points pathologiques, c'est-à-dire ceux dont les projections sur l sont confondues. Par la suite et pour simplifier les notations, on utilisera π_A pour $\Pi_l(A)$ et π_B pour $\Pi_l(B)$. Le lemme s'énonce alors de la façon suivante :

Lemme 4.1 *Le plan euclidien \vec{H} associé au plan affine H contient \vec{l} si et seulement si $\pi_A = \pi_B$*

Preuve : nous prouvons tout d'abord l'implication de gauche à droite. Nous supposons donc que $\vec{l} \in \vec{H}$, et notons M le milieu de $[AB]$ et $\Pi_l(M) = \pi_M$. Nous avons alors :

$$\vec{OA} = \vec{O\pi_M} + \vec{\pi_M M} + \vec{MA}$$

Donc, si nous notons Π_l^* la projection vectorielle engendrée par Π_l , nous avons :

$$\begin{aligned} \Pi_l^*(\vec{OA}) &= \Pi_l^*(\vec{O\pi_M}) + \Pi_l^*(\vec{\pi_M M}) + \Pi_l^*(\vec{MA}) \\ &= \vec{O\pi_M} + \vec{0} + \vec{0} \end{aligned}$$

Nous avons $\Pi_l^*(\vec{MA}) = \vec{0}$ parce que \vec{AM} est orthogonal à tout vecteur appartenant à \vec{H} par définition, ainsi, comme $\vec{l} \in \vec{H}$, \vec{AM} est orthogonal à \vec{l} et donc $\Pi_l^*(\vec{MA}) = \vec{0}$.

De plus, nous avons de la même manière :

$$\Pi_l^*(\vec{OA}) = \Pi_l^*(\vec{OB}) = \vec{O\pi_M}$$

Et donc finalement, $\pi_A = \pi_B = \pi_M$.

Pour prouver l'implication inverse, supposons $\pi_A = \pi_B$. Nous avons donc $\vec{\pi_M A} \perp \vec{l}$ et $\vec{\pi_M B} \perp \vec{l}$ par définition de la projection. Ainsi, \vec{l} est orthogonal au plan dans \mathbb{R}^n engendré par (π_A, A, B) . En particulier, $\vec{l} \perp \vec{AB}$ et comme \vec{H} contient tous les vecteurs orthogonaux à \vec{AB} , \vec{l} appartient nécessairement à \vec{H} . \square

Nous notons Σ_A (resp. Σ_B) l'intersection entre la cellule de Voronoi associée à A (resp. B) et la droite l (Σ_A et Σ_B peuvent être nuls). A partir du lemme précédent, nous avons les corollaires suivants :

Corollaire 4.1 *Si $\pi_A \neq \pi_B$, l'intersection entre H et l est réduite en un point et les deux segments sur l , Σ_A et Σ_B , sont disjoints.*

La preuve se déduit directement du lemme 4.1.

Corollaire 4.2 *Si $\pi_A = \pi_B$ nous avons deux cas possibles :*

- soit $\text{dist}(A, l) \neq \text{dist}(B, l)$ et donc si $\Sigma_A \neq \emptyset$ (resp. $\Sigma_B \neq \emptyset$) alors $\Sigma_B = \emptyset$ (resp. $\Sigma_A = \emptyset$), et donc $B \notin \mathcal{Q}$ (resp. $A \notin \mathcal{Q}$) ;
- soit $\text{dist}(A, l) = \text{dist}(B, l)$ et dans ce cas $\Sigma_A = \Sigma_B$ (les ensembles peuvent être éventuellement nuls) et donc soit A ou B (arbitrairement) peut ne plus être considéré comme appartenant à \mathcal{Q} .

Preuve : Le premier point de ce corollaire est direct, le second nécessite quelques détails. Montrons que $\Sigma_A = \Sigma_B$.

Soit p un point de Σ_A . Par définition, p appartient à l . Or, l est incluse dans H d'après le lemme 4.1. La cellule de Voronoi C_A engendrée par A est entièrement du même côté de H donc p est sur la frontière de C_A . Comme le diagramme de Voronoi est une partition de l'espace, il existe un site X tel que p soit sur la frontière de C_X . A et X sont des sites plus proches de p que tout autre site, Or nous avons $dist(A, p) = dist(B, p)$ donc soit $X = B$, soit p est aussi sur la frontière de B . Quel que soit le cas de figure, p appartient à Σ_B . Nous obtenons $\Sigma_A \subseteq \Sigma_B$.

De la même manière, nous pouvons prouver que $\Sigma_B \subseteq \Sigma_A$ et donc que $\Sigma_A = \Sigma_B$. \square

Maintenant, étant donné un tube dans une image discrète nD , nous avons le corollaire :

Corollaire 4.3 Soit T un tube selon la k^{ime} dimension d'une image dans \mathbb{Z}^n , soit \mathcal{Q} la liste des sites de $p_i \in P$ (dans ce cas $P \subset \mathbb{Z}^n$) telle que $\mathcal{C}_{p_i} \cap T \neq \emptyset$, nous avons alors :

$$|\mathcal{Q}| \leq m_k \quad (4.9)$$

D'après le corollaire 4.2 et dans une optique de transformée en distance, nous pouvons considérer uniquement des sites de \mathcal{P} tels que $\pi_A \neq \pi_B$ (pour tout couple (A, B)).

Nous présentons maintenant le lemme montrant la préservation de l'ordre entre les cellules et la projection de sites sur l :

Lemme 4.2 Supposons $\mathcal{P} = \{A, B\}$, $\pi_A \neq \pi_B$, $a \in \Sigma_A$ et $b \in \Sigma_B$ ($a \neq b$), l'ordre des points (π_A, π_B) sur l est le même que (a, b) , ce qui s'écrit :

$$\vec{ab} = \kappa \overrightarrow{\pi_A \pi_B} \quad (\text{avec } \kappa > 0) \quad (4.10)$$

où \cdot est le produit scalaire de vecteurs.

Preuve : Par définition, A et B sont de part et d'autre du plan H et \overrightarrow{AB} est normal au plan. Or, a est du même côté de H que A , de même pour b qui est du même côté de H que B . Donc la composante suivant \overrightarrow{AB} du vecteur \vec{ab} est positive. En d'autres termes :

$$\vec{ab} \cdot \overrightarrow{AB} = \lambda \quad (\lambda > 0) \quad (4.11)$$

Par définition de la projection, la composante suivant $\overrightarrow{\pi_A \pi_B}$ du vecteur \overrightarrow{AB} est $\overrightarrow{\pi_A \pi_B}$. Ainsi :

$$\overrightarrow{\pi_A \pi_B} \cdot \overrightarrow{AB} = \|\overrightarrow{\pi_A \pi_B}\|^2 = \mu \quad (\mu > 0) \quad (4.12)$$

En remplaçant \vec{ab} par sa valeur (4.10) dans l'équation (4.11), nous obtenons

$$\kappa \overrightarrow{\pi_A \pi_B} \cdot \overrightarrow{AB} = \lambda$$

C'est-à-dire :

$$\kappa = \frac{\lambda}{\mu} > 0$$

\square

Nous pouvons maintenant conclure sur tous les points de \mathcal{P} en appliquant le lemme précédent sur chaque paire de points de \mathcal{P}

Corollaire 4.4 *Si les sites $\{p_i\}_{1..n}$ de \mathcal{P} sont tels que $\Pi_l(p_1) < \dots < \Pi_l(p_n)$, alors l'ordre des Σ_{p_i} sur l est compatible.*

Preuve : *si on considère l'intersection de $\mathcal{V}_{\mathcal{P}}$, pour chaque paire de points (p, q) de \mathcal{P} tels que $\Pi_l(p) \neq \Pi_l(q)$, les cellules, sur l , Σ_p et Σ_q sont des sous-ensemble (peut-être vides) des cellules du diagramme de Voronoi qui ne considère que les points p et q . Ainsi, d'après le lemme 4.2, l'ordre sur l des projectés $\Pi_l(p)$ et $\Pi_l(q)$ est le même que celui des cellules Σ_p et Σ_q . Étant donné que cette propriété est vraie pour tout couple de points de \mathcal{P} , elle est vraie pour tous les points de \mathcal{P} . \square*

Sous l'hypothèse que $\Pi_l(u) < \Pi_l(v) < \Pi_l(w)$, nous pouvons construire un prédicat `CACHÉ_PAR_l` qui permet de décider si v est caché par u et w sur la droite l et ce quelle que soit la dimension.

Considérons les sites $\{p_i\}$ de \mathcal{P} tels que :

$$\Pi_l(p_i) < \Pi_l(p_j) \quad \forall (i, j) \ 1 \leq i < j \leq n \quad (4.13)$$

Ainsi, en utilisant le prédicat `CACHÉ_PAR_l` et l'algorithme 17 de suppression de sites, nous avons le corollaire suivant :

Corollaire 4.5 *Étant donné un ensemble de site \mathcal{P} vérifiant l'équation (4.13) et un prédicat `CACHÉ_PAR_l`, nous calculons l'ensemble des sites $\{q_i\}$ de \mathcal{Q} tels que $\Sigma_{q_i} \neq \emptyset$ en $O(|\mathcal{P}|)$ avec l'algorithme 17.*

4.2.6.2 Description de l'algorithme

Étant donné que toutes les observations de BREU ET AL. ont été montrées en dimension quelconque dans le paragraphe précédent, nous pouvons proposer un algorithme de calcul de diagramme de Voronoi discret et de transformée en distance euclidienne (voir algorithme 21).

Algorithme 21 Diagramme de Voronoi discret en nD

`DIAGRAMME_VORONOI_DISCRET_ND(I,P)`

- 1: Calcul des diagrammes de Voronoi discrets uni-dimensionnels comme initialisation
 - 2: **pour tout** dimension $k \geq 2$ **faire**
 - 3: **pour tout** tube T selon la dimension k **faire**
 - 4: Construction du prédicat `CACHÉ_PAR_T`
 - 5: $Candidats = \{\text{sites des diagrammes de Voronoi discrets dans les } (k - 1) \text{ dimensions le long du tube } T\}$
 - 6: Construction de la liste \mathcal{L}_T à partir de $Candidats$ en utilisant `CACHÉ_PAR_T` et l'algorithme 17
 - 7: **fin du pour**
 - 8: **fin du pour**
-

Théorème 4.1 *L'algorithme 21 est correct et sa complexité est linéaire en le nombre de points de l'image.*

Preuve : la preuve de la correction de cet algorithme est similaire à celle proposée par SAITO et TORIWAKI (1994) pour une approche basée sur des tubes le long des différentes dimensions. De plus, les preuves de la section 4.2.6.1 attestent la correction du processus de suppression de sites pour le cas nD à toute étape k de l'algorithme.

Concernant la preuve de linéarité, nous utilisons le corollaire 4.3 et le fait que l'algorithme 17 est linéaire en la taille de la liste d'entrée (le coût du prédicat CACHÉ_PAR_ T est toujours $O(1)$ quelle que soit la dimension). Ainsi, pour chaque dimension k de 1 à n , le calcul des diagrammes de Voronoi discrets partiels se font en temps linéaire (en le nombre de points de l'image).

Si les dimensions de l'image binaire sont données par $[1, m_1] \times \dots \times [1, m_n]$, l'algorithme précédent a un coût en $O(m_1 m_2 \dots m_n)$ ce qui est optimal pour le problème. \square

4.2.7 Analyse expérimentale

Dans un premier temps, nous illustrons la transformée en distance et le diagramme de Voronoi discret en dimension 2. Il est en effet plus simple d'observer les structures dans cette dimension qu'en dimension 3 directement.

Considérons tout d'abord une distribution aléatoire de sites (voir figure 4.10). Nous représentons donc le diagramme de Voronoi discret en affectant un niveau de gris pour chaque cellule. La figure 4.10-(c) présente la transformée en distance euclidienne exacte et pour des raisons de lisibilité, nous utiliserons par la suite une palette de niveaux de gris réduite pour représenter cette transformée (figure 4.10-(d)).

Dans la figure 4.11, nous calculons le diagramme de Voronoi ainsi que la transformée en distance d'un disque. Pour être plus précis, les sites considérés dans ce cas sont les pixels complémentaires de l'objet. Dans ce cas, nous pouvons voir très nettement le faisceau des cellules du diagramme de Voronoi qui *convergent* vers le centre du disque. La figure 4.12 considère une forme composée de deux disques joints.

La figure 4.13 illustre les algorithmes sur un rectangle plein. Dans l'illustration du diagramme de Voronoi, les phénomènes de dégradé sont dus à l'algorithme d'affectation d'un niveau de gris pour chaque cellule. Cependant, nous pouvons remarquer *le squelette* de la forme sur le diagramme que nous formaliserons par la suite.

Enfin, la figure 4.14 présente les différents algorithmes sur toute une variété d'objets synthétiques.

Pour illustrer les différents algorithmes en 3D, nous considérons tout d'abord l'image binaire de la figure 4.15-(a) qui se compose d'un cube discret, les sites sont placés à chaque sommet de celui-ci ainsi qu'à son centre. Les figures 4.15-(b) à 4.15-(d) représentent les différents diagrammes de Voronoi discrets partiels après respectivement la première phase d'initialisation, la fusion en y puis la fusion en z . Les figures 4.15-(e) à 4.15-(g) correspondent à des visualisations du diagramme de Voronoi 3D suivant différents plans de coupe.

Dans la figure 4.16, nous illustrons le diagramme de Voronoi discret 3D et la transformée en distance euclidienne d'une grille ne contenant que trois sites.

Dans le cadre d'analyse de formes 3D, la figure 4.17 présente un calcul de transformée en distance euclidienne sur un objet synthétique. Nous utilisons dans ce schéma un plan de coupe afin de voir ce qu'il se passe à *l'intérieur* de la forme. Pour cette image, nous pouvons observer les bons comportements de la transformée aussi

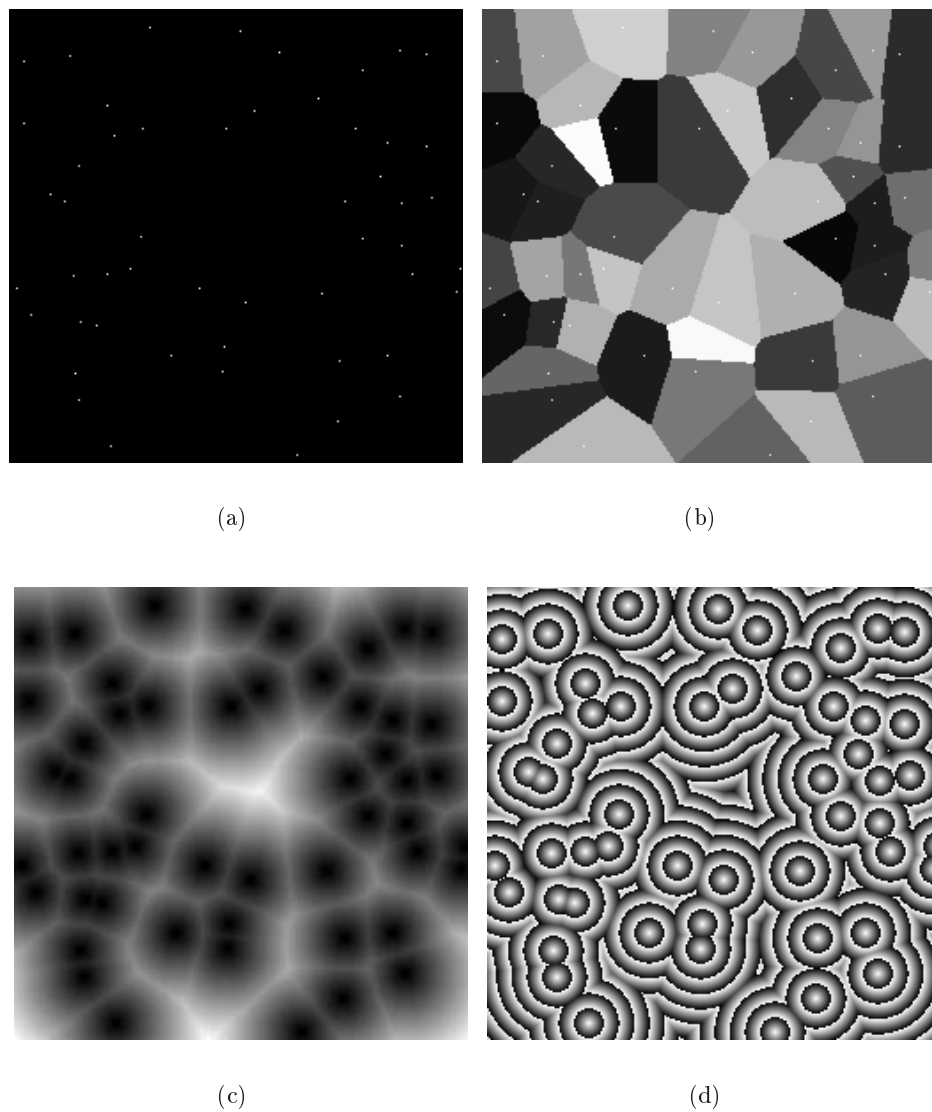


FIG. 4.10 —: *Diagramme de Voronoi et EDT de sites distribués aléatoirement : (a) image binaire des sites, (b) diagramme de Voronoi dans lequel un niveau de gris est affecté à chaque site, (c) Transformée en distance euclidienne (histogramme normalisé) et (d) EDT avec une palette de niveaux de gris réduite pour mieux visualiser les fronts d'ondes.*

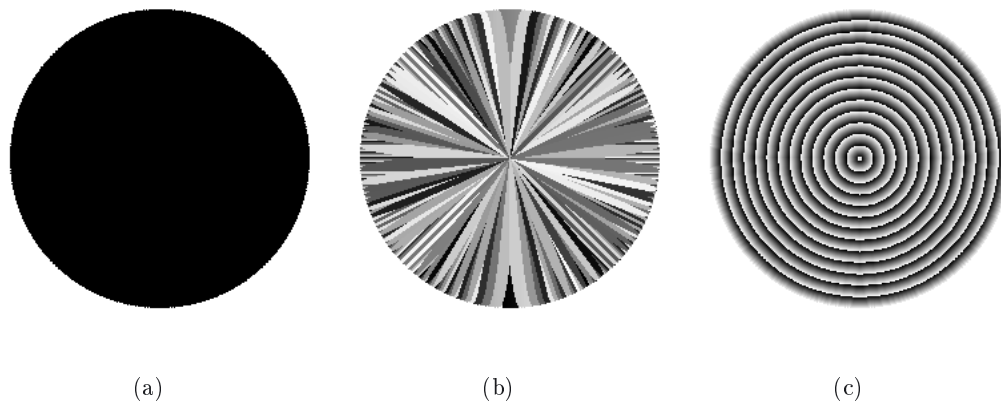


FIG. 4.11 –: *Diagramme de Voronoi et EDT d'un disque.*

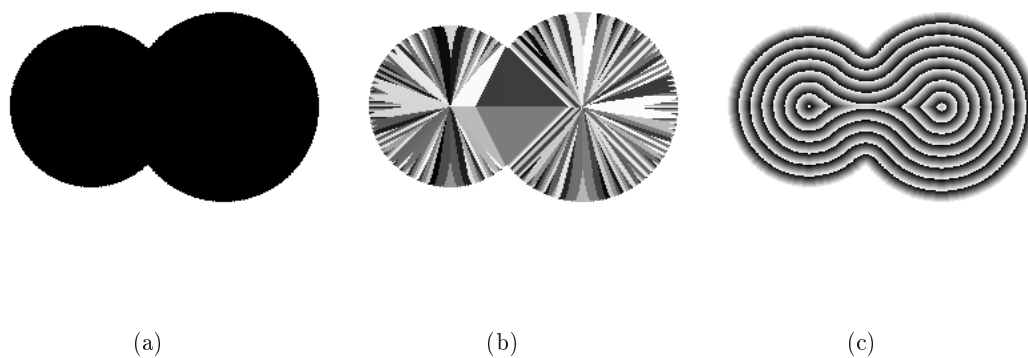


FIG. 4.12 –: *Diagramme de Voronoi et EDT de deux disques avec une intersection non vide : le disque de droite est légèrement plus grand que celui de gauche ce qui permet d'expliquer le non respect de la symétrie.*

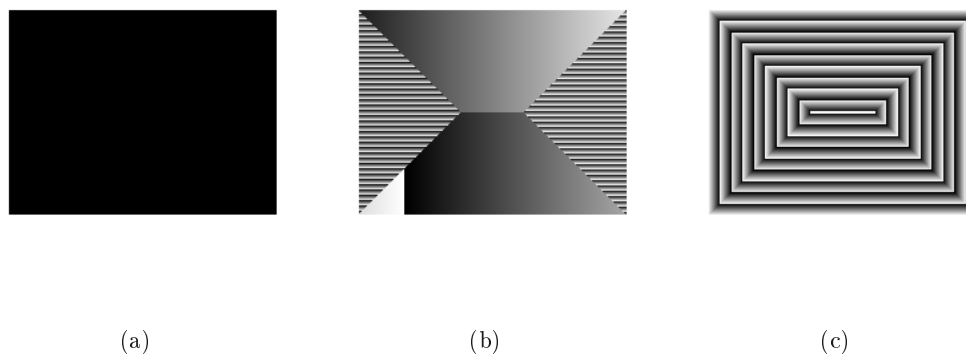


FIG. 4.13 –: *Diagramme de Voronoi et EDT d'un rectangle : (a) image source, (b) Diagramme de Voronoi et (c) EDT avec une palette de niveaux de gris réduite.*

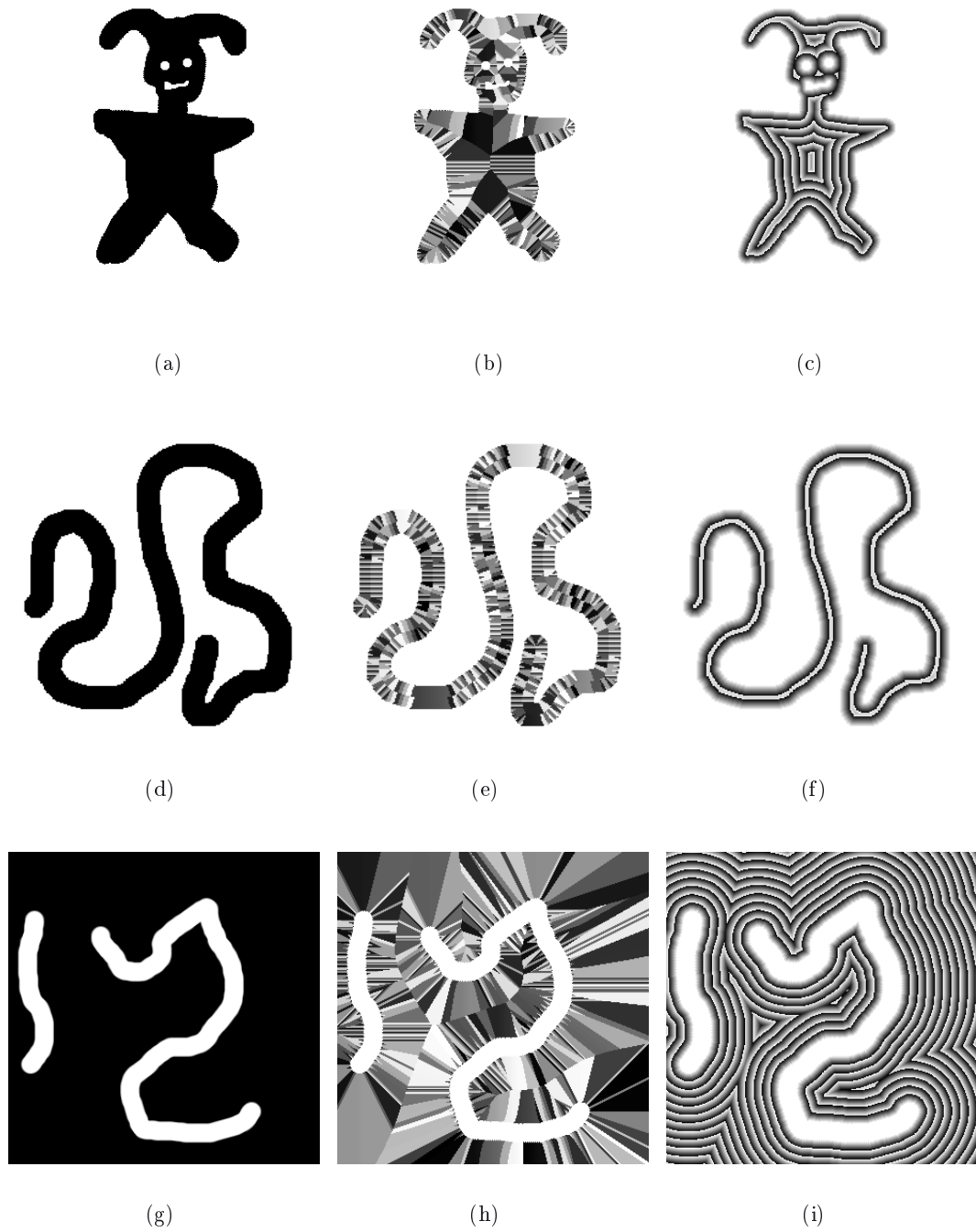


FIG. 4.14 —: *Diagramme de Voronoi et EDT d'objets synthétiques.*

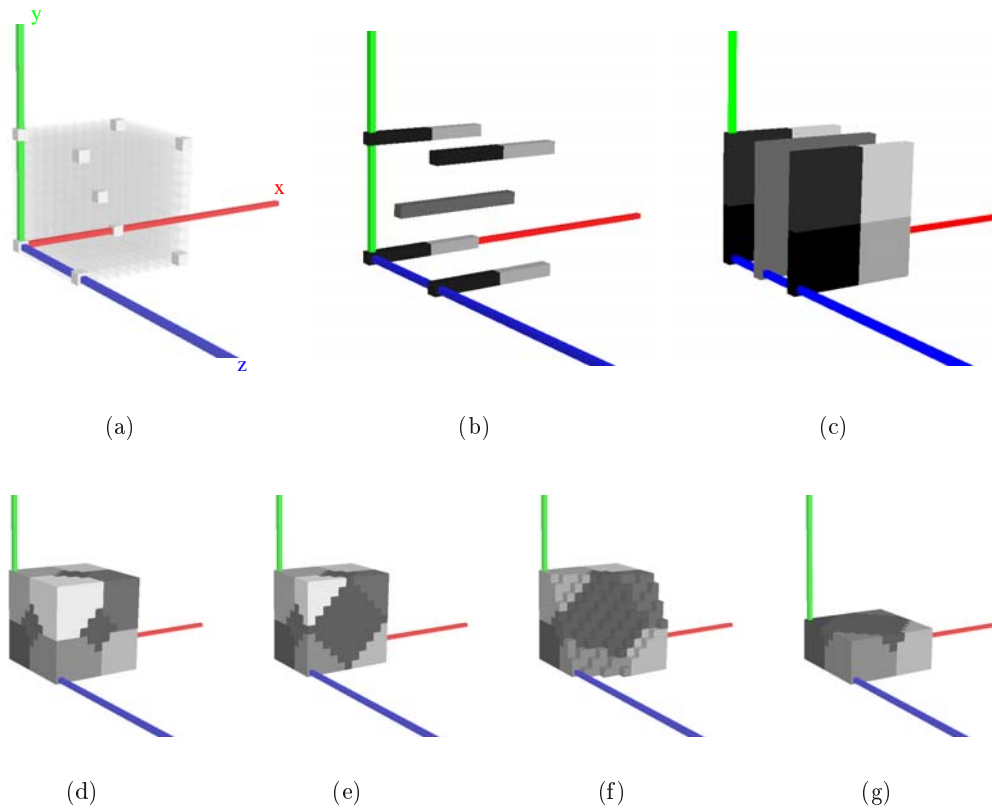


FIG. 4.15 –: *Illustration du diagramme de Voronoi discret 3D : (a) l'image binaire avec les sites en gris, (b) diagrammes de Voronoi discrets uni-dimensionnels après la première phase d'initialisation selon l'axe Ox , (c) diagrammes de Voronoi discrets 2D après la fusion en y , (d) diagramme de Voronoi discret 3D et (e) – (g) le diagramme 3D avec des plans de coupe pour voir les structures internes.*

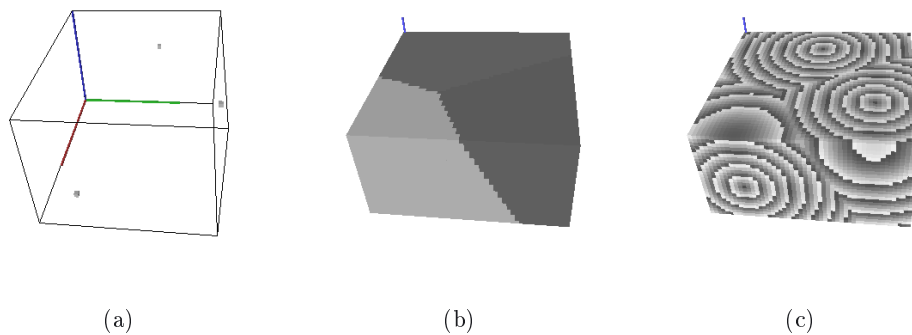
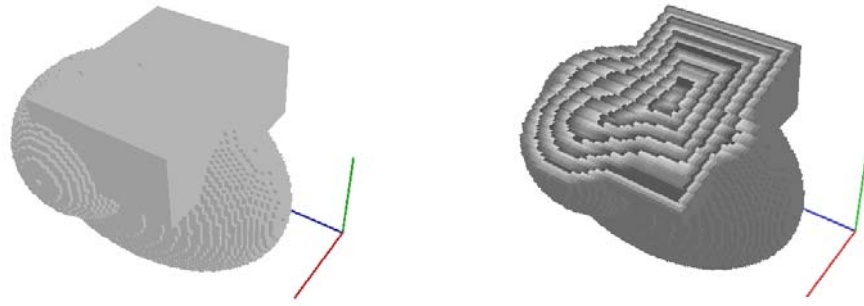


FIG. 4.16 –: *Illustration du diagramme de Voronoi discret 3D et de la transformée en distance : (a) image source avec trois sites, (b) diagramme de Voronoi discret 3D et (c) la transformée en distance (avec une palette réduite).*

bien sur les parties sphériques de la forme que sur la partie parallélépipédique.

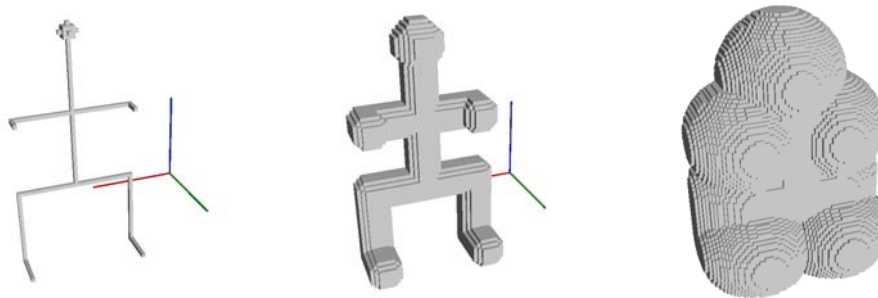


(a)

(b)

FIG. 4.17 –: *Transformée en distance euclidienne 3D : (a) image source et (b) transformée en distance avec une palette réduite et un plan de coupe.*

Finalement, nous pouvons aussi considérer la transformation inverse qui prendrait en entrée un squelette et qui reconstruirait la surface. La figure 4.18 illustre ce processus de reconstruction par surface implicite de l'objet à partir d'un squelette.



(a)

(b)

(c)

FIG. 4.18 –: *Reconstruction par surface implicite à partir d'un squelette : (a) le squelette pouvant être vu comme un ensemble de sites, (b) reconstruction d'une iso-surface à une distance 5 du squelette et (c) reconstruction à une distance 20.*

4.2.8 Diagramme de Voronoi discret et transformée en distance sur des grilles non-cubiques

Dans notre approche, nous introduisons une structure géométrique complexe, le diagramme de Voronoi, entre l'objet binaire et sa transformée en distance. Cette étape

supplémentaire nous permet d'étendre ces algorithmes pour des grilles non-cubiques.

Dans un premier temps, nous considérons des grilles parallépipédiques. En pratique, beaucoup d'appareils d'acquisition classiques en imagerie médicale produisent des grilles qui ne sont pas isotropiques. Par exemple, pour certains types d'appareils et pour certains types de partie du corps, les rapports entre les directions peuvent aller jusqu'à 3 ou 5.

De nombreux travaux existent pour résoudre le problème de la transformée en distance sur ce type de grille. Dans le cas des distances de chanfrein, SINTORN et BORGEFORS (2002) proposent, par exemple, un calcul des poids du masque adapté en fonction du facteur d'échelle entre les axes. Dans ce cas, si ce facteur d'échelle varie entre deux images volumiques, il faut recalculer les poids optimaux. Pour la cas de la transformée en distance, l'algorithme de SAITO et TORIWAKI (1994), optimisé par HIRATA (1996) et MEIJSTER ET AL. (2000), permet de calculer la transformée exacte et résout ainsi le problème. Nous montrons, par la suite, comment notre approche permet elle aussi de résoudre ce problème avant de passer à des espaces discrets plus complexes (grille hexagonale par exemple).

Soit \mathcal{P} un ensemble de sites tel que $\mathcal{P} = \{(x, y, z) \in \mathbb{Z}^3\}$. Nous notons $\mathcal{P}_{\alpha\beta}$ l'ensemble de sites $(x, \alpha y, \beta z)$ avec $(x, y, z) \in \mathbb{Z}^3$ et $\alpha, \beta \in \mathbb{R}^2$

Proposition 4.2 *Soit $\mathcal{V}_{\mathcal{P}}$ le diagramme de Voronoi associé aux sites \mathcal{P} et $\mathcal{V}_{\mathcal{P}_{\alpha\beta}}$ le diagramme de Voronoi associé aux sites $\mathcal{P}_{\alpha\beta}$, alors :*

$$\mathcal{V}_{\mathcal{P}_{\alpha\beta}} = \mathcal{E}_{\alpha\beta}(\mathcal{V}(\mathcal{P}))$$

où $\mathcal{E}_{\alpha\beta}$ correspond à un changement d'échelle de α suivant y et un changement d'échelle de β suivant z .

Preuve : *La preuve est directe, on considère des points p, q , et r de \mathcal{P} et p', q' et r' leurs analogues³ dans $\mathcal{P}_{\alpha\beta}$, alors nous avons :*

$$\text{dist}(p, q) < \text{dist}(p, r) \Leftrightarrow \text{dist}(p', q') < \text{dist}(p', r')$$

où *dist* est la distance euclidienne.

Ainsi, si un point p appartient à la cellule du site q dans $\mathcal{V}_{\mathcal{P}}$, alors il appartient à la cellule du site q' dans $\mathcal{V}_{\mathcal{P}_{\alpha\beta}}$, et réciproquement. \square

Finalement, nous pouvons utiliser exactement les mêmes algorithmes pour extraire le diagramme de Voronoi discret de la grille \mathbb{Z}^3 . Il suffit de les transformer avec les changements d'échelle $\mathbb{Z} \times \alpha\mathbb{Z} \times \beta\mathbb{Z}$ et ainsi obtenir, après réécriture, la transformée en distance euclidienne sur ces grilles (voir figure 4.19). Bien évidemment, ce que nous avons illustré en 3D est aussi valable quelle que soit la dimension et donc les algorithmes proposés s'adaptent aussi pour des grilles anisotropiques en dimension n .

Si nous considérons maintenant des grilles non-cubiques, par exemple des grilles hexagonales, très peu de solutions existent pour le calcul de la transformée en distance. Cependant, dans le cas particulier des grilles hexagonales, de nombreux objets à analyser existent, notamment en physique.

Tout d'abord, nous pouvons remarquer qu'à aucun endroit de la méthode ni de son analyse théorique le fait que les tubes soient parallèles aux axes de la grille n'est

3. un point (x, y, z) de \mathcal{P} a pour analogue dans $\mathcal{P}_{\alpha\beta}$ le point $(x, y/\alpha, z/\beta)$.

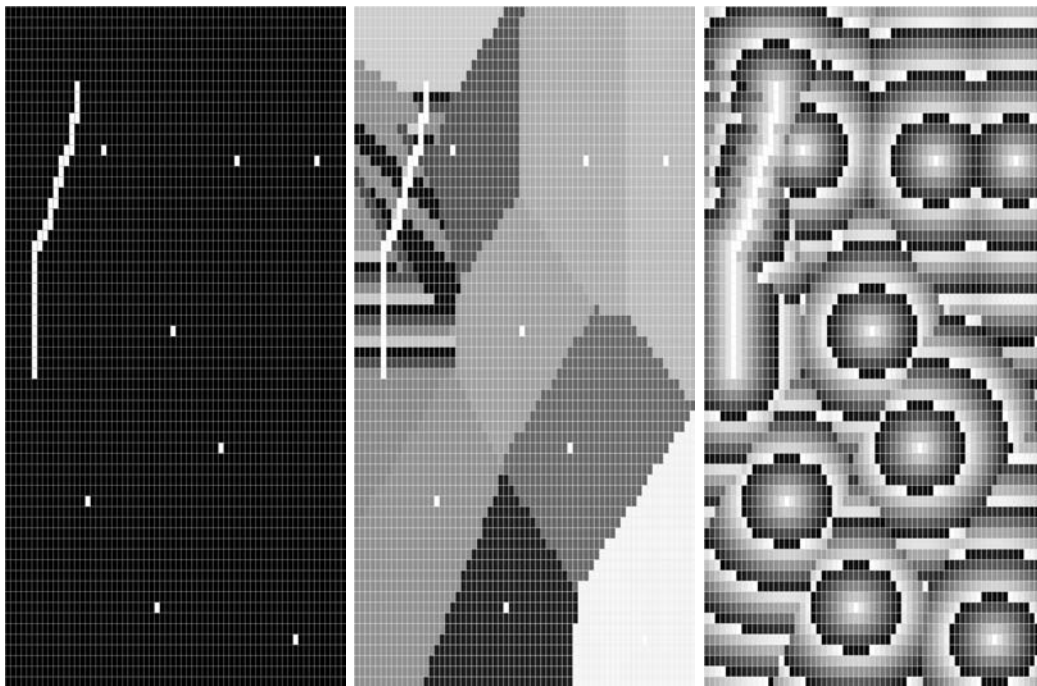


FIG. 4.19 –: *Diagramme de Voronoi et EDT sur une grille anisotropique : le facteur d'échelle entre l'axe Ox et l'axe Oy est de 2.*

utilisé. En d'autres termes, nous pouvons très bien envisager des suppressions de sites, dans des tubes orthogonaux, qui ne soient pas forcément parallèles aux axes. En effet, nous pouvons effectuer le processus de fusion avec un tube donné par une équation du type $y = \alpha x + \beta$, il suffit dans ce cas d'adapter le prédicat `CACHÉ_PAR` pour tenir compte de l'équation. Considérons pour cela une droite réelle notée l et telle que $y = \alpha x + \beta$, ainsi que deux points u et v tels que $\Pi_l(u) \neq \Pi_l(v)$ ce qui s'écrit dans notre exemple $2(u_x - v_x) + 2\alpha(v_y - u_y) \neq 0$. Dans ce cas nous avons les coordonnées du point $\hat{u}v$, intersection de la médiatrice de $[uv]$ et de l , qui sont données par :

$$\hat{u}v = \left(\frac{u_x^2 + u_y^2 - v_x^2 - v_y^2 + 2\beta(v_y - u_y)}{2(u_x - v_x) + 2\alpha(v_y - u_y)}, \frac{\alpha(v_x^2 + v_y^2 - u_x^2 - u_y^2) + 2\beta(u_x - v_x)}{2(u_x - v_x) + 2\alpha(v_y - u_y)} \right)$$

Ainsi, il est facile de construire un prédicat `CACHÉ_PAR` (α, β) qui permet de tester, étant donnés trois sites u , v et w tels que $\Pi_l(u)_y < \Pi_l(v)_y < \Pi_l(w)_y$ (on suppose $\alpha \neq 0$), si la cellule associée à v est *cachée* par les cellules de u et de w sur la droite l . En ne changeant que ce simple prédicat et la construction des tubes (qui peut être basée sur les notions de droites discrètes présentées dans le chapitre 2.2), nous réutilisons tous les algorithmes présentés pour le diagramme de Voronoi discret et la transformée en distance euclidienne.

Bien évidemment, utiliser des tubes qui ne sont pas parallèles aux axes n'a pas beaucoup d'intérêt dans le cas de grilles orthogonales. Cependant, dans le cas de grilles hexagonales, cette propriété est fondamentale. Considérons l'échantillonnage de la grille proposé dans la figure 4.20 qui consiste à ajouter aux points discrets du maillage certains points qui sont sur les arêtes *en bas à gauche*. Ainsi, en utilisant une phase d'initialisation dans les tubes paramétrés par $y = \sqrt{3}x + b$ puis en appliquant

le processus de suppression de sites avec le prédicat $\text{CACHÉ_PAR_}(-1/\sqrt{3}, b')$ dans les tubes orthogonaux aux précédents, nous obtenons un algorithme de calcul du diagramme de Voronoi discret sur notre maillage. Étant donné que l'échantillonnage est linéaire en le nombre de points de la grille hexagonale d'origine, la complexité du calcul reste linéaire en le nombre de points et donc toujours optimal pour le problème. La figure 4.21 illustre ce calcul.

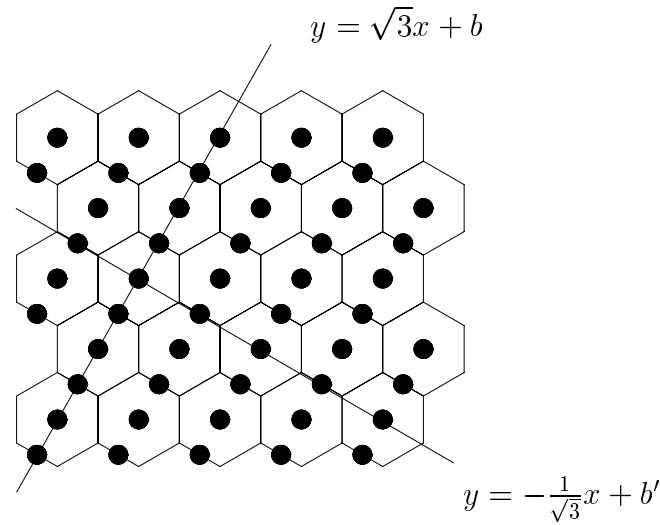


FIG. 4.20 – Rééchantillonnage de la grille hexagonale pour le calcul du diagramme de Voronoi discret

Si maintenant nous revenons à notre grille initiale d'hexagones, nous construisons le diagramme de Voronoi discret pour cette grille en ne conservant du diagramme que les points qui sont au centre des hexagones. Nous obtenons ainsi le diagramme de Voronoi discret et la transformée en distance euclidienne en temps optimal pour ce type de grille.

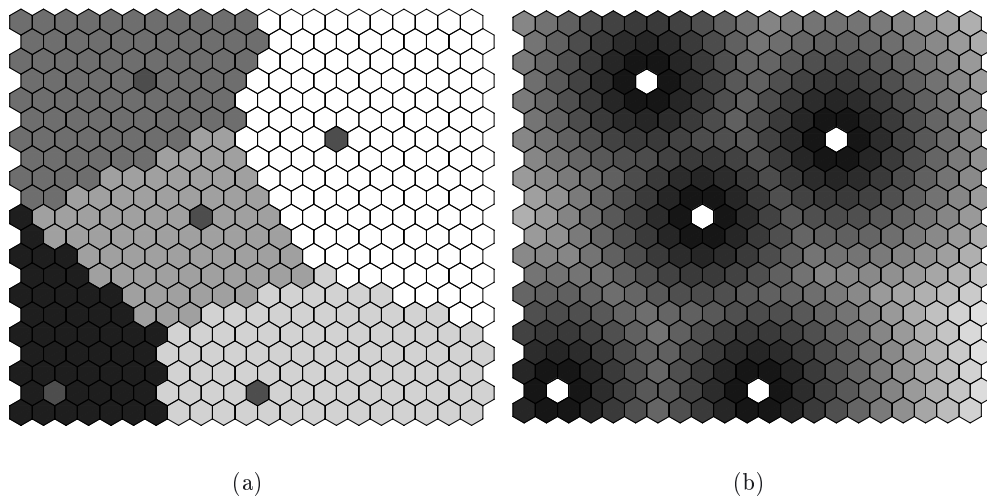


FIG. 4.21 – Diagramme de Voronoi et EDT sur une grille hexagonale : (a) diagramme de Voronoi discret (les sites sont les cellules noires) et (b) transformée en distance euclidienne.

4.2.9 Conclusion et perspectives

Nous avons montré dans cette partie un algorithme calculant la transformée en distance euclidienne exacte sur des images de dimension n en temps optimal, c'est-à-dire en temps linéaire en le nombre de points discrets de la grille. Cet algorithme se base sur la construction d'une étape intermédiaire qui est diagramme de Voronoi discret. Nous avons montré l'intérêt de cette construction dans le cas des grilles anisotropiques et des grille hexagonales.

4.3 Squelette de forme et axe médian

Maintenant que nous avons des algorithmes optimaux pour le calcul de la transformée en distance, nous allons extraire de celle-ci un objet géométrique appelé *squelette* ou *axe médian* qui permet une description géométrique et topologique de la forme. Nous présentons donc les approches classiques pour construire de tels objets en géométrie discrète puis nous proposons un algorithme direct et optimal en temps basé sur le diagramme de Voronoi discret que nous venons de définir.

4.3.1 Axe médian et squelette en géométrie discrète

Comme nous l'avons déjà présenté dans le paragraphe 4.2, la notion d'axe médian proposée par BLUM (1967) correspond à une généralisation des axes de symétrie d'une forme. ROSENFELD et PFALTZ (1966) parlent de squelette de forme qui est défini comme étant le sous-ensemble minimal de la forme permettant la reconstruction de celle-ci en appliquant une transformée en distance inverse.

En géométrie discrète, nous trouvons généralement le terme *squelette* pour décrire un objet topologique homotope à la forme (nous reviendrons sur ce point plus tard). Dans ce qui suit et quand nous ne précisons pas, nous parlerons toujours de squelette géométrique.

Nous pouvons définir formellement cette notion de *minimalité* en introduisant la notion de *boule maximale* pour une métrique donnée.

Définition 4.4 (Boule maximale) *Une boule, au sens de la métrique considérée, de centre p et de rayon r est dite maximale si celle-ci est entièrement contenue dans la forme et qu'il n'existe aucune autre boule de la forme qui la contienne.*

Ainsi, l'axe médian se définit de la façon suivante :

Définition 4.5 (Axe médian) *L'axe médian d'une forme est l'ensemble des centres des boules maximales de celle-ci.*

Selon ces définitions, on peut tout de suite remarquer que le test qui permet de décider si un point p est centre de boule maximale est primordial. Pour un point p , le rayon de la boule susceptible d'être maximale est bien évidemment donné par la transformée en distance selon la métrique considérée. La figure 4.22 illustre cette notion de maximum local dans la carte de distance : la transformée en distance est représentée sous forme de surface dont la hauteur est la valeur de la transformée. Sur ce type de schéma, une définition intuitive de l'axe médian correspond aux *crêtes* des ces surfaces.

Pour calculer cet axe médian, plusieurs approches sont envisageables. Nous présenterons uniquement une approche récente de REMY (2001). Étant donné un point

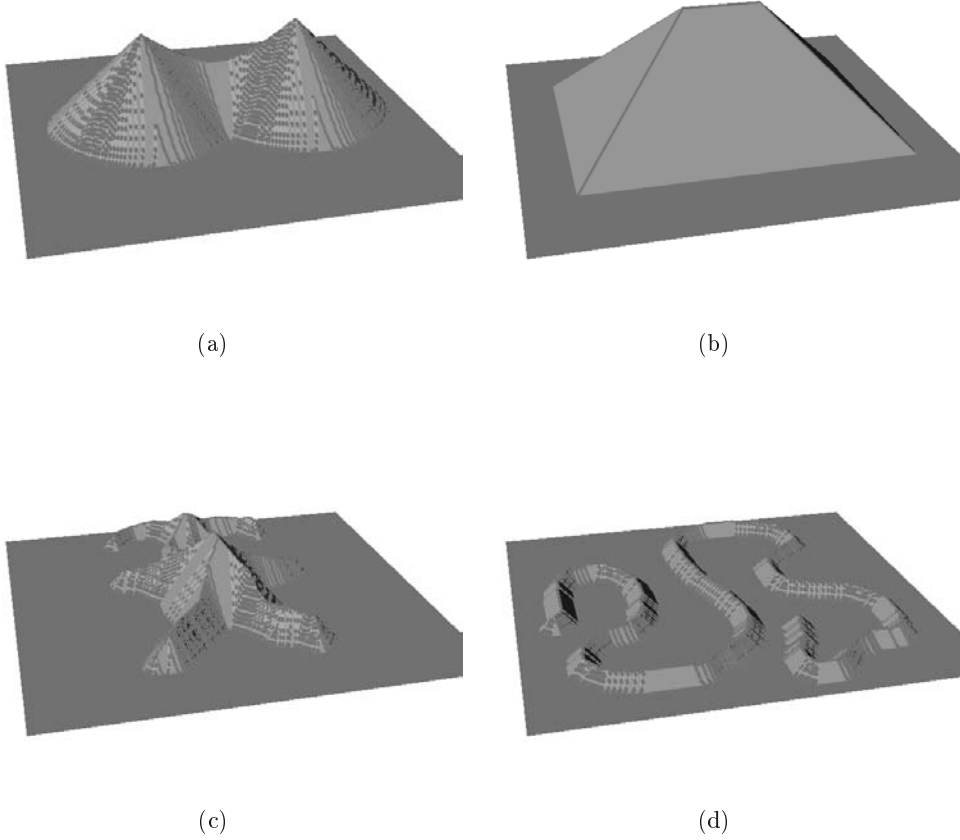


FIG. 4.22 – : Représentation surfacique de la transformée en distance : l'axe médian correspond aux crêtes de ces surfaces.

de la transformée en distance, décider si celui-ci est centre de boule maximale consiste à regarder dans un certain voisinage s'il n'existe pas de centres de boules plus grosses contenant la première. De manière plus formelle, on considère un point p de la transformée à une distance d du bord et un point q à une distance d' du bord. L'idée est la suivante : quelles sont les conditions sur d' pour que la boule $B(q, d')$ contienne la boule $B(p, d)$? Pour résoudre ce problème on peut pré-calculer les valeurs de d' pour, étant donné le déplacement \vec{pq} et la valeur d , décider si p fait parti de l'axe médian. Ainsi, le test d'appartenance à l'axe médian s'effectue après une recherche dans une table de correspondance pré-calculée (Look-Up Table ou LUT dans la bibliographie). REMY (2001) propose donc une méthodologie générale pour le calcul de cette table de correspondance pour des métriques de chanfrein et le carré de la distance euclidienne.

Il existe d'autres types de squelettes d'objets discrets, basé par exemple sur des propriétés uniquement topologiques. Ainsi, le squelette *topologique* d'une forme peut se définir comme étant le sous-ensemble minimale de points discrets de l'objet ayant la même homotopie que l'objet global (même nombre de trous, même nombre de composantes connexes...). Ce genre de squelette se construit par exemple avec des suppressions séquentielles ou parallèles de points de la forme vérifiant un certain nombre de critères locaux (notion de *point simple*) mais garantissant une propriété globale d'homotopie (voir par exemple LOHOU et BERTRAND 2002).

Nous ne nous intéresserons par la suite qu'aux squelettes *géométriques*

4.3.2 Squelette n -dimensionnel basé sur le diagramme de Voronoi discret

Considérons dans un premier temps une forme polygonale. Dans ce cas, le squelette et le diagramme de Voronoi généralisé intérieur de la forme coïncident (voir figure 4.23). En effet, les sommets de ce diagramme ainsi que les arêtes sont centres de boules maximales pour la forme. Une des premières approches pour construire ce squelette a été proposé par MONTANARI (1969).

Dans le cas de formes quelconques, DOMINIQUE ATTALI (1995) propose un calcul de squelette en considérant le diagramme de Voronoi interne d'un échantillonnage du bord de la forme. Ce squelette approxime le squelette exact de la forme continue sous-jacente mais a des propriétés de convergence si l'on considère un échantillonnage tendant vers l'infini.

De manière plus précise, étant donné un échantillonnage et le diagramme de Voronoi de celui-ci, le squelette peut être extrait du diagramme selon plusieurs approches (voir ATTALI 1995) :

- ensemble des sommets du diagramme inclus dans la forme
- ensemble des éléments du diagramme (sommets et arêtes) inclus dans la forme
- intersection entre le diagramme et la forme
- dual de la forme

Dans les algorithmes présentés précédemment, non seulement nous avons une transformée en distance sans erreur en temps linéaire en le nombre de points de la forme, mais nous avons surtout dans le même temps le diagramme de Voronoi discret de l'ensemble des points du complémentaire. Il est donc tout à fait normal de construire un squelette directement sur ce diagramme plutôt que sur la transformée

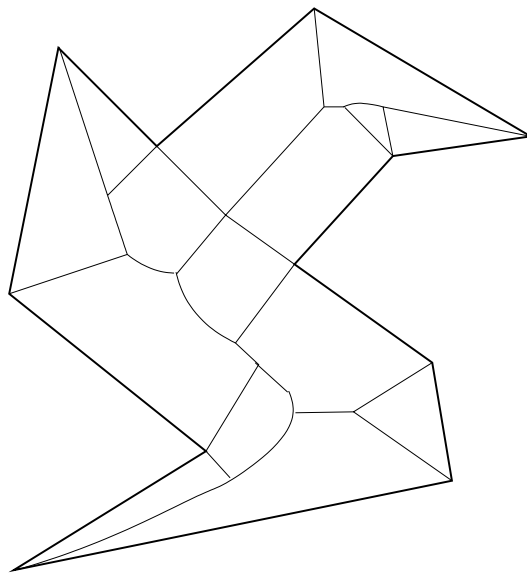


FIG. 4.23 – : *Squelette d'un polygone basé sur le diagramme de Voronoi généralisé (extrait de ATTALI 1995)*

en distance. L'information primordiale pour une extraction de squelette que n'ont pas les autres algorithmes de transformée en distance, c'est le lien entre tout point de la forme et le point du complémentaire le plus proche.

Nous définissons donc notre squelette de la manière suivante :

Définition 4.6 (Squelette basé sur diagramme de Voronoi discret) *Étant donnée une forme discrète et le diagramme de Voronoi discret de son complémentaire, le squelette associé à cette forme est l'ensemble des points discrets sommets ou sur les arêtes du diagramme de Voronoi discret.*

Comme le souligne ATTALI, un squelette défini de la sorte ne peut pas être utilisé directement, il faut le simplifier. Une des difficultés introduites par le diagramme discret est que nous ne connaissons pas la structure globale du graphe du diagramme de Voronoi, nous savons uniquement pour tout point, la cellule à laquelle il appartient. Ainsi, nous ne pouvons utiliser que des critères locaux pour la simplification du squelette. Nous considérons donc les filtrages locaux du squelette proposés par ATTALI. Pour cela, nous reprenons les définitions suivantes (voir figure 4.24) :

Définition 4.7 (Fonction d'étanchéité (ATTALI, 1995)) *Soit X une forme continue et $Sk(X)$ son squelette. La fonction d'étanchéité $\rho(\cdot)$ est la fonction qui à chaque point s du squelette associe le rayon de la boule maximale centrée en s .*

Cette fonction correspond à la donnée de la transformée en distance en tout point du squelette.

Définition 4.8 (Angle bissecteur (ATTALI, 1995)) *Soit X une forme continue et $Sk(X)$ son squelette. Soit s un point du squelette. La boule maximale centrée en s possède deux points de contact p_0 et p_1 avec la frontière de X . On appelle angle bissecteur l'angle $\widehat{p_0sp_1}$ compris entre 0 et π . Par la suite, on note $\alpha(s)$ l'angle*

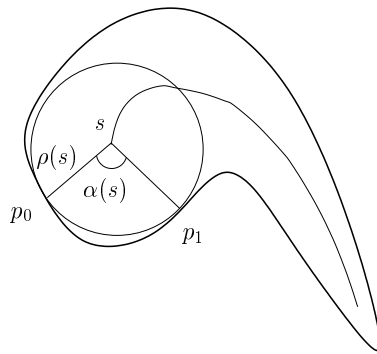


FIG. 4.24 –: *Illustration de la fonction d'étanchéité et l'angle bissecteur de ATTALI (1995).*

bissecteur au point s . L'angle bissecteur aux points terminaux et multiples se calcule par passage à la limite.

De manière informelle, on ne voudrait garder comme points du squelette que des points qui sont centres de boules maximales *assez grosses* et qui ne sont pas des extrémités *trop vives* du squelette. Nous filtrons donc les points du squelette construits précédemment avec deux seuils α_0 et ρ_0 et nous gardons les points dont l'angle bissecteur $\alpha(s)$ et l'étanchéité $\rho(s)$ vérifient :

$$\alpha(s) > \alpha_0 \quad \text{et} \quad \rho(s) > \rho_0$$

Nous appliquons cet algorithme d'extraction de squelette sur des formes géométriques les plus diverses. La figure 4.25, représente le calcul du squelette d'un disque. Pour illustrer le processus de simplification du squelette, nous représentons tous les points de celui-ci sur un graphe dans lequel en abscisse se trouve la valeur de l'angle bissecteur et en ordonnée la valeur de la fonction d'étanchéité. Ainsi, un filtrage correspond à extraire les points de ce graphe contenus dans une certaine fenêtre (figure 4.25-(c)). Pour le cas du disque, on peut observer que le centre de celui-ci est très isolé des autres points sur ce graphe ce qui permet de l'extraire facilement pour construire le squelette.

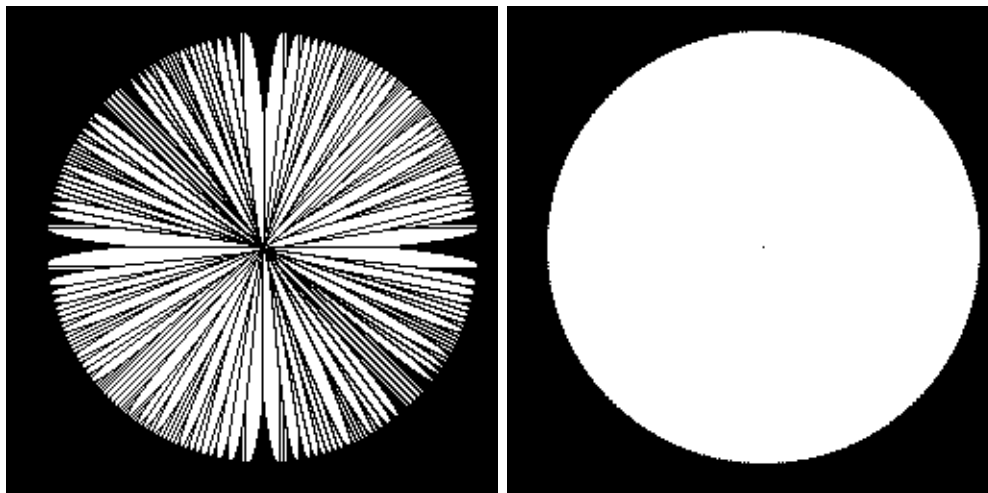
La figure 4.26 illustre le calcul du squelette sur une forme composée de deux disques d'intersection non vide. On peut noter que ces disques n'ont pas exactement le même rayon et les centres ne sont pas exactement sur l'horizontale. Dans la figure 4.26-(c), on notera que le squelette *attendu* se localise assez bien sur le graphe des paramètres. Les formes *hyperboliques* que l'on peut observer dans ces graphes s'expliquent par le fait que si on cherche l'ensemble des points d'un squelette généré par deux points du complémentaire distants de d , nous obtenons la fonctionnelle (ATTALI, 1995) :

$$\rho = \frac{d}{2 \sin \frac{\alpha}{2}}$$

Finalement, les figures 4.27 et 4.28 illustrent le calcul du squelette sur un carré et une forme *humanoïde*.

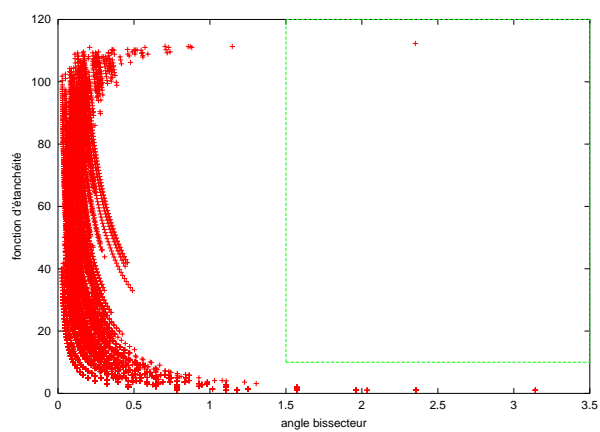
4.3.3 Conclusion et discussion

Nous avons présenté une définition de squelette discret basée sur la notion de diagramme de Voronoi discret. Nous obtenons donc un algorithme d'extraction qui ne



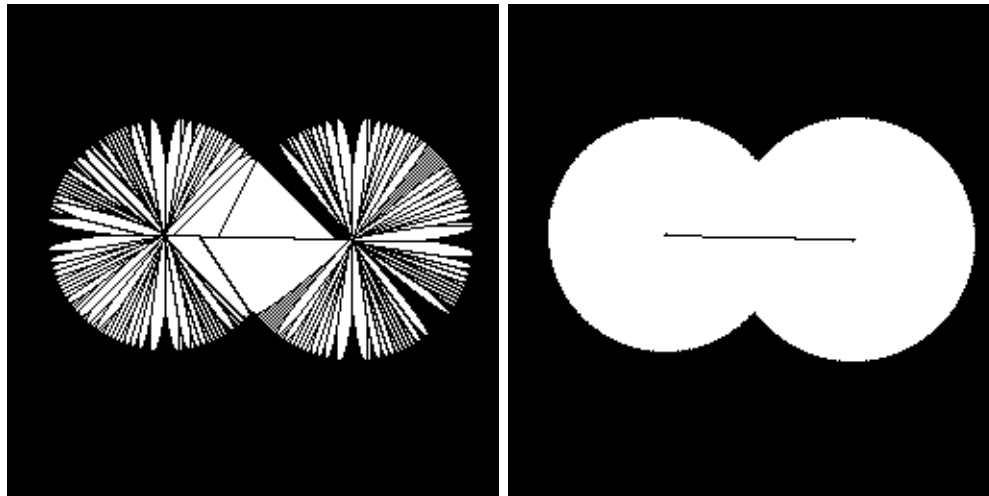
(a)

(b)



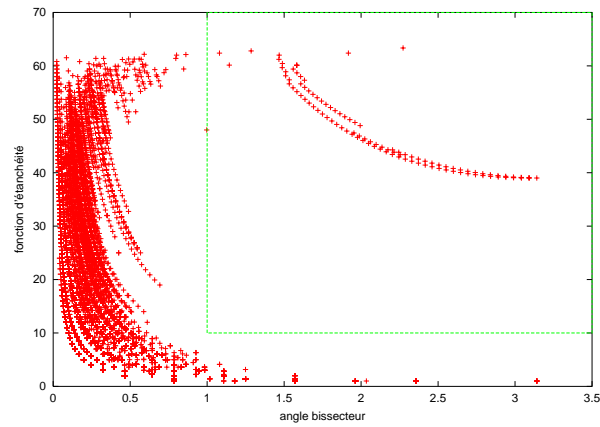
(c)

FIG. 4.25 –: *Squelette d'un disque : (a) points du squelette sans simplification, (b) squelette après simplification avec $\alpha_0 = 1.5$ et $\rho_0 = 10$, et (c) graphe des paramètres $\alpha(s)$ et $\rho(s)$ avec les seuils.*



(a)

(b)



(c)

FIG. 4.26 —: Squelette de deux disques joints : (a) points du squelette sans simplification, (b) squelette après simplification avec $\alpha_0 = 1$ et $\rho_0 = 10$, et (c) graphe des paramètres $\alpha(s)$ et $\rho(s)$ avec les seuils.

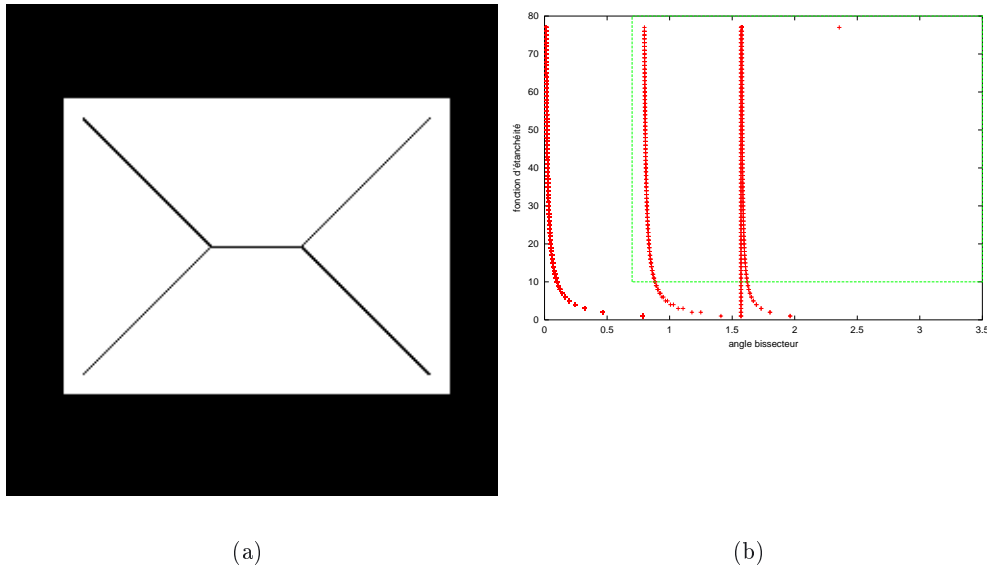


FIG. 4.27 – *Squelette d'un rectangle : (a) squelette après simplification avec $\alpha_0 = 0.7$ et $\rho_0 = 10$, et (b) graphe des paramètres $\alpha(s)$ et $\rho(s)$ avec les seuils.*

nécessite pas de structure de type table de correspondance (*Look-up table*) et qui construit le squelette en temps optimal (linéaire en le nombre de points de l'image). Nous avons aussi utilisé quelques techniques de simplification de squelette sur des critères locaux de ATTALI (1995). Ce travail nous a permis de vérifier la faisabilité pratique de cette technique de construction de squelette.

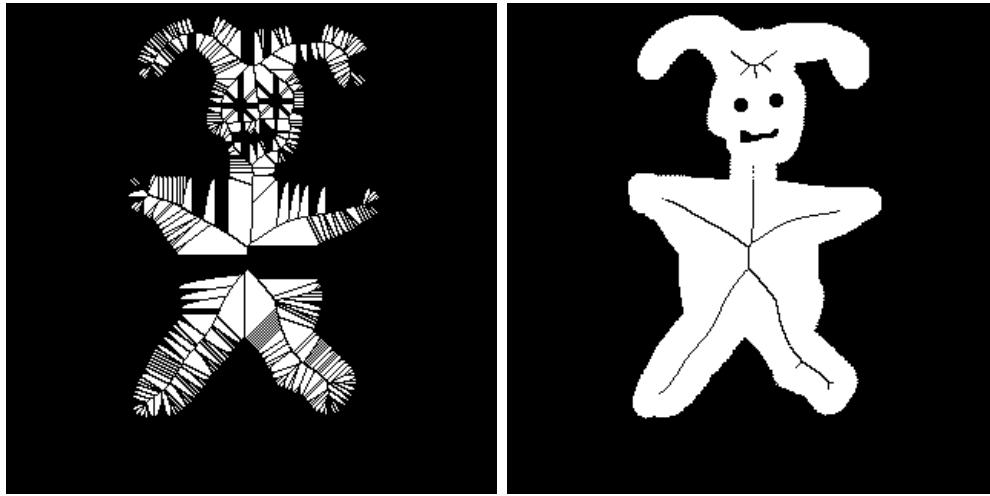
De plus, ce squelette peut s'extraire de la même manière quelle que soit la dimension de l'image et pour des formes de grilles variées. En effet, la construction du diagramme de Voronoi discret, comme nous l'avons vu précédemment, se fait en temps optimal. Les techniques de simplification de squelette présentées s'adapte aussi en dimension supérieure (voir ATTALI 1995 pour le cas 3D).

Nous avons donc un algorithme optimal en temps pour toute dimension qui permet d'extraire un squelette d'une forme discrète.

Dans le schéma proposé, nous n'avons appliqué que des simplifications locales du squelette sans aucune préservation de quelconque propriété topologique. Ce travail de simplification globale apparaît donc comme une perspective importante de cette approche.

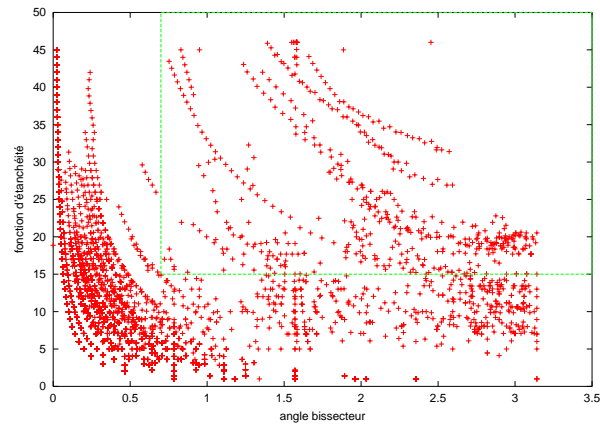
Au niveau des simplifications locales du squelette, de nombreuses autres approches peuvent être envisagées. On peut, par exemple, proposer une simplification par *hysteresis* se référant à des techniques classiques de détection de contour en analyse d'images. Ainsi, un tel processus de simplification manipulerait deux paires de seuils (α_0, ρ_0) et (α_1, ρ_1) et aurait le fonctionnement suivant :

1. si $\alpha(s) > \alpha_1$ et $\rho(s) > \rho_1$, le point du squelette est conservé
2. si $\alpha(s) < \alpha_0$ et $\rho(s) < \rho_0$, le point du squelette est supprimé
3. dans le cas intermédiaire, on regarde s'il existe un chemin d'une certaine taille joignant le point p et un point satisfaisant la première règle. On peut même *diriger* la recherche de ce chemin en regardant dans la direction de la médiatrice



(a)

(b)



(c)

FIG. 4.28 – : Squelette d'une forme synthétique : (a) points du squelette sans simplification, (b) squelette après simplification avec $\alpha_0 = 0.7$ et $\rho_0 = 15$, et (c) graphe des paramètres $\alpha(s)$ et $\rho(s)$ avec les seuils.

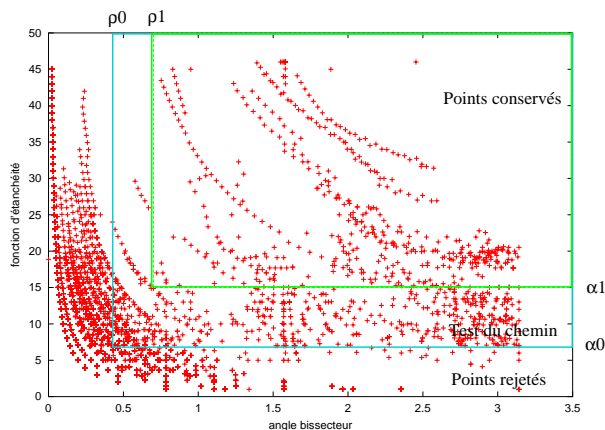


FIG. 4.29 – Illustration sur le graphe des paramètres de la simplification du squelette par hysteresis.

du segment $[p_0p_1]$ de la définition 4.8 (voir figure 4.29 pour une illustration dans le graphe des paramètres du nouveau filtre).

Sur le plan méthodologique, une perspective intéressante serait la comparaison de ce type de squelette avec les axes médians calculés par table de correspondance. Au delà de cette comparaison, ATTALI ET AL. (1997) ont illustré une extraction de squelette utilisant à la fois le diagramme de Voronoi continu et l'axe médian associés à la forme. Une tel calcul hybride de squelette se basant sur notre diagramme de Voronoi discret nous semble être une perspective importante.

4.4 Visibilité et géodésiques discrètes

Dans cette partie, nous nous intéressons aux notions de visibilité en géométrie discrète et de métrique géodésique. L'intérêt de définir de tels outils est double : d'une part nous augmentons le modèle discret d'un nouvel outil d'analyse, mais nous répondons aussi à un besoin pratique dans de nombreuses applications (par exemple l'angioscopie virtuelle imagerie médicale).

En géométrie algorithmique classique, la notion de visibilité se définit de la manière suivante : étant donné un domaine avec des obstacles polygonaux, un point est *visible* d'un autre si le segment entre ces deux points ne rencontre aucun obstacle. En se basant sur cette définition, nous pouvons définir la notion de chemin géodésique. Considérons deux points et le graphe de visibilité dont les sommets sont les sommets des obstacles polygonaux et les deux points considérés. Les arêtes de ce graphe correspondent à la relation binaire de visibilité, et sont pondérées par la distance euclidienne. Ainsi, le chemin géodésique (ou plus court chemin) entre les deux points correspond au plus court chemin dans le graphe précédent (DE BERG ET AL., 2000; CORMEN ET AL., 1990). La figure 4.30 illustre ces différentes définitions.

En géométrie discrète, nous considérons un domaine dont les obstacles sont des ensembles de pixels connexes. Une première définition de visibilité discrète a été proposée par SOILLE (1991, 1994, 1999). Celle-ci se base sur le tracé de droite de BRESENHAM (1965) : le test de visibilité consiste à tracer le segment de BRESENHAM entre les deux pixels et à vérifier qu'aucun obstacle n'est rencontré. En se basant sur cette définition, SOILLE propose aussi un calcul de chemin géodésique. Nous reviendrons sur cette approche ultérieurement.

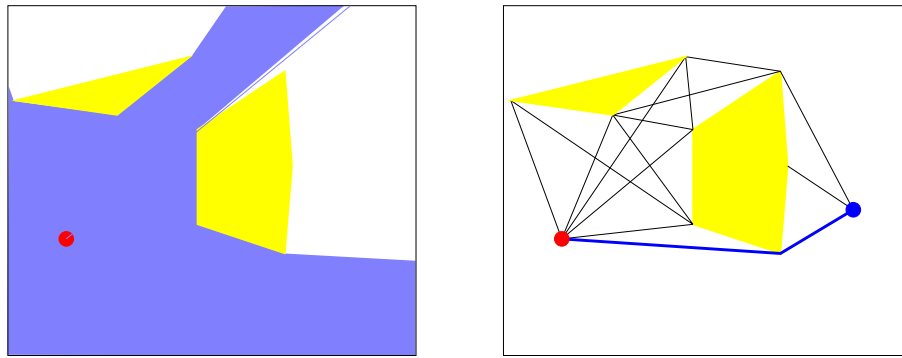


FIG. 4.30 –: *Illustration des définitions : à gauche, tous les points dans la zone bleue sont visibles du point rouge. À droite, construction du graphe de visibilité et du plus court chemin (en gras) entre le point rouge et le point bleu.*

Par la suite, LENOIR (1999) propose une approche basée sur des notions de combinatoire et de graphe. Dans le cas 2D, le chemin géodésique entre deux points est défini de la manière suivante : on commence par dénombrer tous les chemins 4-connexes joignant les deux points considérés. Ensuite, à chaque pixel visité par un chemin 4-connexe, nous affectons une valeur numérique qui correspond au nombre de chemins qui passent par ce point. Finalement, le plus court chemin correspond au chemin qui maximise la somme de ces valeurs. En d'autres termes, ce chemin géodésique passe par les pixels *les plus empruntés*. Ce processus se généralise très bien sur des surfaces. LENOIR a montré que cette définition de chemin géodésique sur une grille 2D sans obstacle correspond à une droite discrète 2D. Cependant, dans le cas surfacique ou avec des obstacles, aucune équivalence avec des droites ni aucune propriété métrique de la distance ainsi définie n'est proposée. Cette approche à l'avantage de ne se baser que sur le graphe d'adjacence des points et peut donc se généraliser à d'autres grilles régulières.

Dans ce qui suit, nous proposons une définition de visibilité basée sur la notion de droite discrète (voir chapitre 2). Cette proposition permet l'utilisation d'algorithmes efficaces pour le test de visibilité ainsi que pour le calcul de chemins géodésiques.

Dans un premier temps nous présentons les différentes définitions et le calcul de visibilité proposé. Nous décrivons par la suite un algorithme efficace pour le calcul de chemins géodésiques basé sur notre définition de visibilité. Enfin, nous donnons quelques pistes pour la généralisation en dimension 3.

Cette partie correspond aux articles COEURJOLLY (2002) et COEURJOLLY ET AL. (2002b).

4.4.1 Visibilité dans des domaines discrets

4.4.1.1 Notations et définitions

Dans ce qui suit, nous notons \mathcal{D} un domaine discret correspondant à un ensemble de pixels k -connexes. On notera $\bar{\mathcal{D}}$ le complémentaire de \mathcal{D} . Ce complémentaire sera appelé par la suite indifféremment le *fond* ou *l'ensemble des obstacles*. Par la suite, nous considérons un domaine \mathcal{D} 8-connexe.

Par analogie avec le cas polygonal, nous définissons la visibilité discrète de la façon suivante :

Définition 4.9 (Visibilité discrète) *Soient s et t deux pixels de \mathcal{D} , la visibilité*

discrète est une relation binaire $v \subseteq \mathcal{D} \times \mathcal{D}$ telle que nous avons $v(s, t)$ si et seulement s'il existe un segment de droite discrète 8-connexe joignant s et t dont les pixels appartiennent à \mathcal{D} .

Avant de continuer avec le problème d'étiquetage en visibilité, rappelons les liens entre les droites discrètes et l'intersection de bandes dans l'espace des paramètres vus au paragraphe 2.2.2.1.

La discrétisation *Grid Intersect Quantization* (ou GIQ) d'une droite réelle d'équation $y = \alpha x + \beta$ est l'ensemble des points discrets vérifiant :

$$\Delta(\alpha, \beta) = \{(x, y) \in \mathbb{Z}^2 \mid -\frac{1}{2} \leq \alpha x + \beta - y < \frac{1}{2}\}$$

Le choix du processus de discrétisation n'a aucune influence sur la méthode. Nous avons choisi GIQ pour sa propriété *d'approximation au plus proche* de la droite réelle.

Dans l'espace des paramètres, l'équation précédente nous permet de définir l'ensemble des droites réelles dont les discrétisations contiennent un pixel $p(x, y)$:

$$S_p = \{(\alpha, \beta) \in \mathbb{R}^2 \mid -\frac{1}{2} + y \leq \alpha x + \beta < \frac{1}{2} + y\}$$

Ainsi, un pixel de \mathcal{D} définit une *bande* dans l'espace des paramètres délimitée par les contraintes $L_1 : \alpha x + \beta - y \geq -1/2$ et $L_2 : \alpha x + \beta - y < 1/2$. Rappelons que pour savoir si un ensemble de pixels est un morceau de droite discrète, il suffit de vérifier que l'intersection des bandes données par les pixels est non-vide dans l'espace des paramètres. Le domaine résultant de cette intersection définit toutes les droites réelles dont les discrétisations contiennent l'ensemble de pixels considéré (voir l'exemple donné figure 4.31).

On notera par la suite $\mathcal{S}(s, t)$ l'intersection des bandes S_s et S_t , respectivement associées à deux pixels s et t .

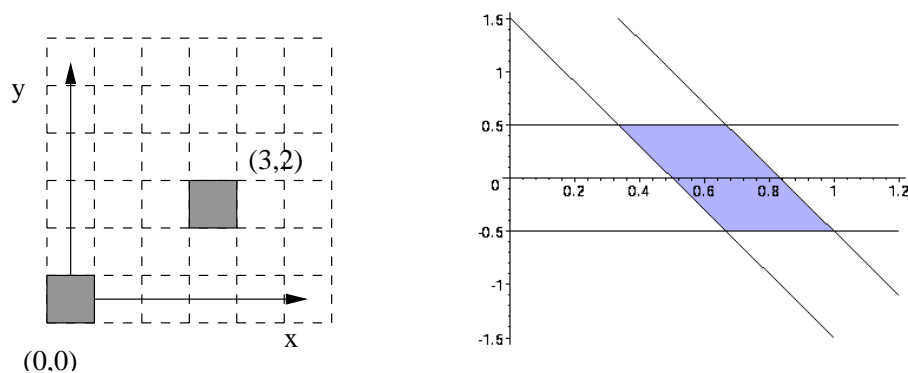


FIG. 4.31 – Exemple de domaine $\mathcal{S}(s, t)$ des pixels $(0, 0)$ et $(3, 2)$: le domaine $\mathcal{S}(s, t)$ est défini par les inéquations $\{\beta < 1/2, \beta \geq -1/2, \beta < -3\alpha + 5/2, \beta \geq -3\alpha + 3/2\}$.

En ce qui concerne le problème de visibilité entre s et t dans des domaines non-convexes, l'objectif est d'analyser les contraintes liées aux obstacles et aux deux pixels s et t dans l'espace des paramètres.

4.4.1.2 Domaine de visibilité

Soit o un pixel obstacle (*i.e.* $o \in \bar{\mathcal{D}}$). L'ensemble des droites réelles dont la discrétisation ne contient pas le pixel o se décrit simplement dans l'espace des paramètres. En

effet, il nous suffit de considérer les négations des contraintes L_1 et L_2 précédentes. Ainsi, nous associons à un pixel obstacle o les contraintes $\bar{L}_1(o) : \alpha x + \beta - y < -1/2$ et $\bar{L}_2(o) : \alpha x + \beta - y \geq 1/2$.

Maintenant, pour décider si le pixel o empêche la visibilité entre s et t , il nous suffit de tester si le domaine, dans l'espace des paramètres, $L_1(s) \cap L_2(s) \cap L_1(t) \cap L_2(t) \cap (\bar{L}_1(o) \cup \bar{L}_2(o))$ est vide ou non.

Plus généralement, nous considérons un ensemble $\mathcal{O} = \{o_i\}_{i=1..n}$, sous-ensemble de $\bar{\mathcal{D}}$, de pixels obstacles tel que toutes les abscisses de ces points sont comprises entre les abscisses de s et de t (tous les autres obstacles pouvant être supprimés du problème de visibilité entre s et t). Ainsi nous avons :

Lemme 4.3 *Soient s et t deux points de \mathcal{D} , s et t sont visibles dans \mathcal{D} si et seulement si :*

$$\mathcal{S}(s, t) \cap \left(\bigcap_{i=1..n} (\bar{L}_1(o_i) \cup \bar{L}_2(o_i)) \right) \neq \emptyset$$

La preuve de ce lemme est directement induite de la définition de la visibilité. Le domaine issu de cette intersection est appelé **domaine de visibilité** de s et t (voir figure 4.32).

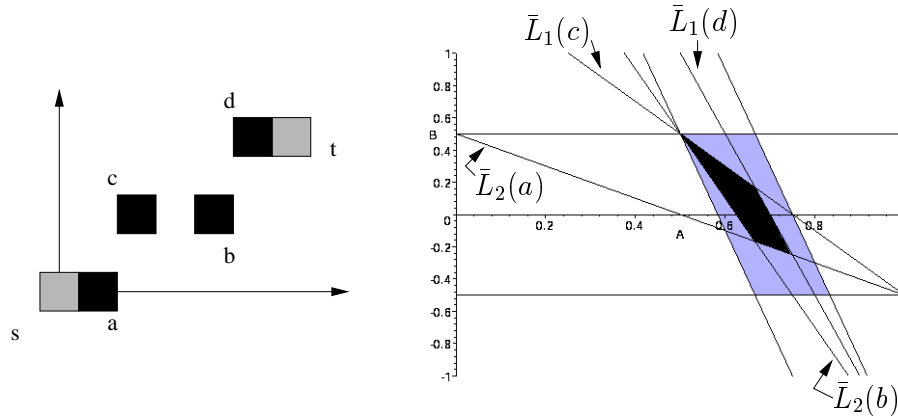


FIG. 4.32 — *Domaine de visibilité associé à un ensemble de pixels : le domaine en gris clair correspond à $\mathcal{S}(s, t)$ et le polygone noir au domaine de visibilité associé à s, t et aux pixels bloquants a, b, c et d .*

Dans ce qui suit, nous cherchons à caractériser l'ensemble des pixels les plus contraignants pour notre problème. Plus formellement, ces pixels sont ceux définis comme suit :

Définition 4.10 (Pixel bloquant) *Un pixel o de \mathcal{O} est dit bloquant pour le test de visibilité $v(s, t)$ si nous avons :*

$$\mathcal{S}(s, t) \cap (\bar{L}_1(o) \cup \bar{L}_2(o)) \neq \mathcal{S}(s, t)$$

et l'abscisse de o est entre les abscisses de s et t .

Nous pouvons présenter un premier lemme :

Lemme 4.4 *Soit o un pixel bloquant pour le test de $v(s, t)$, ou bien le domaine $\mathcal{S}(s, t) \cap (\bar{L}_1(o) \cup \bar{L}_2(o))$ est vide, ou bien il n'a qu'une seule composante connexe.*

Preuve : Nous considérons le domaine $\mathcal{S}(s, t)$ et un pixel bloquant o tel que o, s et t ne sont pas alignés (dans le cas contraire, le domaine de visibilité est vide).

Nous montrons que soit $\bar{L}_1(o)$ soit $\bar{L}_2(o)$ intersecte le domaine $\mathcal{S}(s, t)$ mais jamais les deux. Sans perte de généralité, nous pouvons considérer les différents cas illustrés dans la figure 4.33-(a). Étant donné que l'abscisse de o doit être entre celle de s et t , la pente des contraintes $\bar{L}_1(o)$ et $\bar{L}_2(o)$ doit nécessairement être comprise entre les pentes des contraintes associées à s et t . Nous pouvons donc supprimer les deux premiers cas de la figure 4.33-(a).

De plus, par construction, la hauteur verticale entre L_1 et L_2 est égale à 1. Ainsi, l'intersection en a' entre $\bar{L}_1(o)$ et la droite verticale passant par b implique que l'intersection b' , entre $\bar{L}_2(o)$ et cette même droite verticale, est en dehors du segment $[ab]$ (voir figure 4.33-(b)). De plus, étant donné que la pente de $\bar{L}_2(o)$ est plus grande, en valeur absolue, que celle de la droite (cb) , $\bar{L}_2(o)$ ne peut pas intersecter le domaine $\mathcal{S}(s, t)$.

De la même manière, si $\bar{L}_2(o)$ coupe le domaine, la contrainte $\bar{L}_1(o)$ ne le coupe pas. Le dernier cas de figure de la figure 4.33-(a) n'est donc pas valide.

Finalement, $\mathcal{S}(s, t) \cap (\bar{L}_1(o) \cup \bar{L}_2(o))$ est égal, soit à $\mathcal{S}(s, t) \cap \bar{L}_1(o)$, soit à $\mathcal{S}(s, t) \cap \bar{L}_2(o)$. Ce domaine correspond donc à une intersection de contraintes linéaires, il n'a donc qu'une seule composante connexe qui est de plus convexe.

□

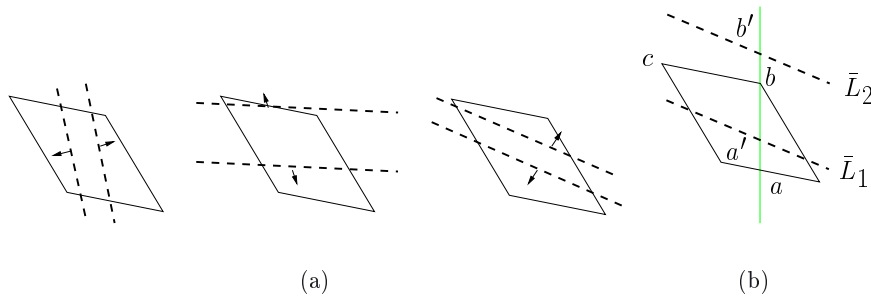


FIG. 4.33 – Illustration de la preuve du lemme 4.4 : (a) différents cas de figure qui introduisent deux composantes connexes et (b) notation pour la preuve du lemme.

D'un point de vue géométrique, si un pixel o est bloquant pour $v(s, t)$ (avec $[st]$ dans le premier octant) et si celui-ci est au dessus du segment $[st]$, seule la contrainte \bar{L}_1 doit être prise en compte. De la même manière, si o est sous $[st]$, seule la contrainte \bar{L}_2 intervient (voir figure 4.32 pour une illustration). Cette remarque nous permet d'avoir une première caractérisation géométrique de cet ensemble de pixels bloquants.

Nous pouvons étendre cette analyse proposée dans COEURJOLLY (2002); COEURJOLLY ET AL. (2002b) par le théorème suivant :

Théorème 4.2 *Étant donné s et t dans \mathcal{D} , le nombre de pixels bloquants suffisant pour évaluer $v(s, t)$ est au plus de deux. Le domaine de visibilité est calculé à partir de $\mathcal{S}(s, t)$ avec au plus quatre contraintes.*

Preuve : Pour plus de clarté, nous choisissons la représentation arithmétique des droites discrètes. Nous avons déjà montré dans le paragraphe 2.2.2.3 les liens entre cette représentation et l'analyse duale.

Sans perte de généralité, soit $[st]$ dans le premier octant et $(a, b) \in \mathbb{Z}^2$ tel que a/b est la pente de $[st]$ (avec $\text{pgcd}(a, b) = 1$). Une droite discrète naïve (8-connexe) qui passe par s et t est composée de l'union de b pointillés consécutifs parmi les $2b - 1$ pointillés :

$$\begin{aligned} ax - by &= -b + 1 \\ &\dots \\ ax - by &= b - 1 \end{aligned}$$

Un pixel obstacle $o(x_o, y_o)$ de reste $r = ax_o - by_o$ invalide tout le pointillé r de l'ensemble $\{-b + 1, \dots, b - 1\}$. De plus, si celui-ci est tel que $r < 0$, il invalide tous les pointillés $\{-b + 1, \dots, r\}$. En effet, nous ne pouvons pas trouver b pointillés consécutifs dans cet ensemble. Ainsi, un autre pixel obstacle dont le reste est dans cet ensemble ne sert plus pour $v(s, t)$ puisqu'il a déjà été exclu par o .

De même, si $r > 0$, tous les pointillés de l'ensemble $\{r, \dots, b - 1\}$ sont invalidés ainsi que tous les pixels obstacles dont le reste est dans cet ensemble.

Ainsi, parmi l'ensemble \mathcal{O} des pixels bloquants, seuls deux pixels nous intéressent :

- parmi les pixels bloquants de restes négatifs, celui dont le reste r_{max} est maximal. Ce point est noté o_{max} .
- Parmi les pixels bloquants de restes positifs, celui dont le reste r_{min} est minimal, noté o_{min} .

Ainsi, il existe une droite discrète si et seulement s'il reste b pointillés non invalidés et donc si :

$$r_{min} - r_{max} + 1 \geq b$$

Si nous revenons au domaine de visibilité $\mathcal{S}(s, t)$, quatre points sont nécessaires. Ils correspondent aux pixels bloquants de reste r_{max} et r_{min} d'abscisse minimale et maximale entre s et t . De plus, grâce au lemme précédent, une seule contrainte, \bar{L}_1 ou \bar{L}_2 , est à prendre en compte. Ainsi, le domaine de visibilité est exactement calculé à partir de $\mathcal{S}(s, t)$ et des quatre contraintes issues des pixels bloquants précédent.

□

Une illustration de ce théorème est donnée dans la figure 4.32 : les pixels c et d sont tous les deux sur le pointillé d'abscisse r_{max} pour le segment $[s, t]$. De même, les points a et b sont sur le pointillé r_{min} .

4.4.1.3 Algorithme de calcul de visibilité

Dans ce paragraphe, nous présentons un algorithme efficace permettant de calculer la classe d'équivalence de la relation de visibilité associée à un pixel source s . En d'autres termes, étant donné un point s de \mathcal{D} , nous voulons marquer tous les pixels de \mathcal{D} qui sont visibles de s . Pour cela, nous proposons deux algorithmes : le premier correspond au calcul de la classe de visibilité basée sur les définitions ci-dessus. Le second algorithme se base sur une version modifiée de la définition de visibilité mais permet un étiquetage plus efficace du domaine.

L'algorithme 22 présente donc une implémentation directe des définitions et théorèmes précédents. Cet algorithme se base sur un parcours en largeur du domaine \mathcal{D} , implémenté par la structure de file FIFO. À chaque étape de l'algorithme, nous maintenons une liste des obstacles \mathcal{O} et nous étiquetons en visibilité tous les pixels de \mathcal{D} rencontrés.

Algorithme 22 Classe d'équivalence de visibilité à partir d'une source s

CALCUL_VISIBILITÉ(s, \mathcal{D})

```

1: Soit  $Q$  une file FIFO
2: Soit  $\mathcal{O}$  une liste de pixels obstacles
3: AJOUT_FILE( $s, Q$ )
4: tant que  $Q$  non vide faire
5:    $t = \text{PREMIER\_ELEMENT}(Q)$ 
6:   pour tout  $p$ , 8-voisin de  $t$  non étiqueté fermé ou visible faire
7:     si  $p$  est un obstacle alors
8:       AJOUT_LISTE( $p, \mathcal{O}$ )
9:     sinon
10:      Soit  $(a, b, c, d)$  les quatre contraintes de  $\mathcal{O}$  décrit dans le théorème 4.2
11:      Évaluer  $v(s, p)$  avec  $(a, b, c, d)$ 
12:      si  $p$  est visible de  $s$  alors
13:        Marquer  $p$  comme étant visible
14:        AJOUT_FILE( $p, Q$ )
15:      sinon
16:        Marquer  $p$  comme étant fermé
17:      fin du si
18:    fin du pour
19:  fin du tant que

```

Analysons la complexité de cet algorithme, pour cela nous notons n le nombre de pixels de \mathcal{D} . Dans l'algorithme 22, le maintien de la file FIFO et de la liste des obstacles se fait en temps constant à chaque étape. Grâce au théorème 4.2, le test de visibilité $v(s, p)$ se fait en $O(1)$ si nous connaissons le quadruplet (a, b, c, d) . Finalement, le point clef de la complexité est donné par la ligne 10 de cet algorithme qui consiste à extraire de \mathcal{O} le quadruplet de pixels bloquants. Or, aucune structure de données ne nous permet d'accéder à ces éléments rapidement dans le cas de la visibilité présenté ci-dessus. Ainsi, si nous notons m le nombre d'obstacles, la complexité globale de cet algorithme est en $O(nm)$, ce qui est assez coûteux, surtout dans un objectif de calcul de géodésiques.

Nous proposons maintenant un second algorithme basé sur une définition légèrement différente de la visibilité. Celle-ci nous permet de mettre en place une structure de données efficace. Nous définissons ainsi la notion de visibilité restreinte :

Définition 4.11 (Visibilité restreinte) *Soient s et t deux pixels de \mathcal{D} , la visibilité restreinte est une relation binaire $v^* \subseteq \mathcal{D} \times \mathcal{D}$ telle que nous avons $v^*(s, t)$ si et seulement s'il existe une demi-droite réelle qui passe par s et dont la discrétisation contient t et aucun pixel de $\bar{\mathcal{D}}$.*

Dans cette définition, nous contraignons donc l'ensemble des droites à passer exactement par s . Cette version de la visibilité est un affaiblissement de la précédente puisque la relation binaire que l'on obtient n'est plus symétrique. Cependant,

cette définition nous permet de proposer une structure de données et un algorithme permettant un étiquetage en visibilité très efficace.

Supposons donc que tous les pixels obstacles sont triés dans un ordre polaire de centre s dans le sens trigonométrique. Nous avons donc :

Proposition 4.3 *Soient un ensemble de pixels obstacles triés en ordre polaire de centre s et un point t . Nous notons u le successeur de t dans le tri polaire et l son prédécesseur. Nous avons alors :*

$$v^*(s, t) \Leftrightarrow \mathcal{S}^*(s, t) \cap \bar{L}_1(u) \cap \bar{L}_2(l) \neq \emptyset$$

où \mathcal{S}^* correspond au domaine des droites vérifiant la contrainte de la visibilité restreinte. Ce domaine correspond à un segment dans l'espace des paramètres.

La preuve de cette proposition est directement induite de la définition de v^* ainsi que des lemmes précédents. Remarquons enfin que le tri polaire des pixels obstacles peut se calculer entièrement en nombres entiers (voir par exemple CORMEN ET AL. 1990).

Nous pouvons maintenant présenter l'algorithme du calcul de visibilité (voir algorithme 23).

Algorithme 23 Classe d'équivalence de visibilité restreinte à partir d'une source s

CALCUL_VISIBILITÉ_RESTREINTE(s, \mathcal{D})

```

1: Soit  $Q$  une file FIFO
2: Soit  $\mathcal{O}$  une liste de pixels obstacles triés en polaire de centre  $s$ 
3: AJOUT_FILE( $s, Q$ )
4: tant que  $Q$  non vide faire
5:    $t =$ PREMIER_ELEMENT( $Q$ )
6:   pour tout  $p$ , 8-voisin de  $t$  non étiqueté fermé ou visible faire
7:     si  $p$  est un obstacle alors
8:       AJOUT_TRI_POLAIRE( $p, \mathcal{O}$ )
9:     sinon
10:      Soit  $u$  le successeur de  $p$  dans le tri polaire  $\mathcal{O}$ 
11:      Soit  $l$  le prédécesseur de  $p$  dans le tri polaire  $\mathcal{O}$ 
12:      Évaluer  $v^*(s, p)$  avec  $\mathcal{S}^*(s, p) \cap \bar{L}_1(u) \cap \bar{L}_2(l) \neq \emptyset$ 
13:      si  $p$  est visible de  $s$  alors
14:        Marquer  $p$  comme étant visible
15:        AJOUT_FILE( $p, Q$ )
16:      sinon
17:        Marquer  $p$  comme étant fermé
18:      fin du si
19:    fin du pour
20:  fin du tant que
21: fin du tant que

```

Dans cet algorithme, le test de visibilité s'effectue aussi en temps constant mais cette fois, l'extraction, à partir de \mathcal{O} , des pixels bloquants *intéressants* pour notre définition de visibilité se fait en $O(\log(m))$ au lieu de $O(m)$. De plus, le maintien de la liste triée, c'est-à-dire le coût de l'insertion d'un obstacle, se fait lui aussi en $O(\log(m))$. Finalement, le coût global de cet algorithme est en $O(n \log(m))$.

Nous pouvons encore optimiser le coût moyen de ce dernier avec la remarque suivante : lors du parcours en largeur du graphe, quand nous insérons les 8-voisins de

t dans la structure, nous mémorisons le cône (s, u, l) qui correspond à la localisation de t dans \mathcal{O} . Ainsi, lorsque nous dépilerons les fils de t , nous mettrons à jour leur localisation à partir de (s, u, l) . De cette façon, nous obtenons un coût de mise-à-jour très faible.

Dans une analyse au pire cas, cette optimisation ne permet pas de réduire la complexité, cependant, si nous considérons des ensembles pixels obstacles assez compacts dans \mathcal{D} , le coût effectif de cette mise à jour sera très faible.

4.4.2 Plus court chemin discret et métrique géodésique discrète

Nous pouvons maintenant définir les notions de chemins géodésiques discrets basées sur le calcul de visibilité.

4.4.2.1 Définitions et état de l'art

Dans un premier temps, rappelons les deux grandes approches pour les définitions de métriques discrètes vues précédemment :

- approches par masque (métriques de chanfrein) : on considère un ensemble de déplacements élémentaires pondérés ;
- approches basées sur une gestion de déplacement vectoriels (d_x, d_y) .

Dans le cas de la métrique géodésique, les approches basées sur des masques offrent une solution très efficace. En effet, à partir de \mathcal{D} et des déplacements élémentaires considérés, nous pouvons construire un graphe d'adjacence pondéré sur lequel nous pouvons appliquer un algorithme classique de plus court chemin dans un graphe comme celui de DIJKSTRA (PIPER et GRANUM, 1987). Pour implémenter cet algorithme de plus court chemin, nous pouvons utiliser des **files de priorité** (ou *bucket lists* en anglais). Cette structure est classique en algorithmique (voir par exemple CORMEN ET AL. 1990) et son utilisation pour le calcul de distance géodésique par masque a été proposée par VERWER ET AL. (1989). Celle-ci nous permet d'avoir un algorithme à coût uniforme pour l'étiquetage en distance géodésique.

Cette structure, illustrée figure 4.34, est construite de la façon suivante : nous considérons une collection de files FIFO indicées par un entier d . Pour notre problème, tous les pixels se trouvant dans la file d'indice d sont à une distance d de notre point d'origine. Le principe de l'algorithme de plus court chemin basé sur cette technique est le suivant : l'algorithme commence par ajouter le point s dans la file d'indice 0. Ensuite, à chaque étape on dépile un pixel p , encore non visité, de la file d'indice minimum non vide, notée d_{min} , et pour chaque voisin n de p , nous ajoutons n à la file d'indice $d_{min} + weight(p, n)$. $Weight(p, n)$ correspond au poids associé au déplacement \vec{pn} dans le masque de chanfrein utilisé.

Cette structure permet donc une implémentation très efficace de l'algorithme de DIJKSTRA. De plus, VERWER ET AL. (1989) proposent une gestion des allocations des files très efficaces, par exemple, nous pouvons réutiliser les files de manière circulaire une fois que celles-ci sont vides.

Dans une approche basée sur la distance euclidienne, MOREAU (1995) propose un étiquetage du domaine en maintenant des listes de pixels du front de propagation. Ce processus de propagation se base sur une construction incrémentale de cercles à rayon croissant. Ainsi, quand aucun obstacle n'est rencontré, le processus construit des cercles de rayon croissant en étiquetant les pixels par la distance euclidienne au

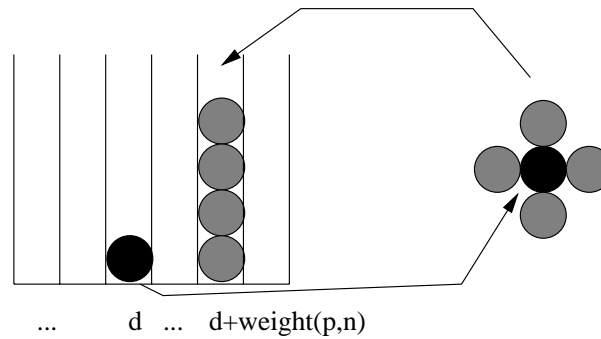


FIG. 4.34 – Structure de files de priorité : ensemble de files FIFO indicées par la distance au pixel source s .

centre du cercle. Si un obstacle est rencontré on subdivise le cercle en arcs avec la possibilité d’initialiser de nouveaux centres d’arcs de cercle. La structure de données de cet algorithme n’est cependant pas évidente à manipuler et des processus coûteux en temps peuvent apparaître lors de la fusion ou de la subdivision d’arcs.

En se basant sur l’utilisation de sa transformée en distance euclidienne par propagation (voir paragraphe 4.2.2.1), CUISENAIRE (1999) propose un algorithme de calcul de distance géodésique que l’on peut qualifier de local. En effet, étant donné un certain rayon r , tous les pixels dans la boule euclidienne de rayon r centré sur le pixel source s sont étiquetés par cette distance euclidienne. Ensuite, le processus de propagation est itéré avec les pixels non étiquetés sur le front de propagation. Ainsi, si le rayon choisi est grand, la distance calculée dans les domaines où il y a peu d’obstacles sera une bonne approximation de la distance euclidienne. Cependant, l’approximation sera mauvaise dans les parties fortement contraintes. Ainsi, la qualité de l’approximation est fonction du domaine \mathcal{D} et du rayon r choisis.

Enfin, SOILLE (1991, 1994, 1999) propose l’utilisation de la notion de visibilité pour l’estimation de chemins géodésiques. Dans le paragraphe suivant, nous présentons une approche similaire mais utilisant le calcul efficace de visibilité présenté ci-dessus.

4.4.2.2 Algorithme d’étiquetage en distance géodésique

L’idée principale de l’algorithme est la suivante : pour tous les points visibles depuis le point source s , l’étiquetage en distance géodésique est trivial puisqu’il correspond exactement à la distance euclidienne calculée à partir du vecteur de déplacement (d_x, d_y) . Par contre, si un point p n’est pas visible depuis la source, nous initialisons un nouveau processus d’étiquetage en visibilité à partir de p . Ainsi, nous étiquetons en distance les pixels visités, visibles de p , par la distance entre p et la source originale s à laquelle nous ajoutons la distance euclidienne entre ces points et p .

Définissons tout d’abord la notion de chemin géodésique discret :

Définition 4.12 (Chemin géodésique discret) *Un chemin géodésique discret entre deux points s et p de \mathcal{D} est une séquence de pixels $\{p_i\}_{i=0..n+1} \in \mathcal{D}$, avec $p_0 = s$ et $p_{n+1} = t$, telle que :*

$$v(p_i, p_{i+k}) \quad \text{pour } k = \{-1, 0, 1\} \quad \text{et } i = 1..n$$

et telle que la distance géodésique $d_{geodes}(s, t)$ est minimale. Cette distance est définie

par :

$$d_{geodes}(s, t) = \sum_{i=0}^n d_{euc}(p_i, p_{i+1})$$

où $d_{euc}(a, b)$ représente la distance euclidienne classique.

En d'autres termes, un chemin géodésique discret est une séquence de segments de droite discrète de longueur minimale joignant s et t . La métrique géodésique proposée pour cette séquence de segments correspond exactement à celle utilisée pour estimer la longueur d'une courbe discrète basée sur une segmentation en droites discrètes (voir paragraphe 5.4).

Afin d'obtenir un algorithme d'étiquetage efficace, nous utilisons la structure de files de priorité de VERWER ET AL. dont les indices sont donnés par $\lceil d_{geodes}(s, p) \rceil$. Cette métrique est consistante avec l'algorithme de VERWER car elle satisfait l'inégalité triangulaire (MOREAU, 1995) :

$$\text{for } a, b, c \in \mathbb{R} \quad a + b \geq c \Rightarrow \lceil a \rceil + \lceil b \rceil \geq \lceil c \rceil$$

Toujours dans un souci d'efficacité, nous utilisons la visibilité restreinte $v^*(s, t)$. Pour chaque pixel p dans la structure de file, nous mémorisons : ses coordonnées, les coordonnées du point source s_i duquel p est visible et la distance géodésique $d_{geodes}(s, s_i)$. Nous maintenons aussi une structure triée de pixels obstacles associés à chaque propagation en visibilité.

L'algorithme 24 présente la méthode proposée.

4.4.3 Analyse expérimentale et discussion

Dans notre analyse expérimentale, nous avons appliqué l'algorithme précédent sur un certain nombre de domaines binaires. Ainsi, la figure 4.35 présente les résultats de l'étiquetage en distance géodésique avec les métriques d_4 , d_8 et la métrique géodésique proposée. Les distances sont représentées avec une palette de niveaux de gris réduite afin de mieux observer les fronts de propagations.

Dans la figure 4.36, au lieu d'étiqueter par la distance géodésique, tous les pixels de même couleur sont dans la même classe d'équivalence de visibilité. Ces illustrations nous permettent de proposer une solution au problème *jouet* classique de placement du nombre minimal de gardes dans un musée. En effet, chaque couleur correspond à la zone de visibilité de chaque garde et le domaine \mathcal{D} correspond au musée. Ainsi, si le premier garde est fixé (source s du problème) nous sommes capable de calculer le nombre minimal de gardes pour couvrir tout le domaine et de placer ceux-ci.

La figure 4.37 présente une application de ces algorithmes en imagerie médicale. L'objectif est ici de mesurer la longueur de vaisseaux sanguins dans des images angiographique segmentées.

Enfin, si nous appliquons cet algorithme d'étiquetage en distance géodésique à partir de plusieurs sources, le résultat obtenu est très proche d'une transformée en distance euclidienne de ces points sources. Cependant, étant donné que notre processus de propagation du domaine se fait localement, nous ne pouvons pas éviter les erreurs de type DANIELSON présentés dans les paragraphes précédents.

Algorithme 24 Étiquetage en distance géodésique à partir d'une source s .

CALCUL_GÉODÉSIQUE(s, \mathcal{D})

```

1: Soit  $Bucket[i]$  un ensemble de files FIFO
2: Soit  $\mathcal{O}[i]$  un ensemble de listes de pixels bloquants triés en polaire
3: Soit  $d_{min}$  l'indice minimal des files non vides initialisé à 0
4: AJOUT_FILE( $s, Bucket[d_{min}]$ )
5: tant que il reste une file non vides faire
6:   si la file  $Bucket[d_{min}]$  est vide alors
7:     ajuster  $d_{min}$ 
8:   fin du si
9:    $t = \text{PREMIER\_ELEMENT}(Bucket[d_{min}])$ 
10:  pour tout  $p$ , 8-voisin de  $t$  non étiqueté fermé ou visible faire
11:    Soit  $i$  l'indice de la liste d'obstacles associé à la source  $s_i$  de  $t$ 
12:    si  $p$  est un obstacle alors
13:      AJOUT_TRI_POLAIRE( $p, \mathcal{O}[i]$ )
14:    sinon
15:      Soient  $u$  et  $l$  la localisation de  $p$  dans le tri polaire  $\mathcal{O}[i]$ 
16:      Évaluer  $v^*(s, p)$  avec  $\mathcal{S}^*(s, p) \cap \bar{L}_1(u) \cap \bar{L}_2(l) \neq \emptyset$ 
17:      si  $p$  est visible de  $s$  alors
18:        Marquer  $p$  comme étant visible
19:        Calculer la distance géodésique  $d'$  de  $n$ 
20:        si  $d' > d_{min}$  alors
21:          AJOUT_FILE( $p, Bucket[\lceil d'x \rceil]$ )
22:        fin du si
23:      sinon
24:        Marquer  $p$  comme étant fermé
25:      fin du si
26:    fin du si
27:  fin du pour
28: fin du tant que

```

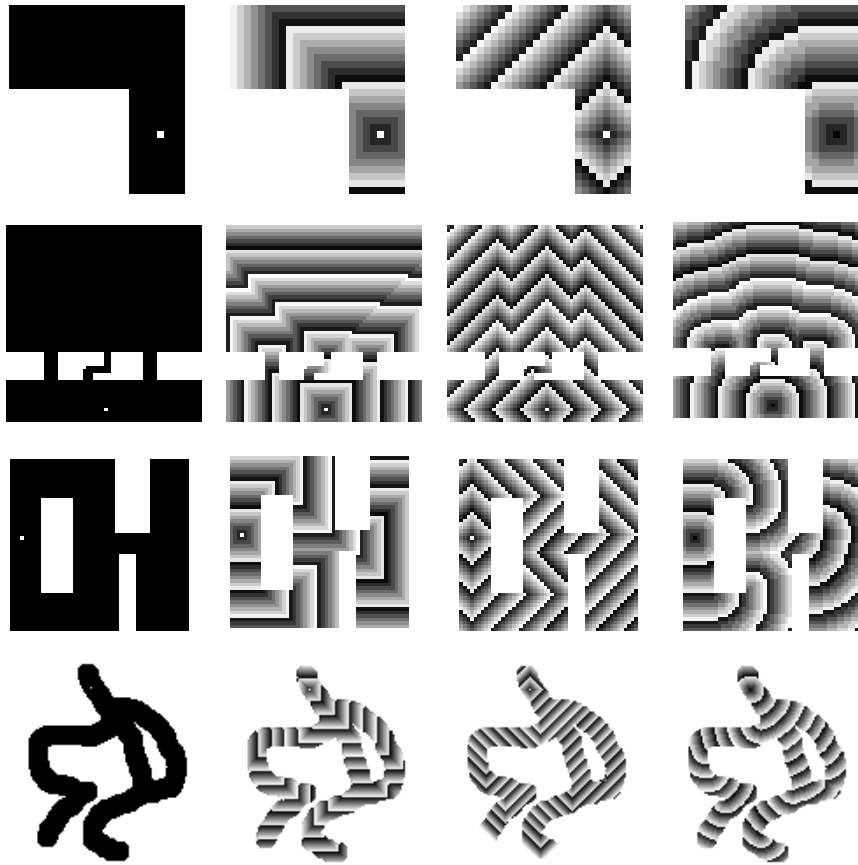


FIG. 4.35 – : Analyse expérimentale de l'étiquetage en distance géodésique : de gauche à droite, les domaines discrets et le point source (pixel blanc isolé), l'étiquetage avec la distance d_8 , avec la distance d_4 et enfin avec la distance géodésique proposée.

4.4.4 Vers une extension en 3D

Une extension naturelle de ce travail consisterait en une analyse de la visibilité et des chemins géodésiques en dimensions supérieures. Nous nous intéressons ici au cas de la dimension 3 et considérons un domaine \mathcal{D} de voxels 26-connexes.

Dans le cas des approches basées sur des distances de chanfrein, des algorithmes efficaces peuvent être proposés pour l'étiquetage en distance géodésique. En effet, nous nous ramenons de toute façon à un problème de graphe (KIRYATI et SZÉKELY, 1993; SANNITI DI BAJA et SVENSSON, 2000).

Dans notre approche de chemins géodésiques basés sur la visibilité, nous pouvons reprendre les mêmes définitions : on dira que deux voxels sont visibles s'il existe un segment de droite discrète 26-connexe tel que tous les voxels sont dans \mathcal{D} , passant par les deux voxels. Nous renvoyons le lecteur au chapitre 2.3 pour plus de détails sur les droites discrètes 3D. Cependant, l'étiquetage en visibilité dans des domaines non-convexes 3D est plus complexe que dans le cas 2D. Pour illustrer cela, regardons une version 3D de la visibilité restreinte. Nous cherchons donc à décrire les droites réelles passant exactement par le centre d'un voxel s et dont la discrétisation contient un voxel t de \mathcal{D} et aucun point de $\bar{\mathcal{D}}$.

Sans perte de généralité, nous nous plaçons dans le premier 48^{ème} d'espace, c'est-à-dire nous considérons $\vec{st} = (a, b, c)^T$ tel que $a \geq b \geq c > 0$. Ainsi, une droite qui

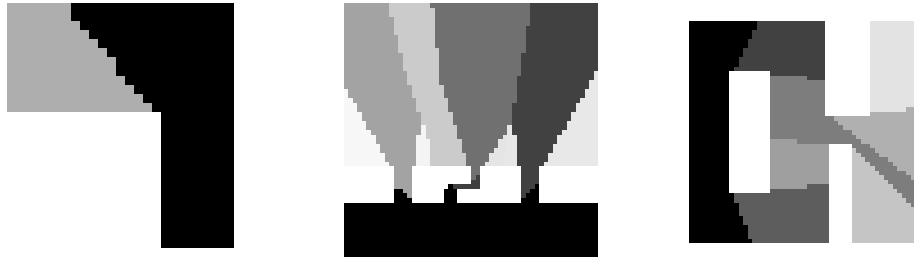


FIG. 4.36 – : *Diagramme de visibilité discrète : tous les pixels de même couleur sont dans la même classe d'équivalence de visibilité, les domaines et points sources sont ceux de la figure 4.35.*

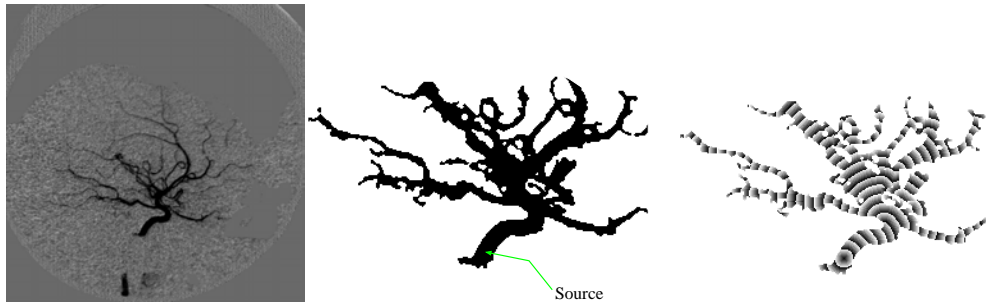


FIG. 4.37 – : *Application de la distance géodésique en imagerie médicale : de gauche à droite, l'image angiographique de vaisseaux sanguins, le domaine binaire issu d'une segmentation et l'étiquetage en distance géodésique.*

passé par s et qui intersecte le carré de longueur de côté 1 centré en t dans le plan (Oyz) est solution au problème de visibilité restreinte (voir figure 4.38-(a)). Si maintenant nous ajoutons des obstacles dans le même $48^{\text{ème}}$ d'espace, nous construisons des pyramides similaires et le point t sera visible de s s'il existe une droite passant par s , qui intersecte le carré centré en t mais ne coupe pas les carrés associés aux obstacles.

Maintenant, projetons l'ensemble de ces carrés sur le plan parallèle à (Oyz) contenant t en utilisant la projection perspective de centre s . Ainsi, t sera visible de s si et seulement si le carré associé à t n'est pas entièrement recouvert par ceux associés aux voxels obstacles (voir figure 4.38-(b)).

D'un point de vue algorithmique, même le cas de la visibilité restreinte n'est pas évident. En effet, l'écriture d'un algorithme efficace qui teste la couverture d'un carré par d'autres n'est pas trivial. De plus aucune structure géométrique évidente des voxels obstacles ne peut être mis en place en 3D pour cette visibilité.

Nous pouvons cependant proposer une première solution en généralisant en 3D l'approche de SOILLE (1991, 1994, 1999). Dans ce cas, le test de visibilité est basé sur un tracé de droite discrète 26-connexe en utilisant un des algorithmes présenté dans le paragraphe 2.3). Ainsi, en utilisant ce test de visibilité et les mêmes structures que pour le cas 2D, nous obtenons un algorithme d'étiquetage en distance géodésique

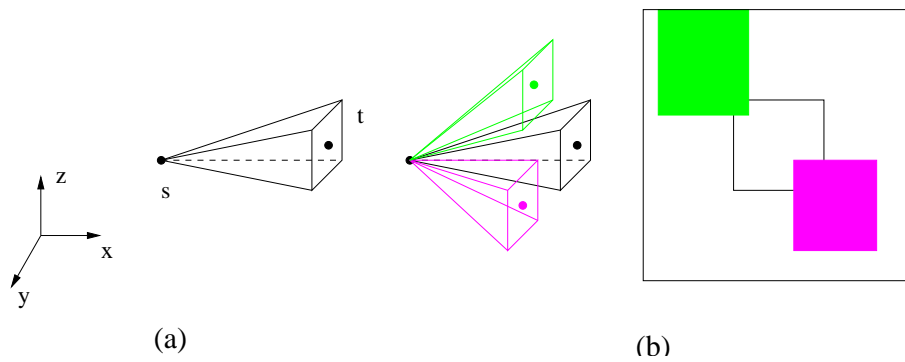


FIG. 4.38 – : *Illustration de la visibilité restreinte en 3D : (a) domaines de visibilité restreinte dans l'espace primal et (b) illustration du test de visibilité par projection dans le plan (Oyz) contenant t des carrés associés aux voxels bloquants.*

dans des domaines 3D. Cette méthode n'est cependant pas très efficace puisque la complexité est en $O(nd)$ où n est le nombre de voxels de \mathcal{D} et d son diamètre (longueur du plus long segment de droite contenu dans \mathcal{D}).

Les figures 4.39 et 4.40 présentent un étiquetage en distance géodésique de domaines 3D non convexes en utilisant une palette de niveaux de gris réduite.

Si nous considérons maintenant une surface discrète comme un ensemble de voxels connexes, l'algorithme précédent nous permet de définir la notion de géodésiques surfaciques. En effet, les voxels d'une telle surface discrète nous donnent le domaine \mathcal{D} . Les voxels obstacles correspondent au complémentaire de \mathcal{D} . La figure 4.41 illustre ces calculs de géodésiques surfaciques.

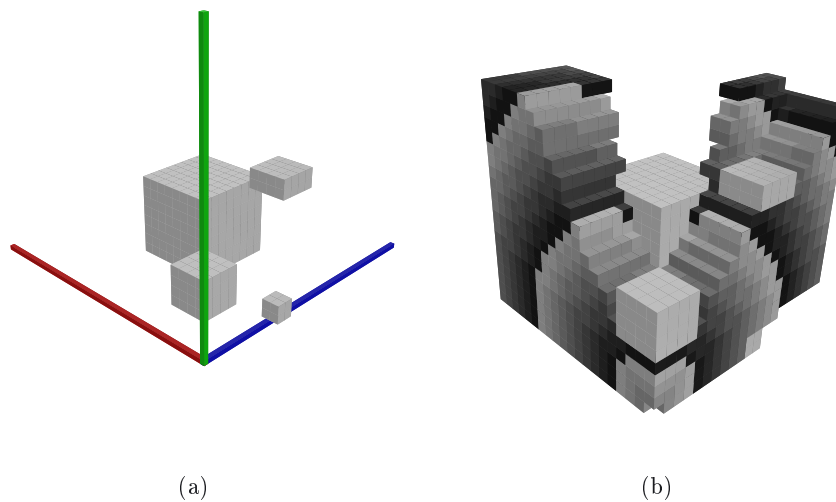


FIG. 4.39 – : *Exemple d'étiquetage en distance géodésique 3D : (a) domaine 3D défini par les obstacles gris et (b), étiquetage en distance géodésique de la première classe de visibilité uniquement. Le voxel source est à l'origine du repère.*

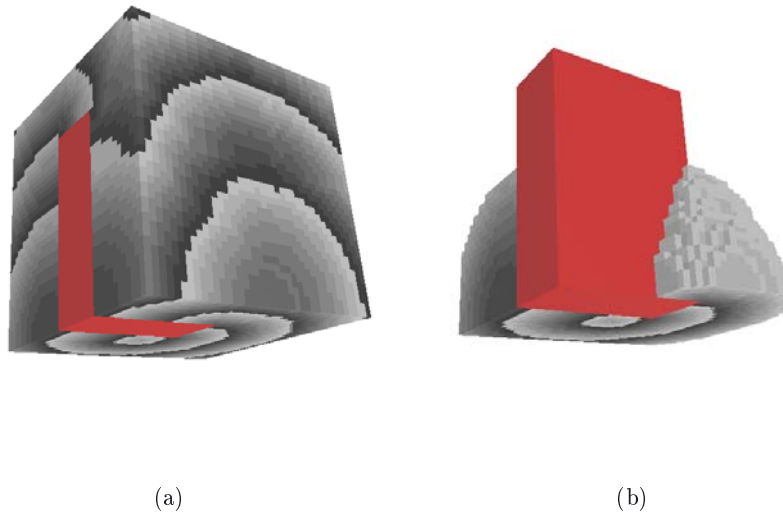


FIG. 4.40 —: *Exemple d'étiquetage en distance géodésique 3D : (a) étiquetage en distance géodésique complet et (b) seuillage de la distance pour mieux observer les structures internes du front d'onde.*

4.4.5 Conclusion et perspectives

Nous avons présenté un test de visibilité discrète basé sur un objet discret classique (la droite discrète). Lors de notre étude de la visibilité discrète, nous avons aussi analysé le comportement de ces droites discrètes dans le cas de domaines non-convexes.

Par la suite, nous avons présenté plusieurs algorithmes permettant : de décider si deux pixels sont visibles étant donné un ensemble de pixels obstacles, de décrire l'ensemble des pixels d'un domaine visibles à partir d'un point. Sur le plan algorithmique nous avons présenté un algorithme d'étiquetage en visibilité efficace dont la complexité est en $O(n \log(m))$ (où n est le nombre de pixels dans \mathcal{D} et m le nombre de pixels obstacles). Nous avons ensuite présenté une définition de chemin géodésique discret basé sur la notion de visibilité ainsi qu'un algorithme de calcul.

Nous nous sommes aussi intéressés au cas tridimensionnel en présentant un premier algorithme ainsi que quelques pistes vers un algorithme efficace.

4.5 Conclusion sur ce chapitre

Dans ce chapitre, nous nous sommes intéressés aux métriques associées à la grille discrète et à l'utilisation de celles-ci pour l'analyse de formes. L'intérêt des descriptions obtenues à l'aide de la transformée en distance euclidienne et du squelette de forme a été montré depuis un certain temps. A la suite de la rédaction de ce chapitre, nous nous sommes aperçus qu'un article récent (MAURER JR. ET AL., 2001) présentait un résultat analogue pour le calcul de la transformée à partir du diagramme de Voronoi discret. L'algorithme présenté a été développé indépendamment des travaux de MAURER JR. ET AL.. Cependant, nous donnons dans ce chapitre des démonstrations et illustrations plus exhaustives. Par contre, la construction d'un squelette basé sur le diagramme de Voronoi discret est originale à notre connaissance.

En ce qui concerne la propagation géodésique, elle peut être utilisée soit di-

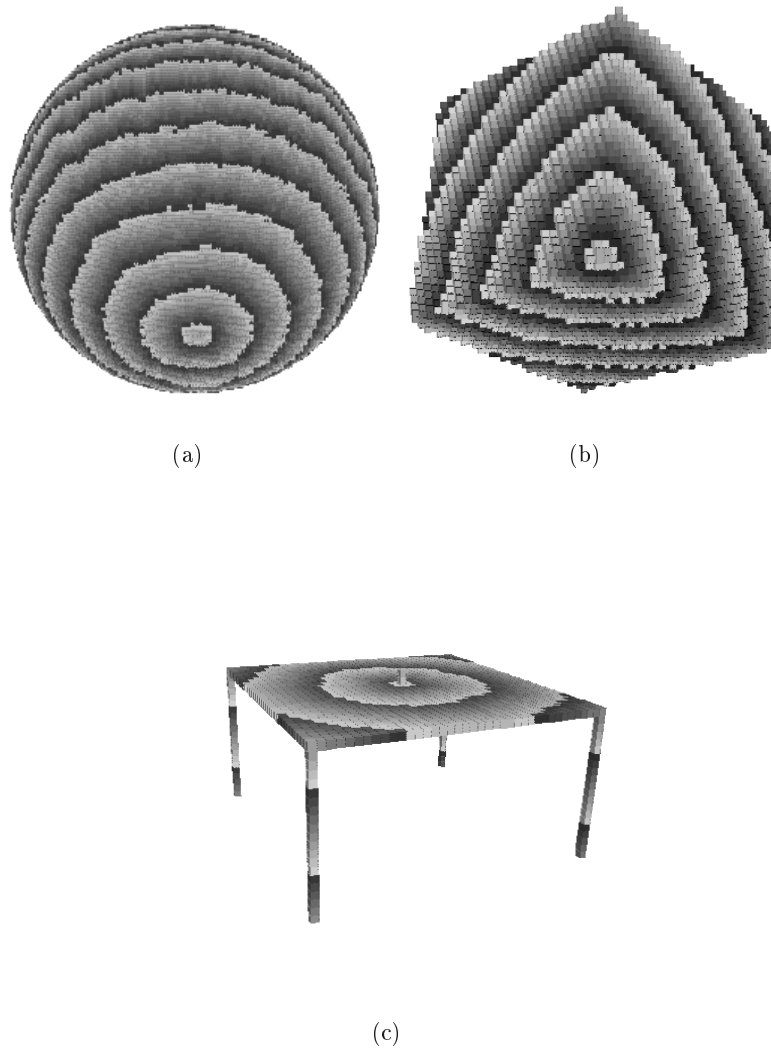


FIG. 4.41 – : *Géodésiques surfaciques : (a) étiquetage en distance géodésique de la surface d'une sphère, (b) de la surface d'un cube et (c) d'une table (le point source est situé au milieu du plateau).*

rectement comme source d'information métrique mais permet aussi une description globale de la forme en considérant les notions de transformée en distance géodésique (SOILLE, 1991, 1999). De plus, la possibilité de construire un *voisinage géodésique* autour d'un voxel d'une surface discrète permet l'écriture d'algorithmes d'estimation de normales et de courbure Gaussienne (voir paragraphes 5.3.2 et 5.5.3.2).

Les perspectives de ces travaux sont nombreuses. Pour le squelette de formes basé sur le diagramme de Voronoï discret, l'étude présentée nous a permis de vérifier la pertinence d'une telle approche. Il nous faut cependant situer un peu plus précisément celle-ci dans la bibliographie et surtout présenter des calculs d'extraction en dimensions quelconques.

Pour la propagation géodésique, nous souhaitons comparer cette approche discrète aux approches basées sur une analyse différentielle (SETHIAN, 1996b,a). Enfin, nous souhaitons aussi poursuivre l'analyse en dimension 3 afin de réduire la complexité de l'algorithme proposé.

Chapitre 5

Mesures sur des objets discrets

Sommaire

5.1	Introduction	161
5.2	Contexte théorique : la convergence asymptotique	161
5.3	Tangentes et normales	163
5.3.1	Tangentes et normales discrètes d'un contour 2D	163
5.3.2	Normales d'une surface discrète	165
5.3.2.1	Approches volumiques	166
5.3.2.2	Approches surfaciques	172
5.3.2.3	Estimateur optimal en temps avec convergence asymptotique	175
5.4	Longueur et aire	176
5.4.1	Longueur d'une courbe discrète 2D	176
5.4.1.1	Approches basées sur des métriques locales	177
5.4.1.2	Approches basées sur des approximations polygonales	178
5.4.1.3	Approches basées sur l'intégration des normales discrètes	181
5.4.1.4	Évaluation comparative	182
5.4.1.5	Remarques sur la plate-forme de tests	185
5.4.2	Longueur d'une courbe discrète 3D	188
5.4.3	Aire d'une surface discrète	192
5.4.3.1	Solutions existantes	192
5.4.3.2	Estimation d'aire de surface discrète par intégration de normales	193
5.4.3.3	Analyse expérimentale	194
5.4.3.4	Conclusion sur l'estimation d'aire d'une surface discrète	197
5.5	Courbures d'objets discrets	197
5.5.1	Quelques définitions mathématiques	197
5.5.2	Courbure de courbes discrètes	201
5.5.2.1	État de l'art	201
5.5.2.2	Estimation du cercle osculateur discret	202
5.5.2.3	Estimation de courbure avec convergence asymptotique	206
5.5.2.4	Courbures de courbes discrètes 3D	208
5.5.3	Courbures de surfaces discrètes	210

5.5.3.1	Courbure Moyenne	210
5.5.3.2	Courbure Gaussienne	213
5.6	Conclusion	218

5.1 Introduction

Dans ce chapitre, nous nous plaçons dans le cadre de l'analyse d'objets basée sur le calcul de mesures euclidiennes sur les contours. En effet, une approche classique de reconnaissance de formes se base sur l'association, au contour de l'objet, d'un vecteur contenant des informations numériques (longueur, aire, graphe de courbure...). On parle alors de la *signature* du contour.

Dans ce qui suit, nous nous intéressons à la création d'un tel vecteur d'attributs dans le cas des courbes et des surfaces discrètes. Nous présentons donc des algorithmes permettant d'estimer la longueur d'une courbe discrète ou l'aire d'une surface discrète, la normale en un point d'une courbe ou d'une surface et enfin les différentes courbures associées à ces objets.

Ces différents estimateurs seront évalués, à la fois numériquement dans des analyses expérimentales, mais aussi sur le plan théorique avec la notion de convergence asymptotique. Pour une meilleure lisibilité de ce chapitre, nous avons placé les preuves de convergence asymptotique dans l'annexe B.

Après avoir présenté ce cadre théorique, nous décrivons tout d'abord les différents estimateurs pour le calcul des normales. Remarquons que la logique voudrait que l'on commence par les estimateurs de longueur ou d'aire, ensuite ceux de normales et enfin les calculs de courbure. Cependant, nous verrons en quoi l'estimation des normales est primordiale pour le calcul de longueur d'une courbe discrète ou d'aire d'une surface.

Afin de faciliter la lecture de ce chapitre, les différentes preuves de convergence asymptotique sont données dans l'annexe B.

5.2 Contexte théorique : la convergence asymptotique

Lorsqu'un estimateur d'une mesure euclidienne est proposé sur une donnée discrète, la justification de celui-ci peut se faire de deux manières différentes : la première justification consiste à montrer la cohérence de l'estimateur par rapport aux propriétés déjà existantes dans le modèle. Une autre justification consiste à prouver la convergence asymptotique de l'estimateur. Dans ce cas, l'estimateur est comparé à la mesure euclidienne approximée par le biais d'un processus de discrétisation sur une grille dont la résolution tend vers l'infini. Un estimateur aura alors la propriété de *convergence asymptotique* si l'erreur commise entre la mesure estimée et la mesure euclidienne tend vers 0 quand la résolution de la grille augmente.

Plus formellement et dans le cas bidimensionnel, nous considérons une courbe euclidienne $\gamma : [0, 1] \rightarrow \mathbb{R}^2$ et une grille orthogonale dans le plan euclidien \mathbb{R}^2 dont la distance entre les points est donnée par $0 < \theta \leq 1$. Nous appelons r la résolution de la grille donnée par $r = 1/\theta$, on parle alors de r -grille. Étant donné un processus de discrétisation de la courbe γ sur la grille, noté $D_r(\gamma)$, nous définissons ainsi un objet multi-grille paramétré par r . Soit \mathcal{F} un descripteur sur la courbe euclidienne γ (la longueur par exemple) et un estimateur \mathcal{E} de la même mesure mais sur le discrétisé $D_r(\gamma)$, nous avons :

Définition 5.1 (Convergence asymptotique) *L'estimateur \mathcal{E} converge asymptotiquement vers \mathcal{F} si et seulement si :*

$$|\mathcal{E}(D_r(\gamma)) - \mathcal{F}(\gamma)| \leq \epsilon(r) \quad \text{avec} \quad \lim_{r \rightarrow \infty} \epsilon(r) = 0$$

L'ordre $O(1/\epsilon(r))$ correspond à la vitesse de la convergence.

L'intérêt d'une telle propriété est double :

- sur le plan théorique, cette propriété nous permet une comparaison formelle entre une mesure continue et une estimation discrète ;
- d'un point de vue pratique, cette convergence nous permet de garantir de bons résultats numériques quelque soit la taille de l'objet discret analysé, ce qui est très important dans une approche multi-échelle par exemple.

Une illustration historique de cette validation remonte aux grecs de l'antiquité : ARCHIMÈDE et LIU HUI (voir par exemple VAN DER WAERDEN 1983) cherchaient à estimer le périmètre d'un cercle, l'objectif principal étant en fait l'estimation du nombre π . Ils utilisaient une approximation basée sur des polygones réguliers à n côtés avec $n_m = 3, 6, 12, 24, 48, 96, \dots$ (voir figure 5.1). Plus précisément, on suppose que ρ est le rayon du cercle et $n_m = 3 \times 2^m$ avec $m \geq 0$. Nous considérons le polygone régulier interne P_{n_m} , possédant n_m arêtes. La longueur p_m d'une arête est donnée par la formule de récurrence :

$$p_{m+1} = \rho \sqrt{2\rho^2 - \rho \sqrt{4\rho^2 - p_m^2}}$$

Ainsi, le périmètre de P_{n_m} est égal à $n_m \cdot p_m$ et donc :

$$\kappa(n_m) = |n_m \cdot p_m - 2\pi\rho| \approx \frac{2\pi\rho}{n_m}, \quad \text{pour } m \geq 1.$$

La fonction $\kappa(n_m)$ définit la vitesse de convergence de l'estimation du périmètre du cercle de rayon ρ en fonction du nombre de côtés du polygone régulier. Dans ce cas, l'estimation du périmètre d'un cercle par des polygones réguliers converge asymptotiquement avec une vitesse en $O(\frac{1}{n_m})$ (voir figure 5.1).

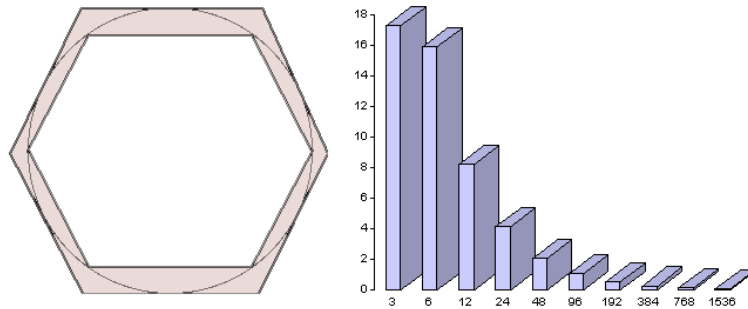


FIG. 5.1 – (gauche) Polygones réguliers interne et externe d'un cercle pour $n_m = 6$, (droite) pourcentage d'erreur de l'estimation du périmètre avec le polygone régulier interne pour n croissant.

Pour revenir au problème d'estimation du nombre π , cette technique a permis d'obtenir les bornes suivantes pour $n_m = 96$:

$$223/71 < \pi < 220/70$$

En plus de la preuve formelle de la convergence d'un estimateur, nous souhaitons avoir une analyse expérimentale de comportement d'algorithme implémentant l'estimateur. Pour cela, nous devons considérer une famille d'objets multi-grilles la plus hétérogène possible. Bien évidemment, il faut que nous puissions calculer la mesure euclidienne $\mathcal{F}(\gamma)$ sur chacun des objets proposés.

Par exemple, la figure 5.2 présente les différents objets multi-grilles pris en compte pour l'estimation de longueur de courbes discrètes (voir paragraphe 5.4.1). Ces objets ont été choisis car ils illustrent beaucoup de cas concrets : courbes polygonales convexes, courbes euclidiennes non-convexes etc. . . . La discrétisation de ces formes sur une grille de taille variable ainsi que la prise en compte de rotations de ces objets sur la grille nous permettent d'avoir un jeu de tests assez complet.



FIG. 5.2 –: *Objets multi-grilles proposés dans KLETTE ET AL. (1999) pour l'analyse d'estimateurs de longueurs.*

Dans le cas tridimensionnel, d'autres objets sont considérés comme nous le verrons dans le paragraphe 5.4.3.

Dans ce qui suit, nous proposons plusieurs estimateurs en 2D et 3D. Nous essayons, autant que possible, de vérifier formellement la propriété de convergence asymptotique. Dans le cas de mesures ponctuelles comme les normales ou la courbure, on dira qu'un estimateur est convergent asymptotique si il y a convergence de l'estimation en tout point.

5.3 Tangentes et normales

5.3.1 Tangentes et normales discrètes d'un contour 2D

Nous considérons dans ce paragraphe le problème de l'estimation de la tangente et de la normale en un point d'une courbe discrète 2D.

Dans une utilisation directe de ces informations, VIALARD (1996b,a) propose l'utilisation de la tangente en tout point pour le calcul du chemin euclidien associé à une courbe discrète : ce chemin correspond à un lissage réversible du contour discret. Plus précisément, chaque sommet initial du contour discret va engendrer un sommet du chemin euclidien, contenu dans un carré de taille 1 centré sur le sommet discret. Ainsi, VIALARD propose l'utilisation d'une notion de tangente discrète pour le positionnement des sommets du chemin euclidien.

Ce calcul de tangente permet aussi la construction d'autres estimateurs de mesures euclidiennes sur des courbes discrètes. De manière intuitive, si nous dérivons la fonction de variation d'angle des tangentes, nous obtenons un estimateur de courbure et si nous intégrons cette information, nous pouvons définir un estimateur de longueur de la courbe discrète considérée. Nous reviendrons plus tard sur ces estimateurs.

D'un point de vue bibliographique, la définition de tangente discrète a été proposée initialement par TROESCH (1993) mais a ensuite été formalisée et analysée par VIALARD (1996b,a) :

Définition 5.2 (Tangente discrète) *La tangente discrète en un point p d'une*

courbe discrète est le plus long segment de droite discrète qu'il est possible de reconnaître, centré en p .

Remarquons que la définition précédente est valide quel que soit le type de courbe discrète considérée (ensemble de pixels ou approche cellulaire), il suffit d'adapter la définition de droite discrète au type de courbe manipulé.

Nous pouvons définir très simplement la notion de normale discrète en un point d'une courbe :

Définition 5.3 (Normale discrète) *La normale discrète en un point p d'une courbe discrète est le vecteur orthogonal à la tangente discrète en p dont le sens est donné par une orientation de la courbe discrète.*

D'un point de vue algorithmique, le calcul de la tangente en un point se fait grâce, par exemple, à l'algorithme 5 de reconnaissance de droite discrète modifié par VIALARD. L'idée est ici de tester l'appartenance à une droite discrète en ajoutant symétriquement des pixels à gauche et à droite (voir illustration figure 5.3). C'est pour effectuer cet ajout symétrique que VIALARD (1996a) a modifié l'algorithme initial de DEBLED-RENNESON (1995).

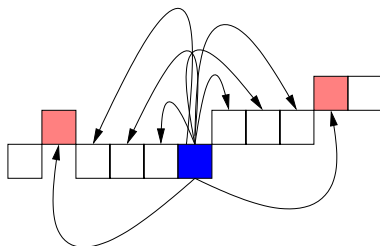


FIG. 5.3 —: Calcul de la tangente discrète par ajout symétrique de pixels et test d'appartenance à une droite discrète. Les pixels colorés terminent la tangente.

Si nous implémentons directement cet algorithme pour le calcul des tangentes discrètes en tout point de la courbe discrète, la complexité obtenue est en $O(n^2)$ où n est le nombre de points discrets de la courbe.

Cependant, FESCHET et TOUGNE (1999) ont proposé un algorithme permettant de réduire le coût en un temps linéaire, ce qui est optimal pour le problème. L'idée principale est d'utiliser les paramètres de la tangente discrète en un point de la courbe pour calculer les paramètres de la tangente des points adjacents.

La figure 5.4 illustre le cœur de cet algorithme : supposons connus les paramètres de la tangente en p et soit q le premier point à droite rejeté lors de la reconnaissance symétrique de segment discret. Calculons ensuite le plus long segment de droite discrète à partir de q vers p , nous notons l le premier point rejeté de ce processus et m le milieu du segment $[lq]$ ¹. Ainsi, FESCHET et TOUGNE (1999) prouvent que, pour tous les pixels compris entre p et m (m exclu), les paramètres des tangentes seront exactement les mêmes que ceux de p .

Ainsi, basé sur cette propriété, ils proposent un algorithme incrémental permettant de mettre à jour les positions des points q , l et m lorsque l'on passe du point p au point à sa droite. Les auteurs montrent que cette mise à jour se fait en temps

1. si le segment $[lq]$ contient un nombre pair de pixels, m est choisi tel qu'il soit plus proche de q que de l

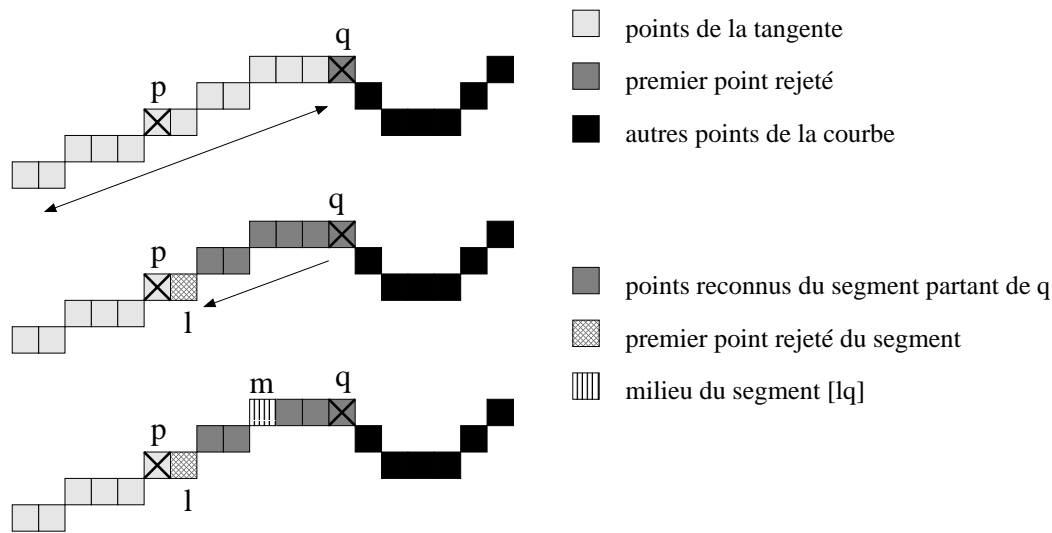


FIG. 5.4 –: Illustration de l’algorithme de FESCHET et TOUGNE (1999) pour le calcul en temps optimal des tangentes discrètes.

constant ce qui permet de calculer toutes les tangentes discrètes en tout point de la courbe en $O(n)$ où n est le nombre de points de cette courbe discrète. Cet algorithme permet une construction des normales en tout point avec la même complexité.

En ce qui concerne la convergence asymptotique, nous avons le théorème suivant :

Théorème 5.1 *Soit Γ la famille des courbes C^2 dans \mathbb{R}^2 à courbure bornée (plus quelques hypothèses techniques). L’estimateur de normales discrètes basé sur la notion de tangente discrète de VIALARD converge asymptotiquement.*

La preuve de ce théorème est donnée dans l’annexe B (théorème B.3).

Pour conclure ce paragraphe, nous avons présenté un calcul de normales d’une courbe discrète qui a la propriété de convergence asymptotique et dont l’algorithme d’estimation est optimal en temps. Nous verrons par la suite que ce résultat est primordial pour de nombreux autres estimateurs.

5.3.2 Normales d’une surface discrète

L’estimation des vecteurs normaux associés à une surface est un problème très important en géométrie discrète. Une des applications privilégiées est la visualisation de volumes binaires. L’idée est de représenter la surface d’un objet discret sous forme lisse en utilisant des techniques d’ombrage se basant sur les normales (ombrage de GOURAUD ou de PHONG par exemple).

Par la suite, nous verrons comment utiliser ces normales surfaciques pour le calcul de nombreux autres estimateurs comme par exemple le calcul de l’aire d’une surface discrète ou le calcul de courbures surfaciques (Gaussienne et Moyenne).

Dans ce qui suit, nous allons présenter les différentes approches de calcul de normales existantes dans la littérature, ainsi que notre contribution à ce problème.

Pour présenter l’état de l’art du calcul de normales, nous reformulons la classification proposée par YAGEL ET AL. (1992). Ainsi, nous pouvons distinguer deux grandes classes d’algorithmes :

Approches volumiques : ces méthodes utilisent des informations volumiques

comme la transformée en distance ou encore la configuration des voxels de l'objet dans un certain voisinage.

Approches surfaciques : dans ce cas, seuls des traitements sur les voxels de la surface sont considérés. Sur le plan algorithmique, ces approches ont l'avantage de manipuler une surface discrète plutôt qu'un volume.

Dans cette bibliographie, nous ne détaillerons pas les approches utilisant des informations de niveaux de gris d'une image non segmentée pour estimer la normale (voir MONGA ET AL. 1994 par exemple).

Quelle que soit l'approche choisie, l'estimation des normales peut se décomposer en deux grandes étapes (YAGEL ET AL., 1992) :

1. Construction des *contextes* : en chaque voxel de la surface discrète, on appelle *contexte* le voisinage autour du voxel qui ne contient pas de discontinuités C^1 .
2. Estimation des normales à partir des contextes : une fois les contextes calculés en chaque voxel, une approximation de la normale est construite à partir des différentes informations contenues dans ce contexte.

Le problème réside principalement dans le calcul de ces contextes. Pour certains algorithmes que nous présentons dans la suite et que nous appelons *locaux*, le contexte en un point est un voisinage de taille fixe. D'autres algorithmes, dits *globaux*, proposent un calcul adaptatif de ce contexte. Enfin, des approches hybrides existent, par exemple un processus adaptatif peut être utilisé pour estimer la normale à partir du contexte de taille fixe.

5.3.2.1 Approches volumiques

Dans ces approches, le volume discret est utilisé pour localiser les différentes discontinuités afin de construire les contextes.

Une première approche que nous présentons a été proposée par THÜRMER et WÜTHRICH (1998). Cet algorithme considère un contexte donné par le parallélépipède centré autour du voxel considéré de taille n_s et un masque de poids w de taille n_s . L'estimation de la normale $(n_x, n_y, n_z)^T$ est donnée par la formule :

$$\begin{aligned} n_x &= \sum_{i,j,k \in \{-n_s \dots n_s\}} \frac{w(i,j,k)\sigma(i,j,k)i}{\sqrt{i^2 + j^2 + k^2}} \\ n_y &= \sum_{i,j,k \in \{-n_s \dots n_s\}} \frac{w(i,j,k)\sigma(i,j,k)j}{\sqrt{i^2 + j^2 + k^2}} \\ n_z &= \sum_{i,j,k \in \{-n_s \dots n_s\}} \frac{w(i,j,k)\sigma(i,j,k)k}{\sqrt{i^2 + j^2 + k^2}} \end{aligned}$$

avec $\sigma(i,j,k) = 1$ si le voxel (i,j,k) est dans l'objet et 0 sinon. Le vecteur $(n_x, n_y, n_z)^T$ est par la suite normalisé.

Cette approche n'utilise pas entièrement le volume de l'objet mais uniquement une épaisseur de celui-ci donnée par n_s . Pour une estimation aussi précise que possible des normales, il faut trouver à la fois la taille de voisinage et le masque des

poinds qui minimisent l'erreur d'approximation sur un ensemble d'objets discrets représentatifs de l'application. Par exemple, THÜRMER et WÜTHRICH (1998) propose une analyse expérimentale basée sur une distribution de plans discrets pour estimer les paramètres de la formule précédente.

Nous présentons une solution un peu particulière qui devrait être classée parmi les approches surfaciques mais qui constitue une base à de nombreuses approches surfaciques. Ainsi, GORDON et REYNOLDS (1985) se placent dans le cas de la visualisation de volume discret et cette approximation utilise la carte des profondeurs (**Z-buffer**) pour estimer la normale des voxels visibles depuis un certain point de vue. Plus précisément, si nous notons $z = f(x, y)$ la surface issue du **Z-buffer**, la normale est donnée par le gradient :

$$\nabla z = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, -1 \right)$$

Ainsi, une fois le **Z-buffer** calculé à partir des voxels à la surface de l'objet, on utilise des calculs de dérivées partielles pour extraire les normales associées aux voxels visibles. Cette approche a cependant un problème majeur lorsqu'il y a des parties concaves dans l'objet qui ne se retrouvent pas dans le **Z-buffer**. Elles peuvent, en effet, engendrer des erreurs de calcul.

Cependant, cette approche peut aussi être vue comme une version simplifiée des calculs basés sur une analyse de la carte de distance. L'idée est la suivante : nous représentons la surface discrète comme une iso-potentielle $f(x, y, z) = 0$ d'une certaine fonction f . Ensuite, si nous sommes capables de construire une telle fonction, la normale en tout point de la surface sera donnée par le gradient de f :

$$\nabla f = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z} \right)$$

Une manière simple de construire cette fonction est de considérer la transformée en distance de la surface discrète (voir illustration en 2D figure 5.5).

Ainsi, une estimation de la normale basée sur ce principe se décompose en trois grandes étapes :

1. calcul de la transformée en distance,
2. application de filtres dérivatifs pour l'estimation des dérivées partielles,
3. construction de la normale sur cette carte de gradients.

Cet algorithme a été implémenté par FLIN ET AL. (2001) avec un processus adaptatif pour construire les normales à partir du gradient de la transformée en distance (noté GTD par la suite). L'idée est de détecter les discontinuités de la surface dans cette carte de distance. Pour cela, si nous considérons un certain voisinage N dans la carte GTD interne à l'objet, la normale à la surface de l'objet est calculée par moyennage des vecteurs dans le voisinage N de la GTD en maintenant trois critères :

- un critère d'angle afin de détecter des discontinuités sur la surface. Ce critère se base sur un écart angulaire, entre la normale estimée à un certain instant et le vecteur de GTD analysé, qui doit être inférieur à 30° (cette valeur correspond aux structures cristallines importantes dans les échantillons de neige, voir paragraphe A.3).

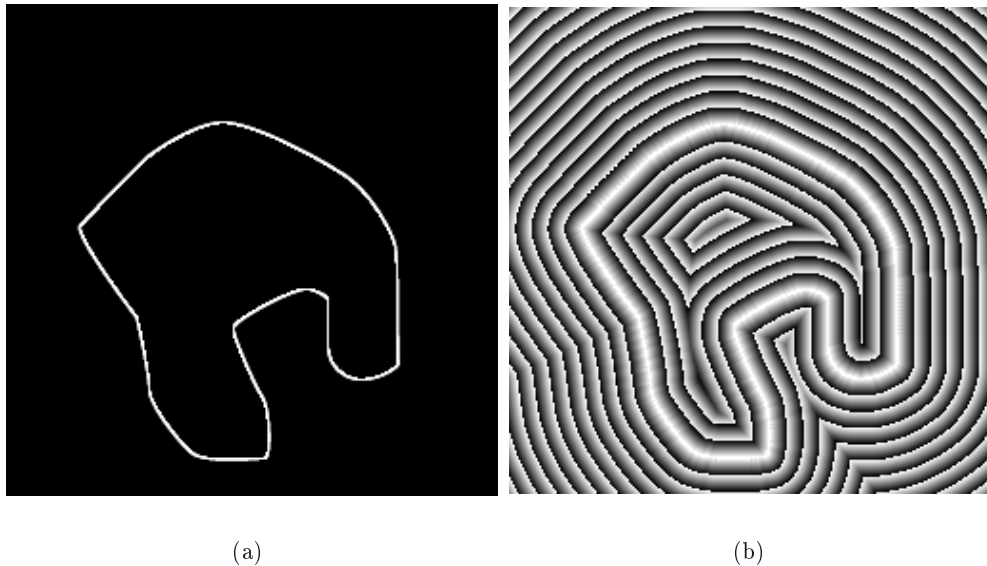


FIG. 5.5 –: *Analyse de la carte de distance pour l'estimation des normales : (a) bord d'un objet discret 2D et (b) utilisation de la transformée en distance euclidienne (algorithme 20).*

- Un critère de symétrie, l'ensemble des vecteurs de la GTD utilisés devant être symétriques autour de la normale estimée.
- Un critère de connexité des points considérés dans le GTD.

Dans COEURJOLLY ET AL. (2002a), nous avons évalué cet estimateur de normale. Une première validation a consisté en un rendu d'objets discrets dont la fonction d'illumination est dirigée par les normales (ombrage de GOURAUD). Ainsi, la figure 5.6 permet de comparer visuellement les rendus entre notre algorithme et l'approche classique par Marching-Cubes (voir paragraphe 2.4.2.4). La figure 5.7 présente quelques résultats de rendu d'un échantillon de neige (voir paragraphe A.3). Nous pouvons remarquer une bonne préservation des différentes discontinuités des objets.

Cet estimateur n'a pas la propriété de convergence asymptotique puisqu'il nécessite une taille de voisinage comme paramètre. Cependant, nous avons évalué expérimentalement la mesure d'erreur. Ainsi, dans la figure 5.8, nous avons estimé les normales sur un ensemble de sphères discrètes de rayons croissants. Pour chaque sphère, nous traçons l'erreur moyenne dans l'estimation des normales en degré ainsi que l'écart type de cette estimation. Ces résultats montrent un très bon comportement de cet estimateur local mais adaptatif basé sur le GTD.

Enfin, nous pouvons proposer une nouvelle approche dans laquelle nous utilisons directement le diagramme de Voronoi discret présenté dans l'algorithme 20. En effet, si nous associons, à tout point de l'objet, le vecteur unitaire vers le site (point du fond) le plus proche, nous obtenons directement une très bonne estimation du GTD sans aucune application de filtres dérivatifs toujours difficiles à paramétrer. La figure 5.9 illustre ce calcul.

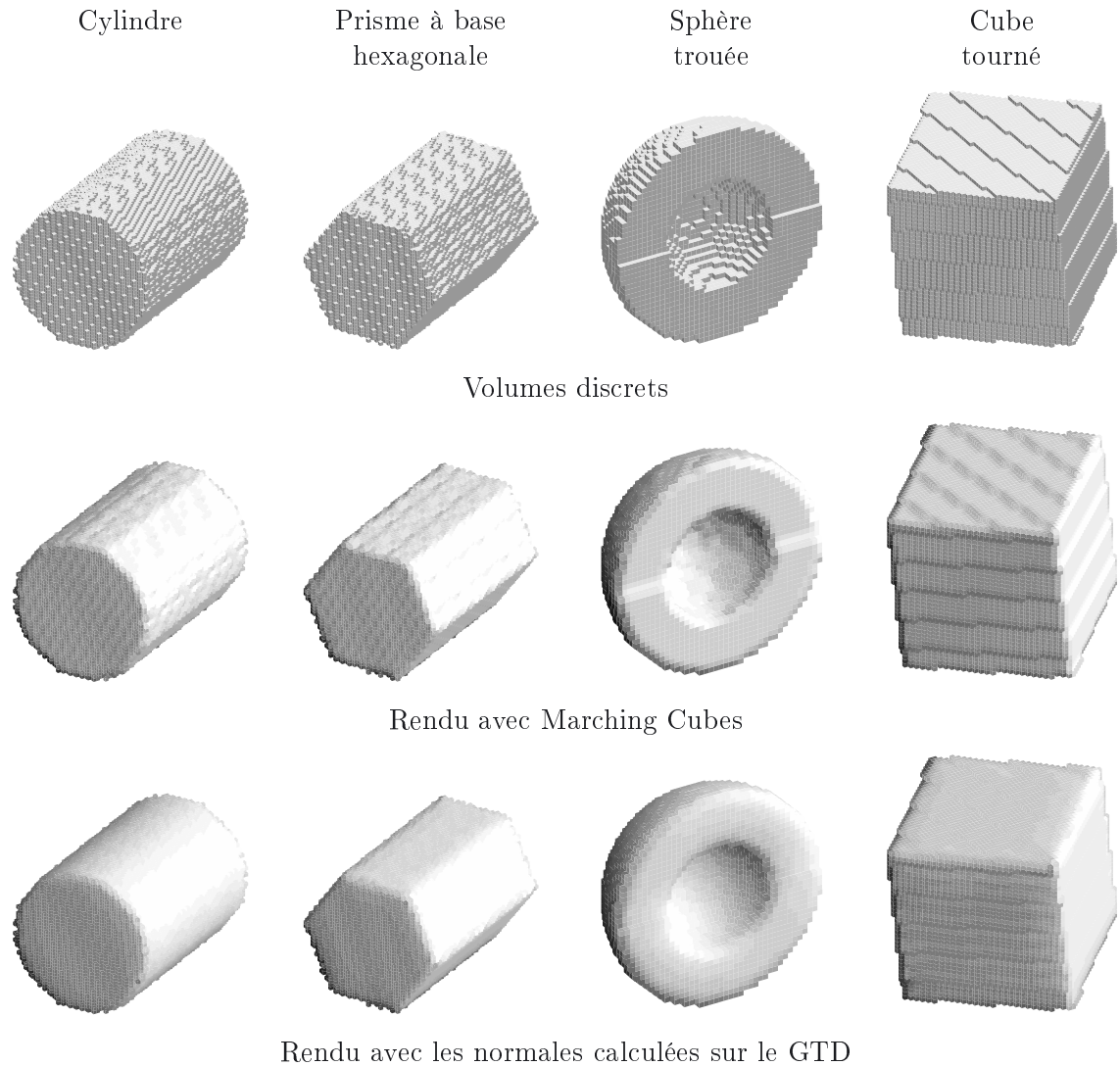


FIG. 5.6 –: *Comparaison entre les visualisations avec l’algorithme des Marching Cubes et avec les normales basées sur le GTD sur des formes géométriques simples.*

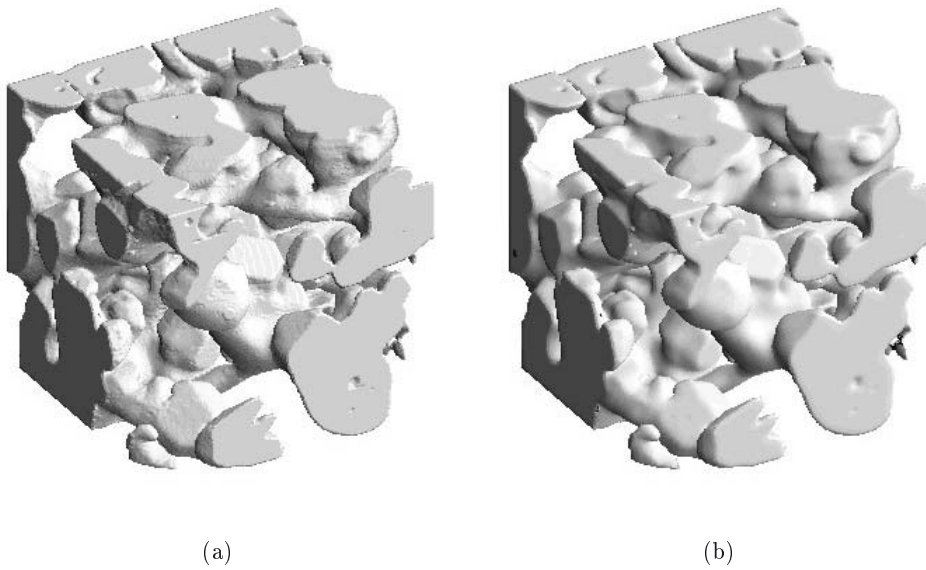


FIG. 5.7 —: Visualisation d'un échantillon de neige avec une estimation de normales basée sur l'algorithme des *Marching-Cubes* (a) et avec l'estimateur basé sur le GTD (b) (Images Météo-France CEN, ESRF, voir les explications dans le paragraphe A.3).

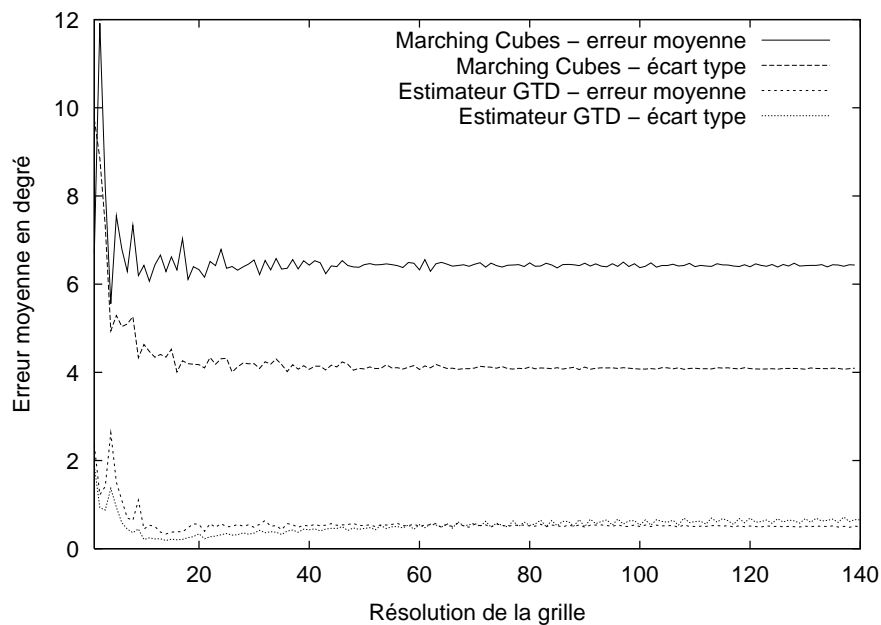
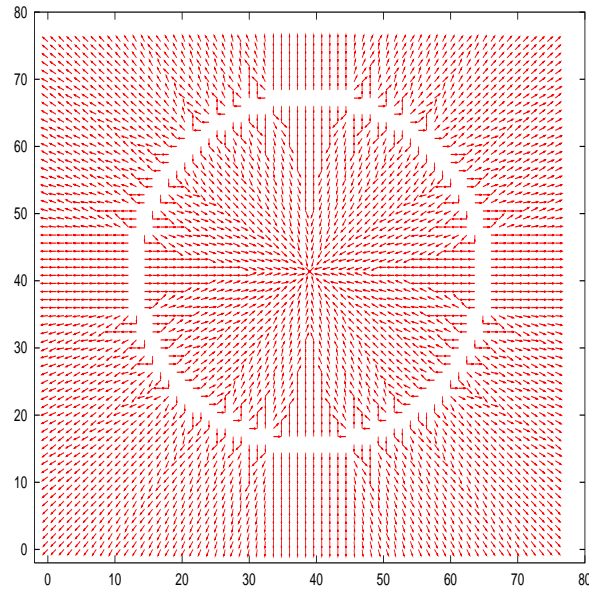
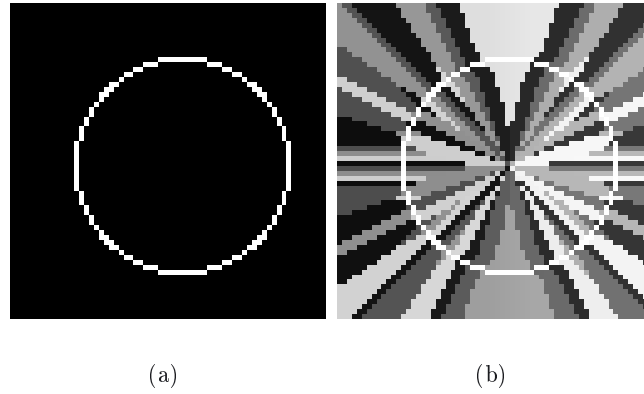


FIG. 5.8 —: Comparaison de l'estimation de la normale entre l'algorithme basé sur le GTD et l'algorithme issu des *Marching Cubes* sur des sphères de taille croissante.



(c)

FIG. 5.9 –: *Illustration 2D de l'utilisation du diagramme de Voronoi discret pour l'estimation directe du GTD : (a) contour d'un objet discret, (b) diagramme de Voronoi discret de ce contour et (c) association à chaque point d'un vecteur unitaire vers le point du contour le plus proche.*

5.3.2.2 Approches surfaciques

Parmi les approches surfaciques, une des premières solutions a été apportée par CHEN ET AL. (1985). Celle-ci se base sur une définition de la surface par complexes cellulaires. Dans ce cas, la surface discrète est composée de surfels munis d'une relation d'adjacence connue (voir paragraphe 1.3.1.2). Ainsi, CHEN ET AL. (1985) estiment la normale en un surfel s en regardant les différentes orientations des surfels adjacents à s (ceux partageant une arête avec s). Ils introduisent donc la notion de *normale élémentaire* qui correspond au vecteur unitaire orthogonal au surfel s , centré sur celui-ci et pointant vers l'extérieur de l'objet (voir figure 5.10). L'estimation de la normale se base donc sur la normale élémentaire en s ainsi que sur les quatre normales élémentaires des surfels adjacents à s . Cette normale estimée correspond à la moyenne de ces normales élémentaires (voir figure 5.11). Étant donné que nous n'avons que six orientations possibles de ces normales élémentaires, le nombre possible de normales estimées est assez faible.

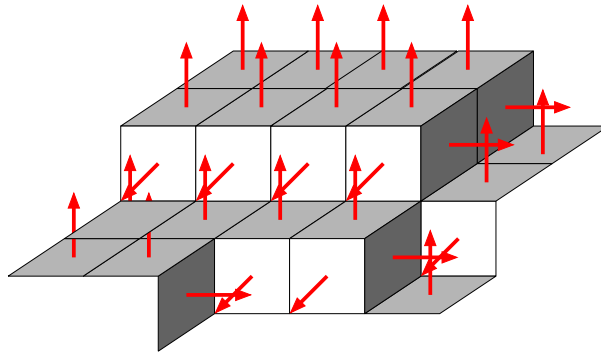


FIG. 5.10 – Illustration des normales élémentaires associées à un morceau de surface discrète.

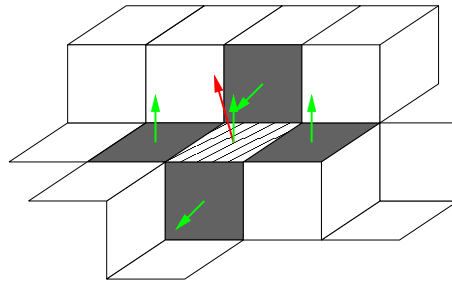


FIG. 5.11 – Approche de CHEN ET AL. (1985) pour l'estimation de la normale basée sur une décomposition cellulaire de la surface discrète.

Par la suite, PAPIER et FRANÇON (1998); PAPIER (1999) proposent l'utilisation d'un voisinage plus grand pour l'estimation de la normale. En se basant sur une définition topologique de ce voisinage (notion d'*ombrelle d'ordre n*), ils proposent un moyennage des normales élémentaires avec des pondérations de manière à favoriser l'apport des normales élémentaires proches de s dans le calcul de la moyenne.

Cette approche permet une bonne estimation des normales mais souffre d'un problème majeur introduit par le choix du voisinage. En effet, celui-ci est purement topologique et ne permet pas de contrôler l'*isotropie* du moyennage. Plus précisément, si nous considérons un plan parallèle à un axe de la grille, le voisinage topologique de PAPIER correspond à des rectangles concentriques (propagation en distance d_8).

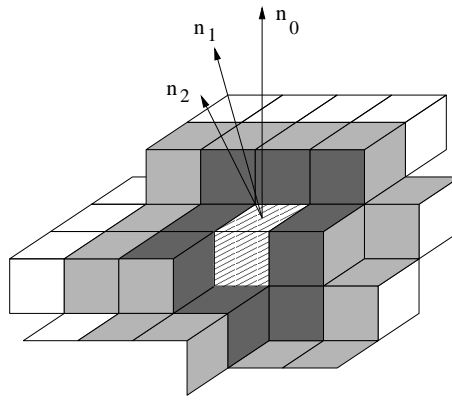


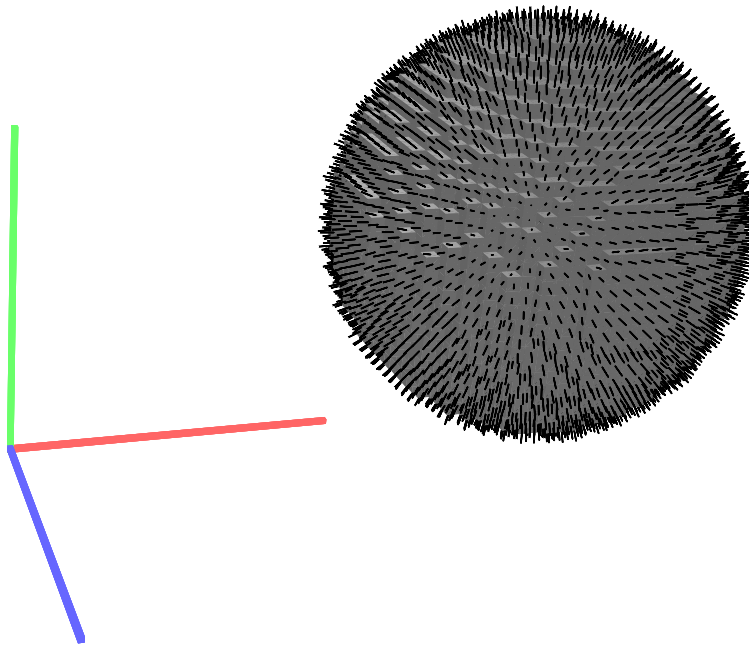
FIG. 5.12 –: *Approche de PAPIER (1999) pour l'estimation de la normale : l'estimation est construite en moyennant les normales élémentaires dans un certain voisinage. Des poids sont utilisés (inversement proportionnels à la distance au surfel s considéré) pour favoriser l'apport des normales élémentaires proches de s dans le calcul de la moyenne.*

Présentons maintenant une nouvelle version de l'algorithme de PAPIER (1999) dont le voisinage se base sur la métrique géodésique décrite dans le paragraphe 4.4.4. L'idée est de considérer tous les surfels qui sont à une distance géodésique inférieure à un certain seuil de s . Cette approche nous permet d'avoir un processus de pondération des normales élémentaires isotropique sur la surface. La figure 5.13 présente les résultats de ce calcul sur une sphère et sur un cube tourné. Les premières mesures d'erreur de cet estimateur indiquent une approximation de la normale un peu meilleure que l'approche de PAPIER (1999) mais de plus amples analyses expérimentales restent à effectuer.

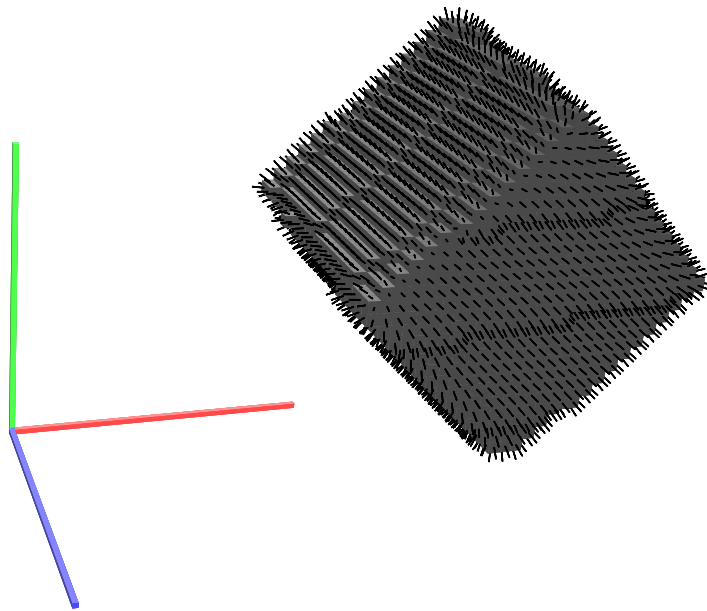
Le problème majeur avec ce genre d'algorithmes reste la construction du contexte qui ne correspond qu'à un voisinage de taille fixe autour du point considéré. Récemment, THÜRMER (2001) a proposé un calcul de contexte basé sur la reconnaissance de plan tangent discret. En fait, THÜRMER propose un algorithme d'estimation des normales en deux étapes : la première consiste en une estimation locale de celles-ci avec l'algorithme de THÜRMER et WÜTHRICH (1998) (voir ci-dessus) et la seconde consiste en un lissage de ces normales dans le contexte défini par le plan tangent discret. Or, comme nous l'avons vu précédemment, la reconnaissance de plan discret ou de plan tangent discret est un problème assez difficile sur le plan algorithmique. Nous voyons ici tout l'intérêt d'une estimation statistique de plan tangent discret comme il a été présenté dans le paragraphe 2.5. En effet, si par exemple nous pouvions stopper le processus de pondération des normales élémentaires par un test statistique d'appartenance à un plan discret, nous pourrions ainsi construire un estimateur très rapide et surtout adaptatif à la forme.

Nous terminons avec les approches basées sur une décomposition en coupes de l'objet discret. Ce type d'estimateur a été proposé initialement par LENOIR ET AL. (1996); LENOIR (1999) puis repris avec les notions de droites discrètes par TELLIER et DEBLED-RENNESON (1999).

Une coupe selon l'axe Oz de la grille d'indice k correspond à l'ensemble des voxels (x, y, k) de l'objet (avec $x, y \in \mathbb{Z}$). De la même manière, nous définissons les coupes selon l'axe Ox et selon l'axe Oy (voir illustration figure 5.14). Si nous considérons une surface discrète composée de surfels et les bords des différentes coupes 2D comme des ensembles de pointels et de lignels, nous avons la propriété importante que chaque



(a)



(b)

FIG. 5.13 —: Exemple de calcul de normales basé sur une pondération de normales élémentaires avec un voisinage géodésique : (a) sur une sphère discrète et (b) sur un cube tourné.

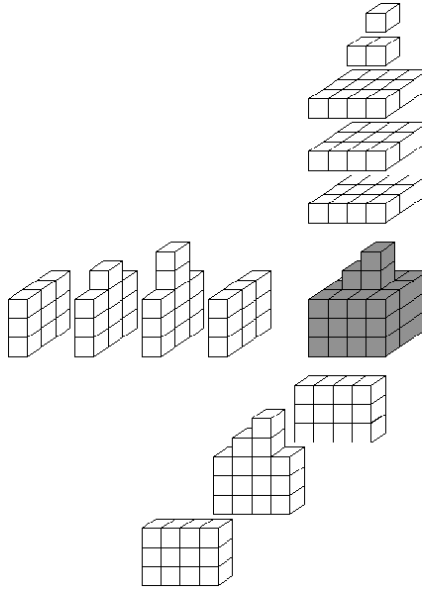


FIG. 5.14 –: *Décomposition en coupes d'un volume selon les axes de la grille.*

surfel engendre deux lignels dans deux coupes différentes (voir figure 5.15). Pour estimer la normale en un surfel s de la surface, LENOIR (1999) propose le calcul suivant : dans un premier temps, nous estimons les tangentes \vec{t}_1 et \vec{t}_2 en dimension 2 dans les deux coupes associées au surfel s . Ensuite, la normale en s est calculée par la formule :

$$\vec{n}(s) = \vec{t}_1(s) \times \vec{t}_2(s)$$

où ' \times ' représente le produit vectoriel usuel.

Ainsi, basé sur ce principe de décomposition du problème en deux problèmes 2D, LENOIR ET AL. (1996) proposent l'utilisation d'un filtre Gaussien récursif pour l'estimation des tangentes 2D et construisent les normales 3D grâce à la formule précédente. Dans une approche basée sur le calcul de tangentes discrètes, TELLIER et DEBLED-RENNESON (1999) proposent un algorithme similaire pour l'estimation de la normale 3D.

5.3.2.3 Estimateur optimal en temps avec convergence asymptotique

Dans ce paragraphe nous proposons une extension de ces approches par coupes en utilisant les résultats précédents sur le calcul de tangentes discrètes 2D et en analysant le comportement asymptotique de l'estimateur.

Tout d'abord, nous pouvons directement reprendre l'algorithme optimal de FESCHET et TOUGNE (1999) pour le calcul des tangentes discrètes 2D. Ainsi, si nous notons n le nombre de surfels composant la surface discrète, le calcul des tangentes discrètes dans toutes les coupes se fait en temps linéaire en le nombre de surfels. Finalement, l'utilisation d'un tel algorithme nous permet l'écriture d'un algorithme d'estimation des normales discrètes 3D en $O(n)$ ce qui est optimal pour le problème.

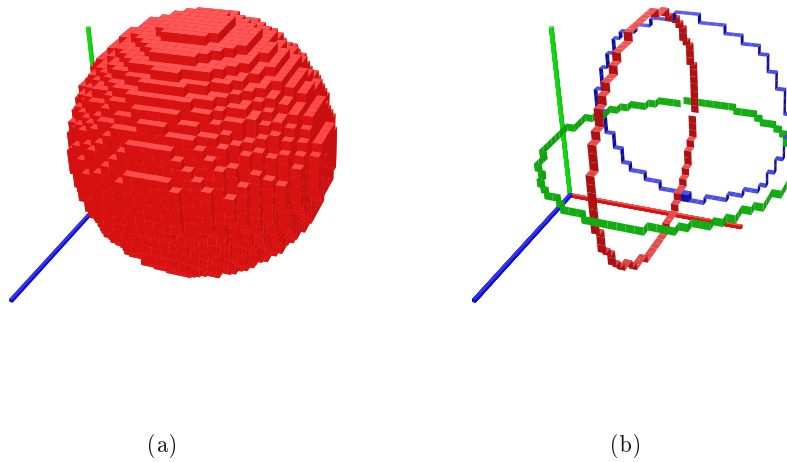


FIG. 5.15 —: Décomposition en coupes d'une surface discrète composée de surfels : (a) surface d'une sphère discrète et (b) exemples de coupes dans les trois axes de la grille.

De plus, nous avons le théorème suivant :

Théorème 5.2 Soit \mathcal{S} une surface C^2 dans \mathbb{R}^3 à courbure bornée (avec quelques hypothèses techniques), l'estimation des normales surfaciques basée sur l'approche par coupes et utilisant les tangentes discrètes converge asymptotiquement.

La preuve de ce théorème est présentée dans l'annexe B (théorème B.5). De manière informelle, si les normales convergent asymptotiquement dans chacune des coupes (voir théorème 5.1), le produit vectoriel aura aussi cette propriété de convergence asymptotique.

En conclusion, l'algorithme présenté ci-dessus est non seulement optimal en temps, mais a aussi la propriété de convergence asymptotique. Une perspective à ce travail consiste en l'implémentation et en l'évaluation de cet estimateur sur un jeu de tests significatifs pour des applications concrètes.

5.4 Longueur et aire

Lorsque l'on associe une *signature* à un contour discret (2D ou 3D), la normalisation des mesures présentes dans cette signature est primordiale avant toute comparaison de tels vecteurs ou plus généralement, avant toute utilisation de cette donnée dans un processus d'apprentissage.

Une normalisation classique des mesures se base sur un calcul préalable de la longueur ou l'aire du contour. Ainsi, si l'estimateur de longueur ou d'aire n'est stable ou n'est pas efficace, c'est toute la validité de la signature qui est remise en question.

5.4.1 Longueur d'une courbe discrète 2D

Dès les premières analyses de contours discrets (FREEMAN 1961 nous semble être un des premiers articles traitant du sujet), le calcul de longueur d'un contour discret est

apparu comme étant un descripteur de forme fondamental et des premières solutions furent proposées. Depuis, de nombreux algorithmes ont vu le jour pour estimer cette mesure euclidienne. Certaines de ces approches se basent sur une estimation de la métrique euclidienne par des approches locales (distances de chanfrein par exemple), d'autres considèrent une polygonalisation de la courbe discrète en utilisant par exemple des algorithmes de segmentation en droites discrètes ou de calcul du polygone interne de longueur minimale.

Nous présentons une évaluation comparative des estimateurs de longueur les plus récents. Cette évaluation porte, à la fois sur une analyse théorique basée sur la notion de convergence asymptotique, et sur une analyse expérimentale. Nous présentons aussi un estimateur de longueur basé sur la notion d'intégration de normales discrètes.

Cette partie a fait l'objet de l'article COEURJOLLY et KLETTE (2002).

5.4.1.1 Approches basées sur des métriques locales

Les estimateurs basés sur des métriques locales furent les premières solutions proposées (voir paragraphe 4.2.1). Rappelons que ces métriques locales définissent tout d'abord un ensemble de déplacements élémentaires et ensuite affectent des poids à ces déplacements. Par exemple, un estimateur simple consiste à affecter un poids de $1/r$ pour tout déplacement horizontal ou vertical sur une r -grille et $\sqrt{2}/r$ pour un déplacement diagonal. Le calcul des poids permettant l'estimation de la longueur se base sur le principe suivant : on considère une distribution probabiliste de segments de droite discrète et on calcule les poids de telle manière que l'erreur d'estimation de la longueur sur la distribution des segments soit minimale (DORST et SMEULDERS, 1987; VERWER, 1991).

Ainsi, en considérant les déplacements diagonaux et les déplacements parallèles aux axes de la grille, DORST et SMEULDERS (1987) définissent l'estimateur BLUE (Best Linear Unbiased Estimator) pour des segments discrets :

$$\mathcal{E}_{BLUE}(D_r(\gamma)) = \frac{1}{r} \cdot (0.948 \cdot n_i + 1.343 \cdot n_d) \quad (5.1)$$

où n_i représente le nombre de déplacements isothétiques (parallèles aux axes de la grille) et n_d le nombre de déplacements diagonaux du contour. Cet estimateur a un intérêt pratique très important du fait de sa simplicité. Cependant, celui-ci n'est pas convergent asymptotiquement dans le cas général mais a une convergence super-linéaire $O(r^{-1.5})$ dans le cas de segments de droite (DORST et SMEULDERS, 1991b). En d'autres termes, il ne converge asymptotiquement que pour la classe des segments de droite discrétisés.

Nous analysons aussi un estimateur proposé par VOSSEPOEL et SMEULDERS (1982), appelé *Corner count*. Celui-ci considère un déplacement élémentaire supplémentaire n_c qui correspond au nombre de changements de codes pair-impair dans le codage de FREEMAN.

$$\mathcal{E}_{CC}(D_r(\gamma)) = \frac{1}{r} \cdot (0.980 \cdot n_i + 1.406 \cdot n_d - 0.091 \cdot n_c)$$

Ce dernier estimateur n'est pas non plus convergent asymptotiquement dans le cas général.

5.4.1.2 Approches basées sur des approximations polygonales

Dans cette catégorie, la courbe discrète est approximée par une courbe polygonale. L'estimation de la longueur de la courbe discrète est ensuite donnée par la longueur de la courbe polygonale.

Dans le chapitre 2, nous nous sommes intéressés aux droites discrètes ainsi qu'aux algorithmes de segmentation d'une courbe discrète en segments discrets. Il est donc normal d'utiliser ces algorithmes pour construire une approximation polygonale.

Nous définissons donc deux estimateurs de longueur basés sur les droites discrètes : le premier, noté \mathcal{E}_{DR-DSS} , correspond à l'algorithme de segmentation de DEBLED-RENNESON et REVEILLÈS (1995) (voir paragraphe 2.2.2.5) sur des courbes de pixels. Le second, noté \mathcal{E}_{VK-DSS} , correspond à l'algorithme de segmentation de KOVALEVSKY (1990) sur une courbe discrète définie selon une approche complexe cellulaire (voir chapitre 1). La figure 5.16 illustre ces différentes approches.

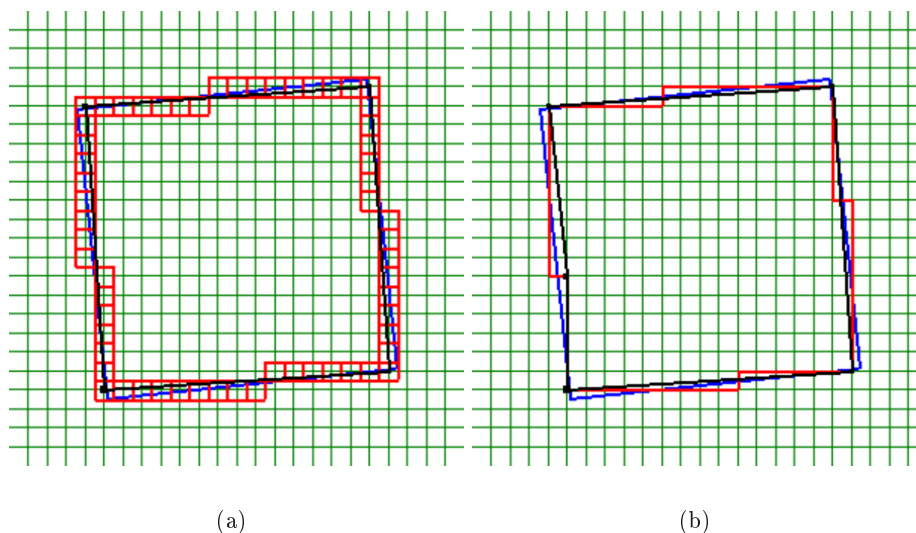


FIG. 5.16 — Illustration des algorithmes de segmentation en droites discrètes : (a) \mathcal{E}_{DR-DSS} et (b) \mathcal{E}_{VK-DSS} sur un carré tourné.

Pour ces approches, nous pouvons formellement prouver la convergence asymptotique :

Théorème 5.3 (KLETTE et ŽUNIĆ (1999)) *Soit P un objet polygonal convexe r -compact de \mathbb{R}^2 , il existe une résolution r_0 telle que pour tout $r \geq r_0$, toute polygonalisation par droite discrète de $D_r(\gamma)$ (avec γ le bord de P) est un polygone dont le périmètre p_r vérifie :*

$$|\mathcal{L}(\gamma) - p_r| \leq \kappa \frac{2\pi}{r}$$

κ étant une constante dépendant de l'algorithme choisi.

Considérant une classe plus large de courbes euclidiennes (non nécessairement convexes), nous proposons un autre théorème :

Théorème 5.4 *Soit Γ la famille des courbes dans \mathbb{R}^2 à courbure bornée (avec d'autres hypothèses techniques), l'estimateur de longueur basé sur une polygonalisation par droite discrète de $\gamma \in \Gamma$ a la propriété de convergence asymptotique.*

La preuve de ce théorème, présentée dans COEURJOLLY ET AL. (2001a), est donnée en annexe B (théorème B.2).

Une autre approximation polygonale, qui a été proposée par SLOBODA ET ZATKO (1996), consiste en un calcul de polygone de longueur minimale (MLP pour *minimum length polygon*) dans une région englobant la courbe réelle γ .

Plus formellement, nous considérons la courbe discrète $J_r^+(\gamma)$ (resp. $J_r^-(\gamma)$) qui correspond au bord de Jordan interne (resp. externe) à γ (similaire à une discrétisation OBQ, resp. BBQ, de γ). Ainsi, l'approximation polygonale de γ consiste en le calcul du polygone de longueur minimale compris dans l'anneau défini par $J_r^+(\gamma)/J_r^-(\gamma)$ (voir illustration figure 5.17). Ce polygone est unique et nous permet de définir l'estimateur de longueur \mathcal{E}_{SZ-MLP} .

Nous utilisons l'algorithme de calcul de MLP proposé par KLETTE ET AL. (1999) dont la complexité est linéaire en la taille de la courbe discrète.

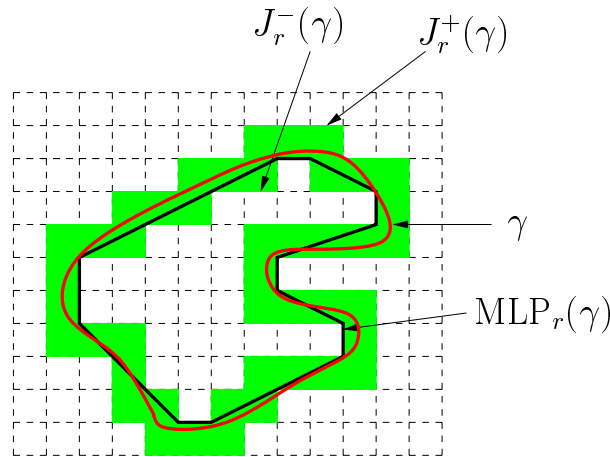


FIG. 5.17 –: Notations pour le calcul de polygone de longueur minimale (MLP)

Cet estimateur converge asymptotiquement :

Théorème 5.5 (SLOBODA ET AL. (1998)) *Soit P un objet polygonal convexe r -compact de \mathbb{R}^2 , il existe une résolution r_0 telle que pour tout $r \geq r_0$, l'estimateur de longueur basé sur SZ-MLP converge asymptotiquement.*

La figure 5.18-(a) illustre cet algorithme sur un carré tourné.

Finalement, nous présentons une dernière approximation basée sur un calcul de MLP mais dans un anneau un peu différent. Dans cette approche proposée par ASANO ET AL. (2001), il est introduit un paramètre réel δ tel que $0 < \delta < \frac{1}{2r}$. Ainsi deux nouvelles courbes de Jordan dans lesquelles chaque sommet v_i peut être déplacé de δ sont définies (voir figure 5.19). Ce processus d'ajustement se fait en deux parcours des sommets de chaque bord. Enfin, un calcul de MLP classique est appliqué dans cet anneau. Notons que cette approche n'est pas entièrement discrète puisque les sommets de la polygonalisation ne sont pas des points de la grille. La figure 5.18-(b) présente un exemple de calcul de cette approximation. Dans ce qui suit, nous notons \mathcal{E}_{AS-MLP} l'estimateur de longueur basé sur cette approximation polygonale.

Cette technique, qui effectue un *lissage* des courbes de Jordan et donc du MLP, est assez similaire à la définition du chemin euclidien proposé par VIALARD (1996a).

Une fois encore, nous avons le théorème suivant :

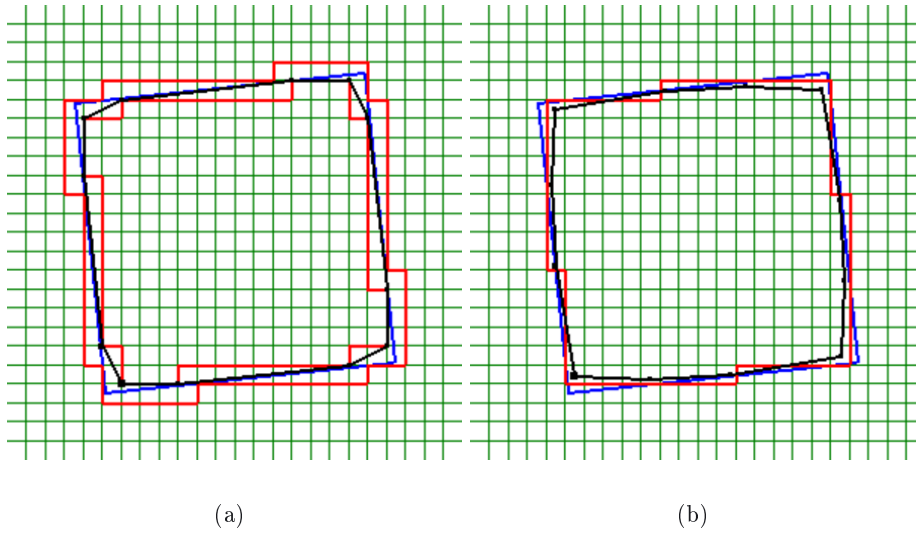


FIG. 5.18 –: Illustration des algorithmes SZ-MLP et AS-MLP : (a) algorithme SZ-MLP avec les courbes de Jordan $J_r^+(\gamma)$ et $J_r^-(\gamma)$, le polygone solution est celui en noir. (b) Approximation basée sur l'algorithme AS-MLP avec $\delta = \frac{1}{4}$. Remarquons que les sommets de la courbe polygonale noire ne sont plus discrets.

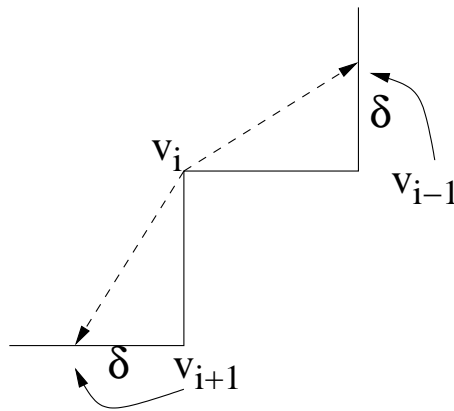


FIG. 5.19 –: Approche AS-MLP : les sommets des bords de Jordan sont déplacés en fonction d'un paramètre δ .

Théorème 5.6 (ASANO ET AL. (2001)) *Soit P un objet polygonal convexe r -compact de \mathbb{R}^2 , il existe une résolution r_0 telle que pour tout $r \geq r_0$, l'estimateur de longueur basé sur AS-MLP converge asymptotiquement.*

Cet algorithme a aussi une complexité linéaire en la taille de la courbe discrète considérée.

5.4.1.3 Approches basées sur l'intégration des normales discrètes

Nous proposons une approche basée sur l'intégration d'un champ de normales discrètes. L'objectif étant de se ramener à une version discrète de la formule classique en géométrie euclidienne :

$$\mathcal{L}(\gamma) = \int_{\gamma} \vec{n}(s) ds \quad (5.2)$$

Cette idée de calcul de longueur par intégration *discrète* est assez ancienne. En effet, elle fut proposée pour la première fois par ELLIS ET AL. (1979) et par LENOIR (1999) pour le calcul de l'aire d'une surface discrète.

L'idée est de remplacer l'intégrale par une somme finie, le vecteur normal par une estimation de celui-ci, notée $\vec{n}^*(s)$, et le terme ds par un produit scalaire avec un vecteur élémentaire \vec{n}_{el} . Dans une représentation cellulaire du contour (séquence de pointels et de lignels), le vecteur élémentaire en un lignel s est un vecteur unitaire orthogonal à s pointant vers l'intérieur de la forme (voir figure 5.20).

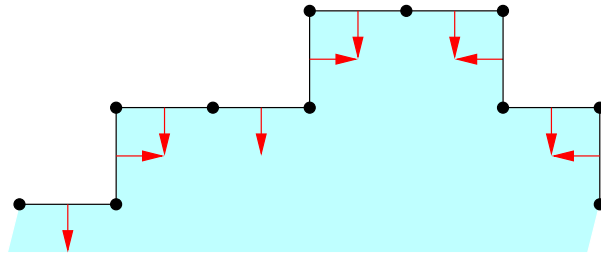


FIG. 5.20 – Illustration des vecteurs élémentaires à une courbe discrète.

Finalement, nous obtenons l'estimateur de longueur donné par la formule :

$$\mathcal{E}_{TAN}(D_r(\gamma)) = \sum_{s \in D_r(\gamma)} \vec{n}^*(s) \cdot \vec{n}_{el}(s)$$

Ce processus de sommation de produits scalaires entre ces deux vecteurs est appelé *processus d'intégration discret*. L'idée intuitive de ce processus est de considérer l'apport de chaque lignel à l'estimation de longueur de la courbe.

Nous avons montré le théorème suivant :

Théorème 5.7 *Soit Γ la famille des courbes C^2 dans \mathbb{R}^2 à courbure bornée, l'estimateur de longueur basée sur le processus d'intégration discret des normales converge asymptotiquement si et seulement si l'estimation des normales a cette propriété de convergence.*

La preuve de ce théorème est présentée dans l'annexe B (corollaire B.4). Dans cette annexe, nous prouvons en fait le problème plus général du calcul d'aire d'une surface discrète (voir paragraphe 5.4.3). Ce dernier problème se ramène, sans difficulté, au problème de l'estimation de longueur d'une courbe discrète.

Or si nous utilisons l'estimateur de normale discrète 2D présenté précédemment, nous obtenons le corollaire suivant :

Corollaire 5.1 *Soit Γ la famille des courbes C^2 à courbure bornée (plus quelques hypothèses techniques), l'estimateur de longueur \mathcal{E}_{TAN} basé sur les normales discrètes du paragraphe 5.3 converge asymptotiquement.*

Preuve : *Il suffit simplement de considérer les théorèmes 5.7 et 5.1. Les hypothèses techniques sont discutées dans l'annexe B.* \square

Dans le cas où la courbe discrète manipulée est donnée sous forme de séquence de pixels connectés, nous pouvons proposer un estimateur de longueur ayant les mêmes propriétés. Dans ce cas, l'estimation de la normale \vec{n}^* est donnée en chaque pixel p et l'apport de ce pixel pour l'estimation de longueur est donnée par :

$$a(p) = \frac{\|\vec{n}^*(p)\|}{\max(|n_x^*(p)|, |n_y^*(p)|)}$$

De manière intuitive, cette mesure quantifie l'écart entre le vecteur normal et l'axe de la grille le plus proche.

Cet apport est donné par la longueur de l'intersection entre la tangente au point p et le voxel considéré (voir figure 5.21). Ainsi, pour ce type de courbe nous avons l'estimateur :

$$\mathcal{E}_{TAN_PIX}(D_r(\gamma)) = \sum_{p \in D_r(\gamma)} a(p)$$

Ce dernier a exactement les mêmes propriétés de convergence asymptotique que \mathcal{E}_{TAN} .

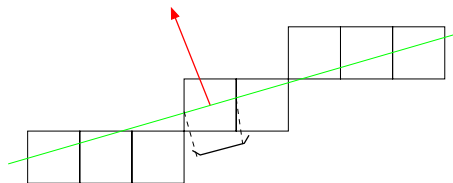


FIG. 5.21 – *Calcul de l'apport d'un pixel dans l'estimation de longueur par intersection entre le pixel et la tangente estimée en celui-ci.*

La figure 5.22 illustre les deux estimateurs précédents.

En utilisant l'algorithme optimal en temps de FESCHET et TOUGNE (1999), les algorithmes implémentant les estimateurs \mathcal{E}_{TAN} et \mathcal{E}_{TAN_PIX} ont une complexité linéaire en la taille de la courbe discrète.

5.4.1.4 Évaluation comparative

Nous présentons maintenant une évaluation comparative des estimateurs de longueurs présentés. Cette évaluation s'effectue, sur le plan théorique avec notamment la propriété de convergence asymptotique, et sur le plan expérimental avec des analyses comportementales des différentes méthodes sur des données réelles.

D'un point de vue théorique, nous évaluons les estimateurs sur les propriétés suivantes :

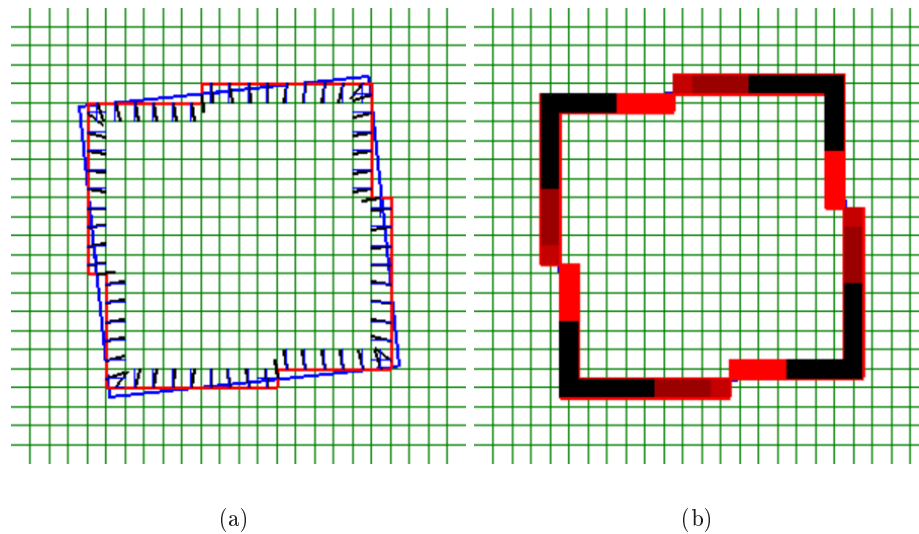


FIG. 5.22 –: *Illustration des estimateurs de longueur basés sur l'intégration des normales discrètes : (a) estimateur \mathcal{E}_{TAN} , à chaque ligne est associé deux vecteurs, l'estimation de la normale et la normale élémentaire ; et (b) estimateur \mathcal{E}_{TAN_PIX} , dans ce cas, l'intensité associée à chaque pixel correspond à la mesure $a(p)$ (le dégradé du clair au sombre correspond aux valeurs croissantes de $a(p)$).*

Convergence asymptotique : est-ce que la mesure donnée par l'estimateur converge asymptotiquement vers la mesure euclidienne de l'objet continu sous-jacent à la discrétisation ?

Unicité : étant donné une courbe discrète, est-ce que l'estimation est unique ?

Calcul discret : est-ce que l'algorithme implémentant l'estimateur considéré est purement discret ? Plus précisément, est-ce que le calcul peut se faire en nombres entiers ?

Extension en 3D : est-il possible d'étendre l'estimateur aux courbes discrètes tridimensionnelles ?

Cette dernière propriété sera reprise dans le paragraphe suivant.

Le tableau 5.1 illustre le comportement de tous les estimateurs présentés vis-à-vis des propriétés ci-dessus. Rappelons cependant que tous les algorithmes ont une complexité linéaire en le nombre de pixels.

Sur le plan expérimental, nous considérons les formes géométriques multi-grilles présentées dans la figure 5.2 et plus précisément leurs discrétisées sur des grilles de plus en plus fines. De plus, pour une résolution donnée, un certain nombre de rotations des formes sont prises en compte. Finalement, nous évaluons tous les estimateurs sur l'ensemble des courbes discrètes obtenues.

Lors de l'évaluation des estimateurs, deux mesures sont analysées. La première consiste en un calcul d'erreur relative en pourcentage par rapport à la longueur euclidienne exacte des formes. Cette mesure nous permet d'évaluer expérimentalement la propriété de convergence asymptotique. La seconde mesure utilisée n'est définie que sur les approches polygonales. Celle-ci est donnée par le produit entre l'erreur relative de l'estimateur et le nombre d'arêtes de l'approximation polygonale. Ce calcul proposé par KLETTE ET AL. (1999) permet d'évaluer l'efficacité de l'approche polygonale pour l'estimation de longueur.

Approche	Convergence asymptotique	Unicité	Discret	Extension 3D	Références principales
Métriques locales	non	oui	cela dépend	BECKERS et SMEULDERS (1992); JONAS et KIRYATI (1998)	DORST et SMEULDERS (1987); VERWER (1991)
DR-DSS et VK-DSS	oui	non	oui	COEURJOLLY ET AL. (2001a)	DEBLED-RENNESON et REVEILLÈS (1995); KOVALEVSKY (1990)
SZ-MLP	oui	oui	oui	KLETTE et BÜLOW (2000)	SLOBODA et ZATKO (1996); KLETTE ET AL. (1999)
AS-MLP	oui	oui pour δ fixé	non	-	ASANO ET AL. (2001)
TAN et TAN_PIX	oui	oui	oui	-	ELLIS ET AL. (1979); COEURJOLLY et KLETTE (2002)

TABLE 5.1 –: *Évaluation théorique des différents estimateurs et principales références bibliographiques.*

Dans notre analyse, nous avons considéré un processus de discrétisation des formes de la figure 5.2 sur une grille de résolution comprise entre 30 et 1000 pixels. De plus pour chaque résolution, les rotations d'angles compris entre 0 et 90 degrés par pas de 1 sont considérées. Finalement, les mesures associées à un estimateur correspondent à des moyennes des mesures obtenues sur toutes les formes et pour toutes les rotations de celles-ci. De plus, un moyennage sur une fenêtre de taille 16 permet un lissage des résultats et donc une meilleure lisibilité des courbes.

La figure 5.23 présente l'analyse de convergence asymptotique expérimentale basée sur l'erreur relative. Comme nous l'avons vu dans l'analyse théorique, toutes les mesures d'erreur des estimateurs, sauf pour BLUE et CC, convergent vers 0 pour une résolution croissante. Les mesures d'erreur des estimateurs basés sur des métriques locales (BLUE et CC) semblent se stabiliser vers une erreur de 2%. En détaillant un peu plus les résultats, nous pouvons observer que les meilleurs résultats sont obtenus par les algorithmes de segmentation en droites discrètes, par l'algorithme AS-MLP ainsi que par l'algorithme TAN. Ensuite nous avons l'approche SZ-MLP et enfin les estimateurs locaux.

La figure 5.24 présente la mesure d'efficacité de l'approximation polygonale pour les algorithmes VK-DSS, DR-DSS, SZ-MLP et AS-MLP. Une valeur faible de cette mesure indique un bon compromis entre l'estimation de longueur et la qualité, en nombre d'arêtes, de l'approximation polygonale. Les meilleurs résultats sont obtenus par les approches basées sur les droites discrètes.

Afin de mieux analyser le caractère anisotrope des estimateurs, nous avons considéré l'estimation de longueur de différentes rotations d'un carré sur une grille de taille fixe (voir figure 5.25). Comme nous pouvions nous y attendre, les estimateurs basés sur des métriques locales ont de très mauvais comportements pour ce type de tests. En effet, certaines orientations de la grille sont privilégiées (45° pour BLUE et 0° et 90° pour CC).

Enfin, la figure 5.26 montre les différents temps de calcul des estimateurs. Nous pouvons alors remarquer que la différence entre les estimateurs locaux et les algorithmes utilisant des outils de la géométrie discrète comme les droites discrètes est faible.

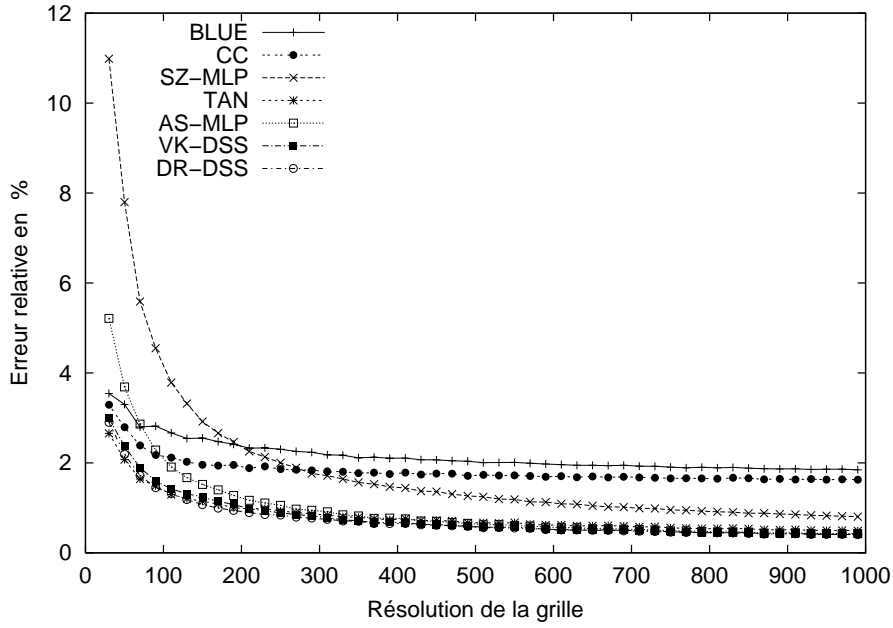
En conclusion, les estimateurs basés sur des métriques locales ne se justifient, ni sur le plan théorique, ni sur le plan expérimental et même leur rapidité peut être discutable dans des cas pratiques.

5.4.1.5 Remarques sur la plate-forme de tests

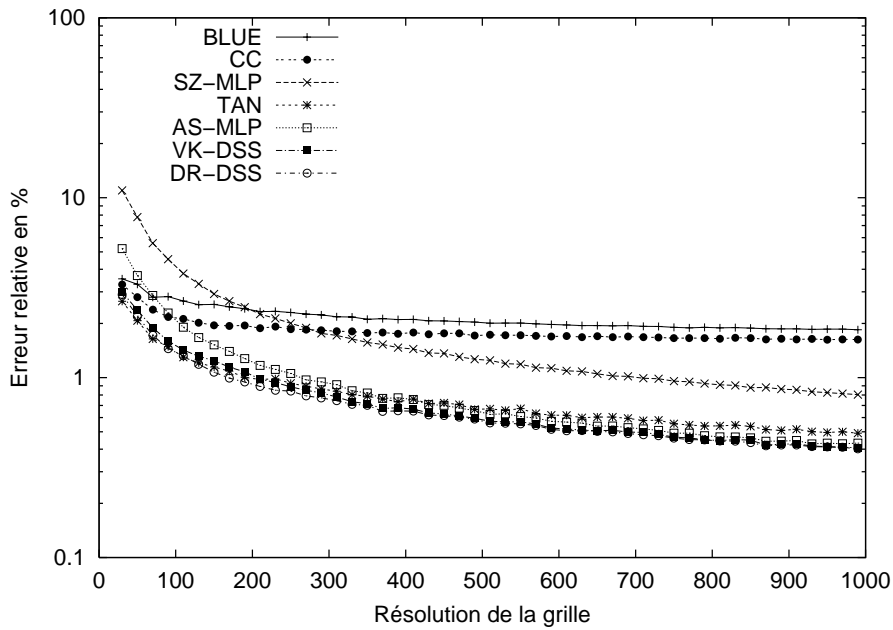
Toutes les analyses expérimentales présentées dans le paragraphe précédent ont été obtenues sur une plate-forme de tests implémentant tous les algorithmes sous une structure de classes C++.

Cette plate-forme contient :

- une série de classes implémentant les différentes formes multi-grilles de la figure 5.2 en fonction de l'orientation de la grille et la résolution de cette dernière ;
- une série de classes implémentant les différents algorithmes ;
- un système d'évaluation d'un estimateur pour une courbe donnée basé sur la lecture de scripts, ce qui permet d'effectuer les tests de convergence asymptotique très facilement.



(a)



(b)

FIG. 5.23 — Convergence asymptotique expérimentale des estimateurs de longueur : (a) erreur relative en pourcent de chaque estimateur sur le jeu de test et (b) mêmes résultats mais sur une échelle logarithmique en y de manière à faciliter la lecture des courbes pour les valeurs proches de 0.

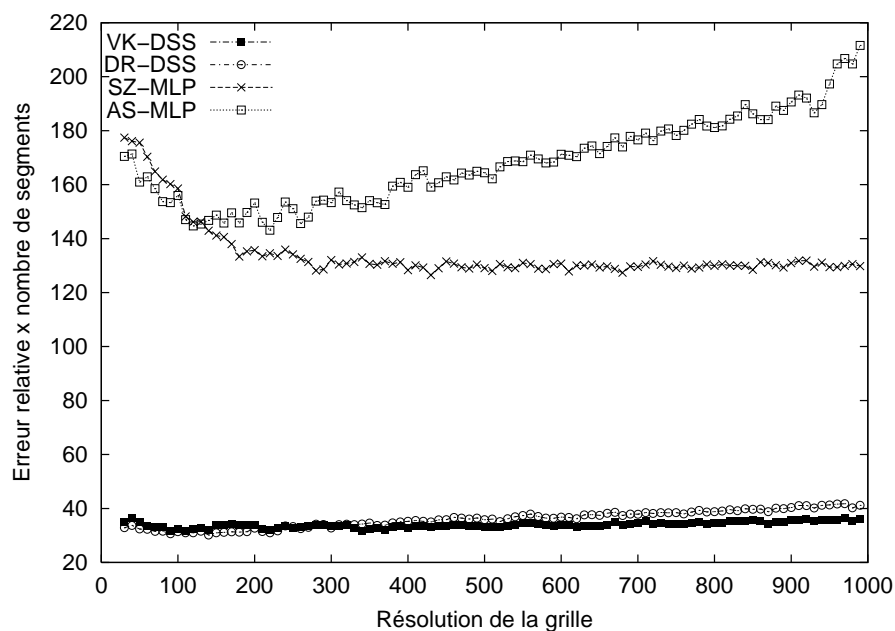


FIG. 5.24 — Mesure d'efficacité de l'approximation polygonale pour l'estimation de longueur : en abscisse se trouve la résolution de la grille et en ordonnée le produit entre le nombre d'arêtes de l'approximation polygonale et l'erreur relative en pourcent.

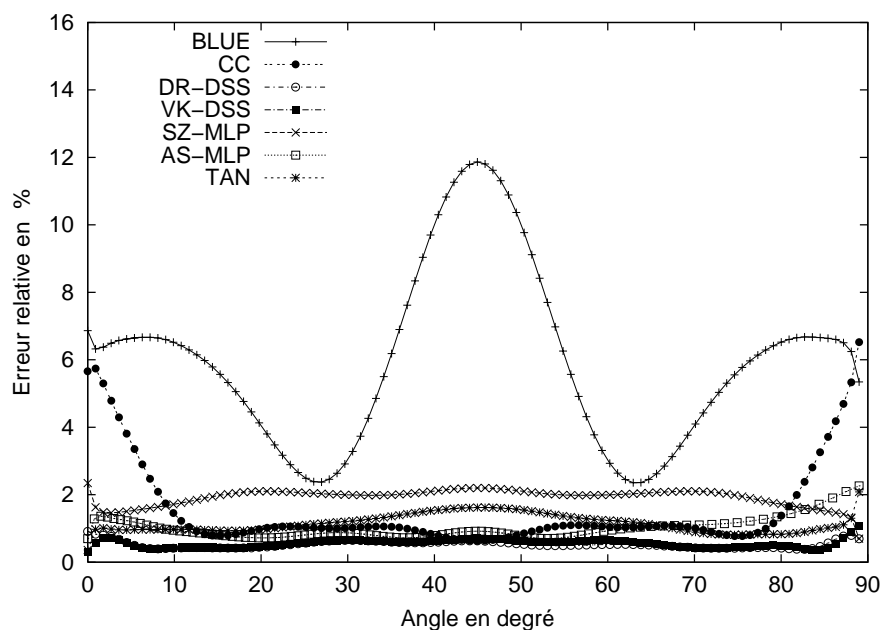


FIG. 5.25 — Erreur relative sur un carré de taille fixe sur lequel une rotation est effectuée : en abscisse se trouve l'angle de rotation entre 0 et 90 degrés et en ordonnée, l'erreur relative des différents estimateurs.

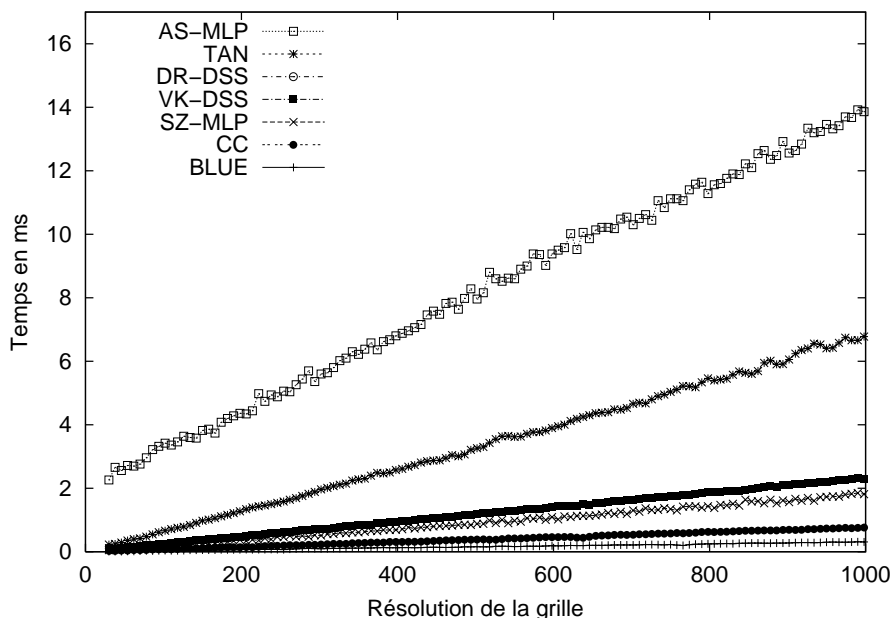


FIG. 5.26 — Temps de calcul des différents estimateurs sur une station Sparc Ultra 10.

Toutes ces classes sont disponibles de manière libre sur le site *Digital Geometry and Topology*² maintenu par R. KLETTE.

Ce projet a démarré en 1999 à l’initiative de BEN YIP et REINHARD KLETTE du laboratoire CITR de l’université de Nouvelle-Zélande avec la création de la structure globale et l’implémentation des algorithmes SZ-MLP et VK-DSS. Par la suite, TETSUO ASANO and YASUYUKI KAWAMURA du laboratoire JAIST au Japon ont ajouté l’algorithme AS-MLP.

Enfin, nous avons implémenté les algorithmes BLUE, CC, DR-DSS, TAN et TAN_PIX, ainsi que les scripts permettant une analyse multi-grille efficace.

5.4.2 Longueur d’une courbe discrète 3D

Nous nous intéressons maintenant à l’estimation de la longueur de courbes discrètes tridimensionnelles. Dans ce cas, nous considérons une courbe $\gamma : [0, 1] \rightarrow \mathbb{R}^3$ et un processus de discrétisation multi-grille $D_r(\gamma)$ adapté.

Comme nous l’avons présenté dans le tableau 5.1, des extensions des algorithmes précédents existent pour l’estimation de longueur de courbes discrètes 3D. Ainsi, une première approche très simple consiste en une généralisation 3D des estimateurs basés sur des métriques locales. Dans ce cas, les déplacements élémentaires considérés sont par exemple les 6-adjacences de voxels (notées n_a), les adjacences de voxels par lignes (notées n_b) et les adjacences de voxels par pointels (notées n_c). Les poids associés à ces déplacements sont calculés de manière à minimiser l’erreur de l’estimation de longueur pour une distribution aléatoire de segments de droites réelles. Par exemple, BECKERS et SMEULDERS (1992) considèrent l’estimateur :

$$\mathcal{E}_{Beckers}(D_r(\gamma)) = 0.8875n_a + 1.34224n_b + 1.59772n_c$$

2. <http://http://www.citr.auckland.ac.nz/dgt>, un miroir est disponible à l’adresse <http://eric.univ-lyon2.fr/~dcoeurjo/download.html>

Par la suite, JONAS et KIRYATI (1998) proposent des estimateurs basés sur d'autres déplacements élémentaires.

En considérant des courbes discrètes 3D 6-connexes (*cube-curve*), KLETTE et BÜLOW (2000) proposent une approximation polygonale basée sur le polygone de longueur minimale inscrit dans ce volume (voir figure 5.27). Ce polygone 3D se calcule en deux étapes :

- dans un premier temps, des sommets *candidats* sont initialisés sur des arêtes dites *critiques* (lignels appartenant à trois voxels adjacents) ;
- des optimisations locales (suppression de sommets, déplacement le long des arêtes...) sont effectuées afin de réduire la longueur du polygone.

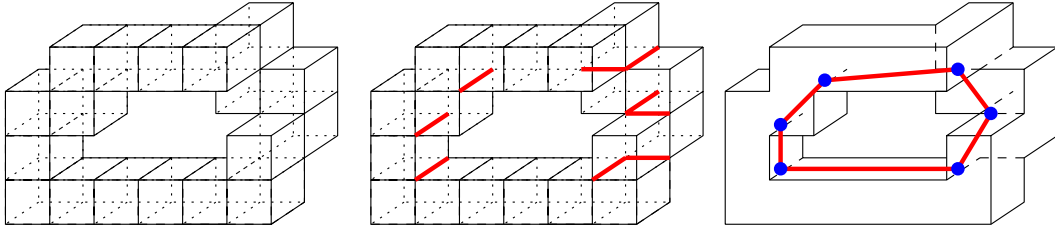


FIG. 5.27 — Illustration du calcul du polygone de longueur minimale associé à une courbe discrète 6-connexe : courbe discrète initiale, sélection des arêtes critiques et calcul du polygone de longueur minimale.

Cet estimateur, noté \mathcal{E}_{3D-MLP} permet l'estimation de longueur d'une courbe 3D mais de nombreux problèmes existent sur le plan algorithmique. En effet, la phase d'optimisation de l'algorithme (seconde étape) est appliquée jusqu'à ce qu'aucune modification ne réduise la longueur du polygone MLP. Or, étant donné que les sommets de ce polygone peuvent se déplacer sur les arêtes critiques, non seulement ce calcul n'est plus discret mais en plus, l'algorithme permettant de trouver le minimum est exponentiel (l'algorithme est NP-complet) en temps. Ainsi, KLETTE et BÜLOW (2000) ont introduit un τ -MLP qui correspond au polygone dont la distance diffère de l'optimal de τ . Enfin, les auteurs proposent un algorithme non-linéaire pour calculer un tel polygone.

Sur le plan de la convergence asymptotique, les problèmes de définition et de construction du polygone MLP en 3D ne permettent pas d'avoir de preuve de convergence.

Dans COEURJOLLY ET AL. (2001a), nous avons proposé un algorithme de calcul de longueur basé sur la segmentation de la courbe en droites discrètes 3D (voir algorithme 10). Ainsi, cet estimateur se base sur l'approximation polygonale en droites discrètes pour estimer la longueur de la courbe discrète (voir figure 2.25 pour quelques exemples de segmentation). Rappelons que cet algorithme est totalement discret et que le calcul de la segmentation s'effectue en $O(n)$ où n est le nombre de voxels de la courbe discrète, ce qui est optimal pour le problème.

D'un point de vue théorique, nous avons le théorème suivant :

Théorème 5.8 *Soit Γ la famille des courbes dans \mathbb{R}^3 à courbure bornée (avec d'autres hypothèses techniques), l'estimateur de longueur basé sur une polygonalisation en droites discrètes 3D de la discrétisation de $\gamma \in \Gamma$ a la propriété de convergence asymptotique.*

Dans la preuve donnée dans l'annexe B (théorème B.2), nous avons utilisé un schéma général de preuve basé sur le fait qu'un segment de droite discrète faisant parti de la segmentation de la courbe $D_r(\gamma)$ peut être vu comme un *tube* dont l'épaisseur est contrôlée et qui englobe un morceau de γ . Ce schéma est valide quelle que soit la dimension du problème et c'est pour cela que nous pouvons utiliser exactement la même preuve pour le cas 2D que pour le cas 3D.

Finalement, nous avons un estimateur de longueur de courbes discrètes 3D, noté \mathcal{E}_{3D-DSS} , dont le calcul est optimal en temps, purement discret et qui converge asymptotiquement.

D'un point de vue expérimental, il est assez difficile de proposer un jeu de tests très pertinent en 3D. Dans les analyses expérimentales des figures 5.28 et 5.29, nous avons considéré des cercles et des ellipses 3D. Dans la figure 5.29-(b) nous avons tracé une mesure que nous n'avions pas présentée dans le paragraphe précédent, celle-ci est l'inverse de l'erreur relative en pourcentage. Cette mesure nous permet de nous rendre compte de la vitesse de convergence de l'estimation. Dans notre cas, la courbe se stabilise vers une vitesse de convergence linéaire.

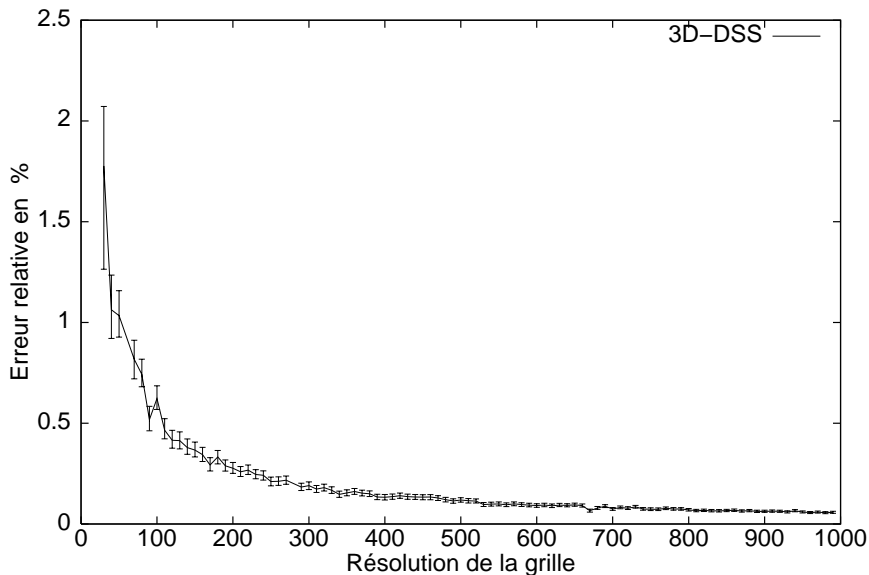
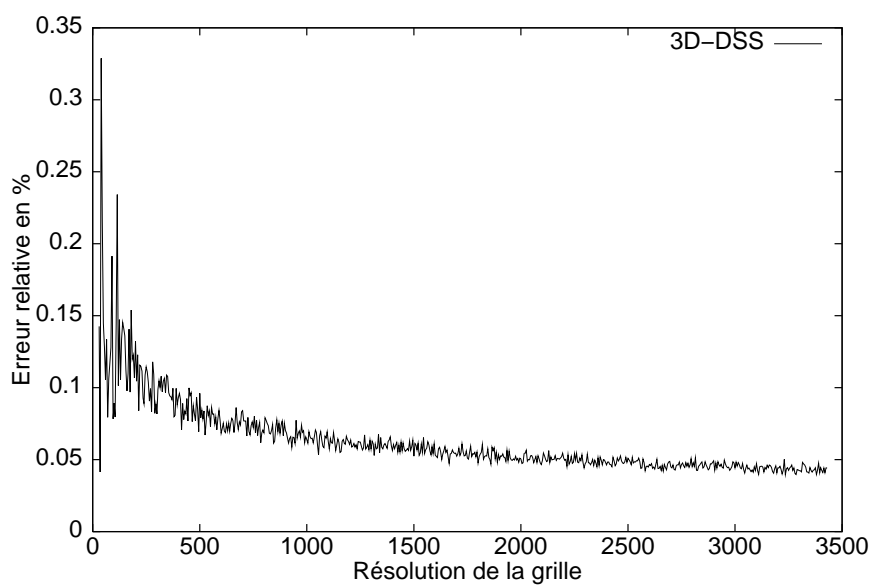
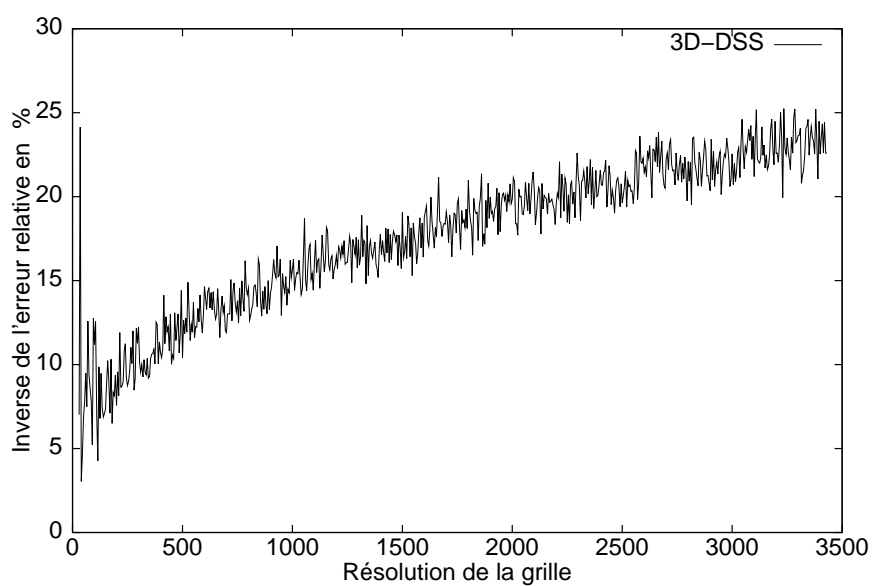


FIG. 5.28 – : *Analyse expérimentale de convergence asymptotique de l'estimateur \mathcal{E}_{3D-DSS} : convergence sur un cercle 3D. Les barres d'erreur correspondent aux différentes mesures observées en fonction du point de départ considéré de la segmentation.*

Une perspective importante de ce travail sur les longueurs de courbes discrètes 3D est bien évidemment la mise en place d'une plate-forme de test similaire à celle utilisée dans le cas 2D. Un travail préalable serait la désignation d'une famille de courbes réelles 3D sur lesquelles nous avons la mesure exacte de longueur et que nous discrétiserons en multi-grille.



(a)



(b)

FIG. 5.29 – : *Analyse expérimentale de convergence asymptotique de l'estimateur \mathcal{E}_{3D-DSS} : (a) sur une ellipse 3D dont le rapport entre les longueurs d'axe est $\frac{1}{\sqrt{2}}$ et (b) inverse de l'erreur relative sur les ellipses 3D afin d'observer la vitesse de convergence.*

5.4.3 Aire d'une surface discrète

En dimension 3, le calcul de l'aire de la surface d'un objet discret est un descripteur de forme très important. En effet, de la même manière qu'en 2D, cette mesure permet une normalisation de toutes les autres caractéristiques que nous pourrions extraire de ces surfaces. D'un point de vue pratique, cette mesure nous permet d'extraire une information physique très importante dans le cas d'analyses d'échantillons de neige (voir paragraphe A.3).

Nous allons, ici aussi, nous intéresser à la notion de convergence asymptotique. Pour cela, nous reformulons la propriété de convergence asymptotique dans le cas des surfaces dans \mathbb{R}^3 . Nous notons \mathcal{S} une surface de JORDAN dans \mathbb{R}^3 . Nous considérons un processus de discrétisation, noté D_r sur une grille 3D de résolution r . Soit $\mathcal{A}(\mathcal{S})$ l'aire de la surface \mathcal{S} et $\mathcal{E}(D_r(\mathcal{S}))$ l'estimation obtenue à l'aide de \mathcal{E} . L'estimateur \mathcal{E} converge asymptotiquement si et seulement si :

$$|\mathcal{E}(D_r(\mathcal{S})) - \mathcal{A}(\mathcal{S})| \leq \epsilon(r) \quad \text{avec} \quad \lim_{r \rightarrow \infty} \epsilon(r) = 0$$

5.4.3.1 Solutions existantes

Dans un premier temps, MULLIKIN et VERBEEK (1993) ont proposé une approche locale pour l'estimation d'aire de surfaces discrètes. L'idée est la suivante : tout d'abord, les différentes configurations locales possibles de voxels, appartenant à une surface discrète, sont classifiées en neuf catégories. À chacune de ces classes, un poids est associé avec un processus statistique de manière à estimer au mieux l'aire de morceaux de plans discrets distribués aléatoirement. Lorsque l'on analyse une surface quelconque, nous comptons, pour chaque classe, le nombre d'occurrences du motif. Finalement, l'aire est estimée par combinaison des occurrences et des poids de chaque catégorie.

Dans une approche basée sur une approximation polyédrale de la surface discrète, deux solutions naïves peuvent être proposées (voir paragraphe 2.4.2.4) : la première consiste en le calcul de l'enveloppe convexe de l'objet discret pour ensuite estimer l'aire de cette enveloppe, la seconde se base sur une utilisation de l'algorithme des Marching-Cubes. Pour la première solution basée sur une enveloppe convexe de l'objet, l'estimateur n'est bien évidemment valide que pour des objets convexes et une preuve de convergence existe dans ce cas. Nous reviendrons sur cette dernière plus tard (SLOBODA et ZATKO, 2001). En considérant l'algorithme des Marching-Cubes, l'estimation de l'aire de la surface surface consiste à sommer l'aire de chaque facette. Cet estimateur n'a pas la propriété de convergence asymptotique compte tenu du fait qu'il s'agisse d'un calcul local. Afin d'augmenter encore la précision d'un tel estimateur dans des cas pratiques, LINDBLAD et NYSTRÖM (02) proposent une pondération des différentes configurations des facettes du Marching-Cubes. Cette dernière solution est proche de l'estimateur de MULLIKIN et VERBEEK (1993) décrit ci-dessus.

KLETTE et SUN (2001) utilisent une facettisation en plan discret et se basent sur cette dernière pour estimer l'aire de la surface. L'idée est ici de mettre en place le même type d'estimateurs que \mathcal{E}_{DR-DSS} ou \mathcal{E}_{VK-DSS} dans le cas 2D. Cependant, comme nous l'avons vu dans le paragraphe 2.4.2.4, de nombreux problèmes existent pour la localisation des sommets d'une polyédrisation et la construction des faces. Ainsi, la mesure donnée par cet estimateur est très variable en fonction des stratégies choisies (propagation, distribution de germes...). De plus, le coût en complexité des algorithmes de reconnaissance de plan discret étant assez important, l'estimateur

obtenu n'est pas très efficace sur le plan algorithmique. Finalement, aucune preuve de convergence asymptotique n'est disponible sur ce type de méthode.

Sur le plan théorique, SLOBODA et ZATKO (2001) proposent l'utilisation d'une enveloppe convexe relative comme approximation polyédrale de l'objet discret. Cette structure géométrique serait équivalente au MLP dans le cas 2D. En se basant sur leur définition de l'enveloppe convexe relative, les auteurs prouvent la propriété de convergence asymptotique de l'estimateur d'aire de surface basé sur ce principe. Cependant, aucun algorithme n'est encore disponible pour calculer une telle surface polyédrale. Finalement, YU et KLETTE (2002) proposent un algorithme permettant d'approximer cette enveloppe convexe relative dans une approche par coupes : dans chaque coupe, selon un axe de la grille, un polygone MLP est calculé. Ensuite, une surface polyédrale est construite en se basant sur les polygones MLP des coupes. Cet algorithme ne construit pas exactement l'enveloppe convexe relative de SLOBODA et ZATKO, et n'a pas la propriété de convergence asymptotique.

Comme nous pouvons le voir, les approches polyédrales pour l'estimation de surface n'offrent pas de solutions pertinentes du point de vue de la convergence asymptotique. Dans le paragraphe suivant, nous détaillons un estimateur d'aire de surface, proposé initialement par LENOIR (1999), dont nous prouvons la convergence asymptotique dans le cas général.

5.4.3.2 Estimation d'aire de surface discrète par intégration de normales

Pour faire une comparaison avec le cas bidimensionnel, cette approche correspond aux estimateurs \mathcal{E}_{TAN} et \mathcal{E}_{TAN_PIX} .

Comme nous l'avons présenté précédemment, nous avons la formule classique en géométrie :

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

Nous pouvons proposer, de la même manière qu'en 2D, l'estimateur d'aire de surface suivant :

$$\mathcal{E}_{TAN_SURF}(D_r(\mathcal{S})) = \sum_{s \in D_r(\mathcal{S})} \vec{n}^*(s) \cdot \vec{n}_{el}(s)$$

où $\vec{n}^*(s)$ correspond à une estimation du vecteur normal centré sur l'élément de surface s de la discrétisation de \mathcal{S} , $\vec{n}_{el}(s)$ correspond au vecteur normal élémentaire en s (voir figure 5.10).

Cet estimateur d'aire de surface a été proposé initialement par LENOIR (1999). Dans ce qui suit, nous évaluons ce dernier autant sur le plan théorique qu'expérimental.

Si la surface discrète est définie en tant qu'ensemble de voxels, et si les normales sont estimées en chaque voxel, nous pouvons décrire un estimateur similaire basé sur la mesure suivante, donnée en chaque voxel v :

$$a(v) = \frac{\|\vec{n}^*(v)\|}{\max(|n_x^*(v)|, |n_y^*(v)|, |n_z^*(v)|)}$$

Et ainsi obtenir l'estimateur suivant :

$$\mathcal{E}_{TAN_SURF_VOX}(D_r(\gamma)) = \sum_{v \in D_r(\mathcal{S})} a(v)$$

Dans notre analyse théorique, nous ne distinguons pas ces deux estimateurs.

Nous étendons le travail de LENOIR (1999) avec le théorème suivant :

Théorème 5.9 *Soit \mathcal{S} une surface C^2 dans \mathbb{R}^3 à courbure bornée, l'estimateur d'aire de surface discrète basé sur l'intégration discrète des vecteurs normaux converge asymptotiquement si et seulement si l'estimation des normales converge.*

La preuve de ce théorème est donnée en annexe B (théorème B.6 et corollaire B.3).

En utilisant le théorème 5.2, nous avons :

Corollaire 5.2 *Soit \mathcal{S} une surface C^2 dans \mathbb{R}^3 à courbure bornée, l'estimateur d'aire de surface discrète basé sur l'intégration discrète des vecteurs normaux, et dont les normales sont données par l'algorithme décrit dans le paragraphe 5.3.2.3, converge asymptotiquement.*

Ainsi, nous avons un algorithme d'estimation d'aire de surface en deux étapes :

1. Estimation des normales en chaque surfel de la surface discrète
2. Sommation, pour chaque surfel, du produit scalaire entre la normale estimée et la normale élémentaire en ce surfel.

Nous obtenons donc un estimateur :

- optimal en temps puisque l'étape 1 s'effectue en $O(n)$ où n est le nombre de surfels, et puisque l'étape 2 a aussi un coût en $O(n)$ (voir paragraphe 5.3.2.3).
- qui a la propriété de convergence asymptotique pour une famille de surface réelle vaste (voir annexe B).

5.4.3.3 Analyse expérimentale

Dans cette analyse expérimentale, nous illustrons le comportement, non pas de l'estimateur basé sur le calcul optimal dont l'implémentation finale est en perspective, mais de celui basé sur le calcul de normales présenté dans le paragraphe 5.3.2.1.

Nous comparons l'estimateur d'aire de surface discrète basé sur l'intégration de normales à, d'une part, l'estimation basée sur le calcul d'enveloppe convexe dans le cas d'objets discrets convexes, et d'autre part à l'estimation basée sur l'algorithme des Marching-Cubes.

Nous considérons les objets multi-grilles suivants (voir figure 5.30) :

- des sphères discrètes ;
- des cylindres dont l'axe principal est parallèle à l'axe de la grille ;
- des demi-sphères creusées par des sphères de rayon plus petit.

Ces objets sont donc considérés sur des grilles de résolution croissante. La mesure utilisée pour évaluer ces estimateurs est l'erreur relative en pourcent.

La figure 5.31 présente les résultats sur les sphères discrètes. Nous pouvons remarquer que l'erreur des estimateurs basés sur l'enveloppe convexe et sur l'intégration des normales décroît vers 0 pour des résolutions croissantes. De plus, nous pouvons

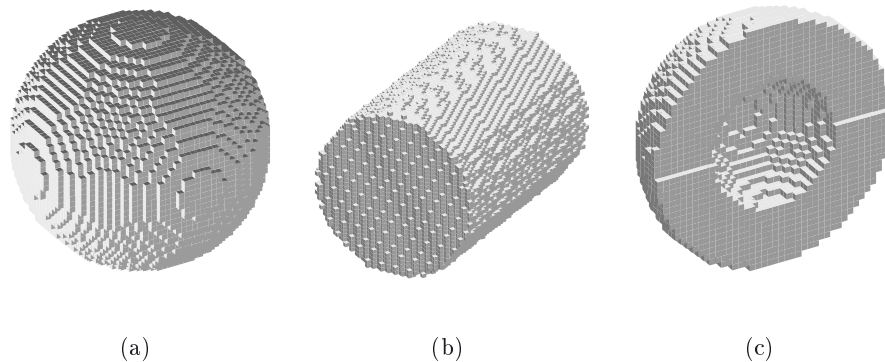


FIG. 5.30 — Volumes discrets multi-grilles considérés pour l'analyse expérimentale des estimateurs d'aire.

remarquer que les résultats obtenus en intégrant les normales théoriques ou en intégrant les normales estimées par l'algorithme du paragraphe 5.3.2.1 sont quasiment confondues. L'approche basée sur l'algorithme des Marching Cubes semble avoir une erreur qui se stabilise vers 10%.

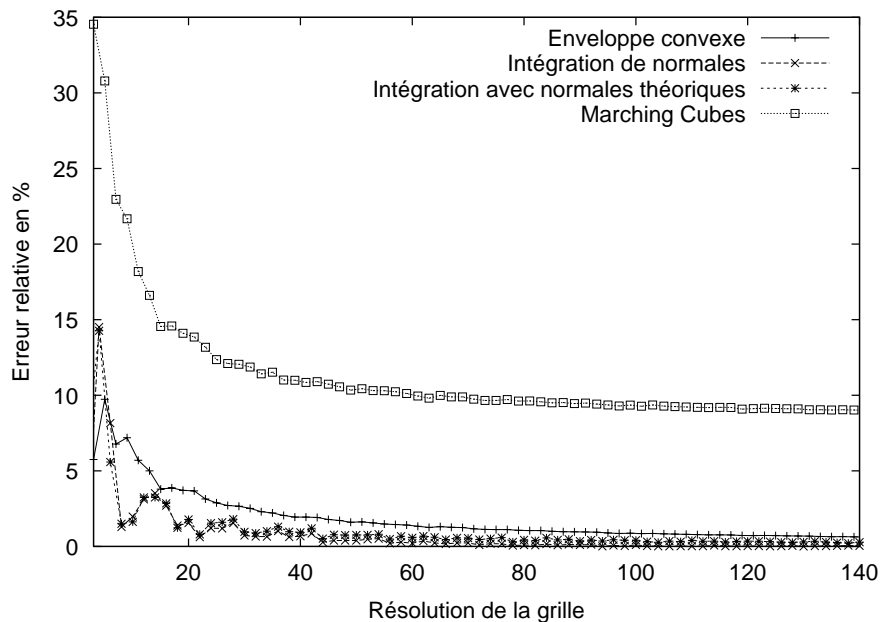


FIG. 5.31 — Convergence asymptotique expérimentale des différents estimateurs sur des sphères multi-grilles.

Les figures 5.32 et 5.33 présentent les différents résultats sur les cylindres et les sphères trouées. Encore une fois, nous pouvons observer le bon comportement de notre estimateur.

Finalement, la figure 5.34 la mesure $a(v)$ associée à chaque voxel et qui correspond à l'apport du voxel v à l'aire globale de la surface. Par définition de cette mesure, celle-ci est comprise entre 1 et $\sqrt{3}$.

Ces résultats ont été présenté dans COEURJOLLY ET AL. (2002a).

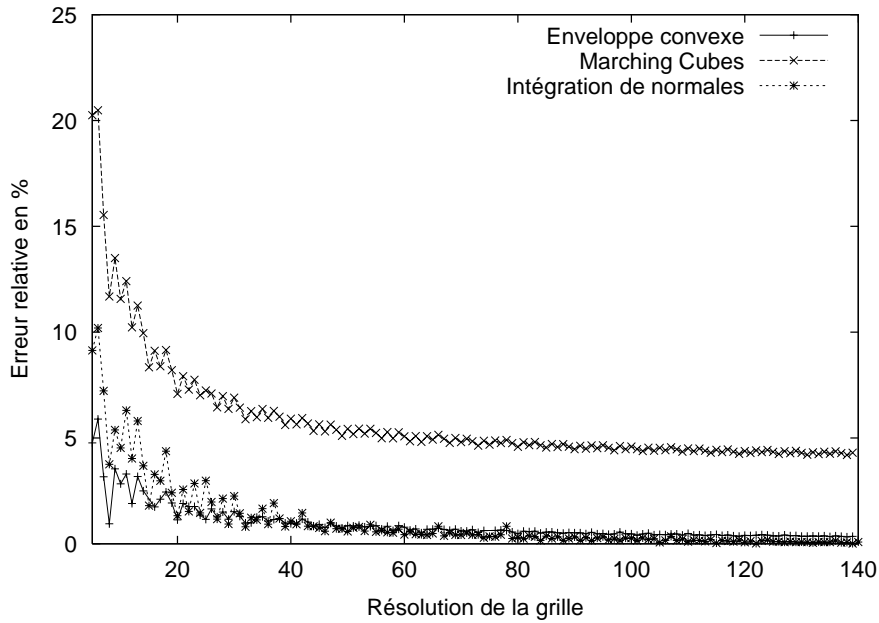


FIG. 5.32 — *Convergence asymptotique expérimentale des différents estimateurs sur des cylindres multi-grilles.*

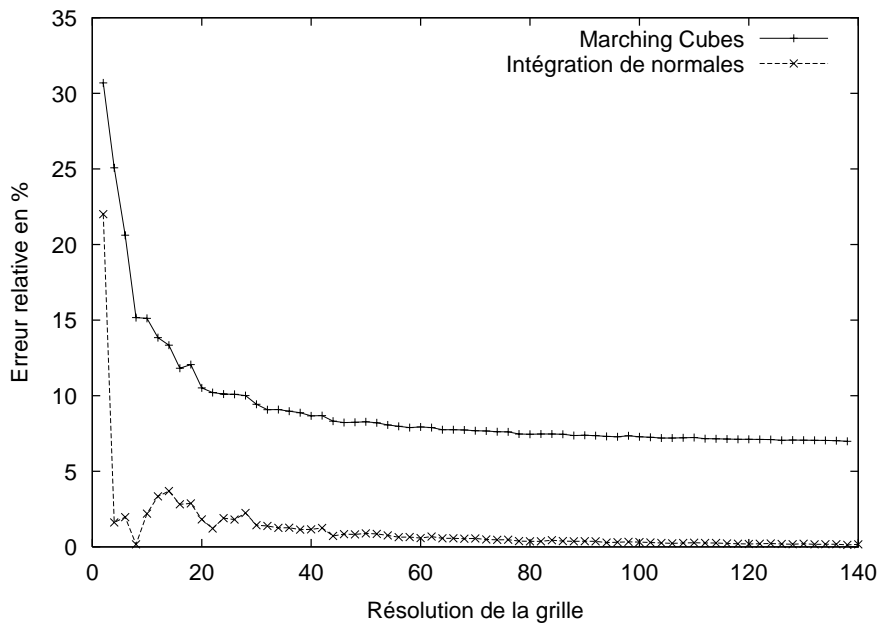


FIG. 5.33 — *Convergence asymptotique expérimentale des différents estimateurs sur des demi-sphères trouées multi-grilles.*

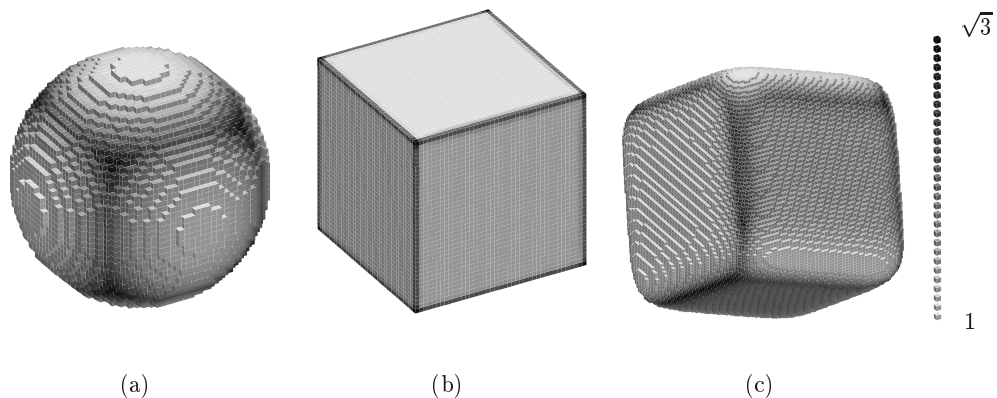


FIG. 5.34 — Illustration de la fonction $a(v)$ représentant l'apport de chaque voxel v à l'aire globale de la surface : (a) sphère discrète, (b) sur un cube et (c) sur un cube lisse tourné.

5.4.3.4 Conclusion sur l'estimation d'aire d'une surface discrète

Sur le plan théorique, nous avons proposé un algorithme optimal en temps et qui a la propriété de convergence asymptotique pour l'estimation de l'aire d'une surface discrète.

Dans la partie expérimentale, nous avons présenté des premiers résultats d'un estimateur qui se base aussi sur l'intégration de normales discrètes mais qui ne correspond pas exactement à l'algorithme optimal.

Une perspective importante de ce travail est l'implémentation, dans une plateforme de tests, de l'algorithme optimal ainsi que d'autres algorithmes proposés dans la littérature.

5.5 Courbures d'objets discrets

5.5.1 Quelques définitions mathématiques

Dans un premier temps, nous présentons les définitions et propriétés de la courbure usuelle dans un espace euclidien. Nous verrons par la suite comment plonger ces définitions dans le modèle discret.

Supposons une courbe $x(s)$, au moins \mathcal{C}^2 et dans un espace de dimension n , paramétrée par son abscisse curviligne :

$$\begin{aligned} x &: [0, 1] \rightarrow \mathbb{R}^n \\ s &\rightarrow x(s) \end{aligned}$$

La courbure, notée $k(s)$, en un point s de x est un scalaire qui mesure la déviation de la courbe par rapport au vecteur tangent. Étant donnée une paramétrisation curviligne de la courbe, nous avons :

$$k(s) = \left| \frac{d^2 x}{ds^2} \right| \quad (5.3)$$

Ce scalaire est bien évidemment toujours positif. Cependant, pour des courbes planes, c'est-à-dire pour $x : [0, 1] \rightarrow \mathbb{R}^2$, nous pouvons signer cette valeur est ainsi

définir la *courbure algébrique* notée k_a et donnée par :

$$\frac{d\vec{t}}{ds} = k_a \vec{N}$$

avec $\vec{t}(s)$ le vecteur tangent à x en s et \vec{N} tel que (\vec{t}, \vec{N}) soit une base orthonormée.

La courbure algébrique peut donc être positive, négative ou nulle et permet de caractériser la concavité ou la convexité locale de x .

Nous avons une définition équivalente de la courbure : si nous considérons $\phi(s)$ l'angle orienté entre le vecteur tangent $\vec{t}(s)$ et une droite fixe, nous avons :

$$k_a(s) = \frac{d\phi}{ds} \quad (5.4)$$

Nous avons une troisième définition donnée par :

$$k(s) = \frac{1}{\rho(s)} \quad (5.5)$$

où $\rho(s)$ est défini comme étant le rayon du cercle qui géométriquement s'ajuste au mieux à la courbe en s . On parle alors de *cercle osculateur*.

Bien évidemment, ces trois écritures de la courbure sont totalement équivalentes. En effet, le cercle osculateur se définit à partir de la dérivée seconde de la courbe. Nous présentons ces différentes approches puisque dans le cas discret, chacune d'entre-elles va engendrer une classe d'estimateurs discrets de courbure.

Considérons maintenant une surface paramétrée $x(u, v)$ dans \mathbb{R}^3 donnée par :

$$\begin{aligned} x &: U \rightarrow \mathbb{R}^3 \\ (u, v) &\rightarrow x(u, v) \end{aligned}$$

avec U un ouvert de \mathbb{R}^2 .

Pour définir une courbure en tout point de cette surface, nous devons introduire un repère en tout point de celle-ci : soit $m = (u, v)$ un point de la surface, nous considérons le repère de FRENET est défini par la normale \vec{N} et un vecteur \vec{t} appartenant au plan tangent à la surface en m (voir figure 5.35). Dans ce repère orthonormé, nous considérons deux courbes $x_1(s)$ et $x_2(s)$ sur la surface contenues dans deux plans Π_1 et Π_2 orthogonaux et, contenant la normale en m (cf figure 5.35). Nous pouvons définir la *courbure moyenne* H en m en fonction des courbures $k_1(m)$ et $k_2(m)$ des courbes x_1 et x_2 :

$$H(m) = \frac{k_1(m) + k_2(m)}{2}$$

Si maintenant nous prenons x_1 (resp. x_2) comme étant la courbe de courbure maximale k_{max} (resp. courbure minimale k_{min}) dans toutes les directions du repère de FRENET. Nous pouvons donner la définition de la *courbure gaussienne* K de la surface au point m :

$$K(m) = k_{max}(m)k_{min}(m)$$

La connaissance de ces deux courbures nous permet de caractériser toutes les configurations locales de la surface (cf tableau 5.2 et figure 5.36).

Nous pouvons remarquer que le calcul de la courbure gaussienne est, en termes de complexité, beaucoup plus difficile à mettre en place que la courbure moyenne qui bénéficie de la propriété d'invariance quelque soit les directions du repère de FRENET.

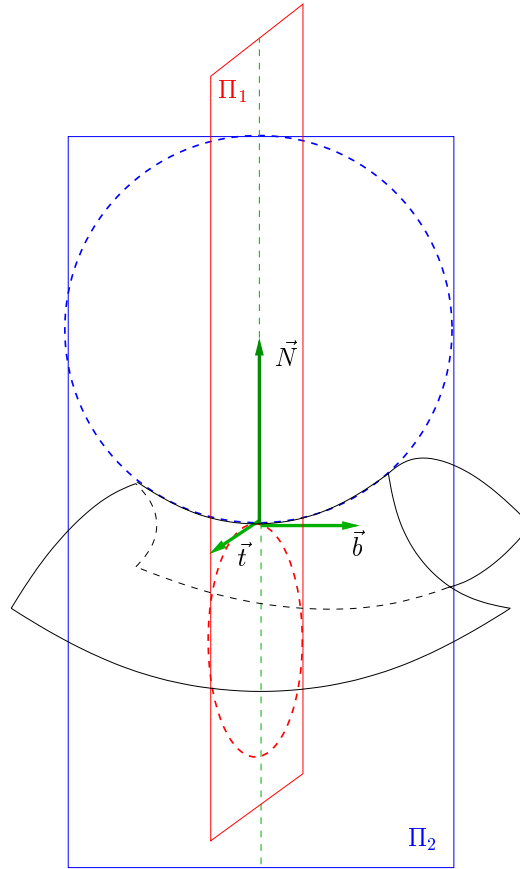


FIG. 5.35 –: Exemple de plans principaux (Π_1 et Π_2) à une surface avec leur cercle principal respectif et le repère de FRENET $(\vec{N}, \vec{b}, \vec{t})$ ainsi défini.

	$K = 0$	$K > 0$ (elliptique)	$K < 0$ (hyperbolique)
$H = 0$	plan	indéterminé	surface minimale
$H > 0$	cylindre concave	vallée	col concave
$H < 0$	cylindre convexe	pic	col convexe

TAB. 5.2 –: Caractérisation locale des surfaces en fonction des courbures moyenne et gaussienne.

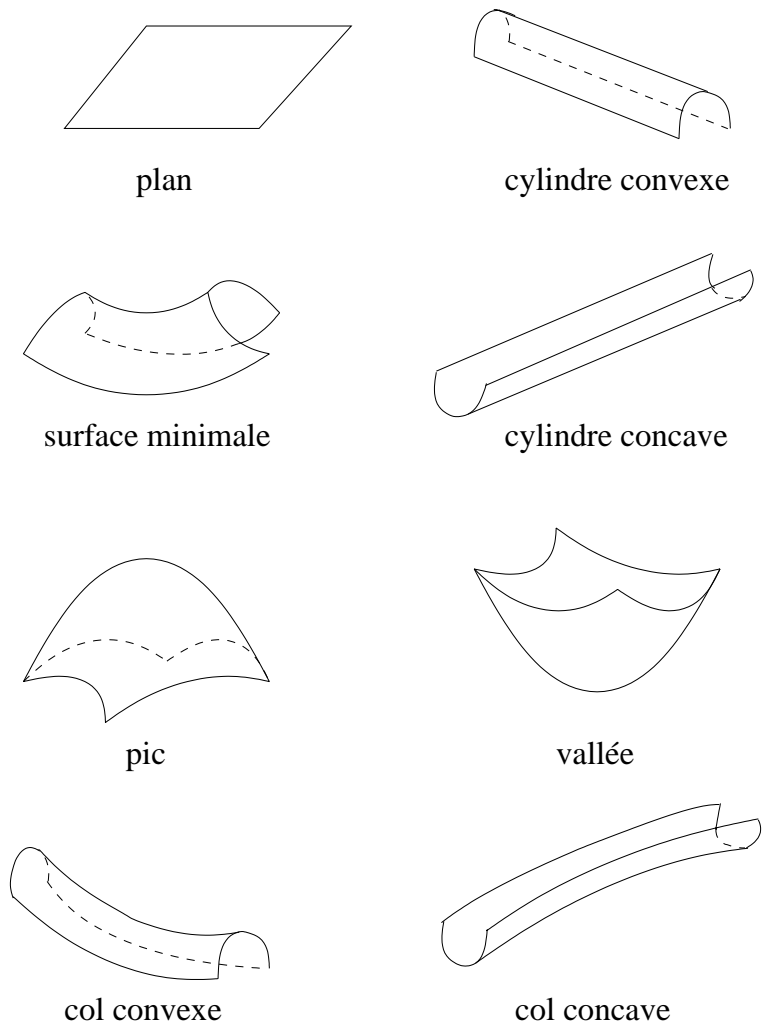


FIG. 5.36 – : *Caractérisation locale des surfaces en fonction des courbures moyennes et gaussiennes.*

5.5.2 Courbure de courbes discrètes

Nous nous intéressons dans ce paragraphe au calcul de la courbure en tout point d'une courbe discrète 2D ou 3D. Cette mesure euclidienne est une information très importante figurant dans la *signature* associée à un contour discret.

Les différents estimateurs que nous proposons correspondent essentiellement aux travaux présentés dans COEURJOLLY ET AL. (2001d) ainsi que dans COEURJOLLY ET AL. (2001c).

5.5.2.1 État de l'art

Dans cette étude bibliographique, nous reprenons l'analyse de WORRING et SMEULDERS (1993) en ajoutant les derniers travaux dans ce domaine. Dans ce qui suit, les estimateurs proposés correspondent tous à des courbures algébriques (donc signées). Le signe peut parfois être ajouté en post-traitement par une analyse de concavité ou convexité locale de la courbe en un point donné.

Ainsi, les équations (5.3), (5.4) et (5.5) ont respectivement introduit des estimateurs basés sur :

1. un calcul de dérivée seconde de la courbe discrète ;
2. un calcul de tangente puis un processus de calcul de variation d'angle des tangentes ;
3. l'estimation ou la reconnaissance du cercle osculateur.

Une des premières solutions pour l'estimation de courbure en un point p_i d'une courbe discrète se base sur la formule suivante (voir CHASSERY et MONTANVERT 1991) :

$$k^*(p_i) = \frac{\angle(p_{i-m}p_i, p_i p_{i+m})}{|p_{i-m}p_i| + |p_i p_{i+m}|}$$

Cette estimation de la courbure est appelée *courbure normalisée d'ordre m* (voir figure 5.37). Dans cette approche, le paramètre m représente un compromis entre précision et lissage du calcul.

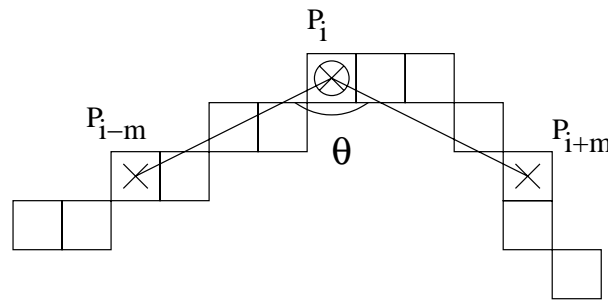


FIG. 5.37 –: Exemple de calcul de courbure normalisée d'ordre m pour $m = 4$.

En se basant sur la formule (5.4), WORRING et SMEULDERS (1993) proposent l'estimateur suivant :

$$k^*(p_i) = \frac{\theta(p_i) * \mathcal{G}'_{\sigma}}{1.107}$$

où :

$$\theta(p_i) = \angle(\text{LINEFITTING}(p_i, m), O_x)$$

et \mathcal{G}'_σ un filtre Gaussien dérivatif d'écart type σ .

Cette approche consiste à estimer une tangente à la courbe en p_i grâce à une technique des moindres carrés sur les points $\{p_{i-m} \dots p_{i+m}\}$ (noté `LINEFITTING` dans la formule). Ensuite, la fonction $\theta(p_i)$ correspond à l'angle entre cette tangente estimée au point p_i et une droite quelconque (ici l'axe O_x). Finalement, une convolution de cette fonction avec un filtre Gaussien dérivatif permet d'estimer la dérivée de θ . Le facteur de régularisation 1.107 correspond à la distance moyenne entre deux pixels de grille sur une distribution aléatoire de droites.

Par la suite, `VIALARD (1996b,a)` remplace l'estimation aux moindres carrés de la tangente par la tangente discrète calculée au point p_i , la formule de l'estimateur reste inchangée. D'un point de vue algorithmique, l'optimisation de `FESCHET et TOUGNE (1999)` permet une estimation de la courbure en tout point de la courbe en temps optimal, c'est-à-dire $O(n)$ pour une courbe de n pixels.

Dans une approche basée sur l'estimation de cercles osculateurs, `WORRING et SMEULDERS (1993)` proposent la formule générale suivante :

$$k^*(p_i) = \frac{1}{r^*(p_i)}$$

avec

$$(c^*, r^*) = \text{ARCFITTING}(p_i, m)$$

En d'autres termes, une estimation du centre c^* et du rayon r^* du cercle osculateur est calculée par moindres carrés (processus noté `ARCFITTING`) sur une fenêtre de m pixels et la valeur de la courbure est donnée par l'inverse du rayon estimé.

Finalement, `WORRING et SMEULDERS (1993)` proposent aussi un estimateur basé sur le calcul de la dérivée seconde de la courbe discrète. Ils montrent cependant que l'erreur engendrée par l'estimation de la dérivée d'ordre deux est assez importante.

Dans ce qui suit, nous allons nous intéresser dans un premier temps à une définition purement discrète de la courbure basée sur l'estimation ou la reconnaissance de cercle osculateur discret. En d'autres termes, de la même manière que `VIALARD` a résolu le `LINEFITTING` en discret, nous souhaitons résoudre l'`ARCFITTING` en discret. Ensuite, nous présentons un estimateur de courbure basé sur la notion de courbure normalisée mais avec l'introduction d'objets discrets que nous appelons *les demi-tangentes discrètes* en un point. Pour ce dernier estimateur, nous présentons une preuve de convergence asymptotique de la mesure.

5.5.2.2 Estimation du cercle osculateur discret

Nous nous intéressons à la résolution du problème `ARCFITTING` de `WORRING et SMEULDERS`. Pour cela, nous allons utiliser l'analyse faite au chapitre 3.

La première solution que nous présentons correspond à une estimation du cercle osculateur très rapide basée sur l'analyse des paraboles discrètes associées aux cercles discrets (voir paragraphe 3.3.2). Nous proposons des premières idées sur un estimateur basé sur la reconnaissance de cercle discret.

Reprenons l'analyse du paragraphe 3.3.2. Le cercle discret arithmétique de centre $(0, 0)$ et de rayon R peut-être construit par le faisceau de paraboles :

$$\begin{cases} y = \left[\sqrt{x(1+2k) - k^2 + 1/4} \right] \\ x = R - k \\ 0 \leq k \leq R \end{cases}$$

Si nous considérons une hauteur l , les cercles arithmétiques dont la première parabole a une hauteur l ont un rayon entier compris entre r_{inf} et r_{sup} donnés par :

$$r_{inf} = \lceil (l - 1/2)^2 - 1/4 \rceil \quad (5.6)$$

$$r_{sup} = \lfloor (l + 1/2)^2 - 1/4 \rfloor \quad (5.7)$$

Cet ensemble correspond au palier de paraboles de hauteur l (voir figure 5.38).

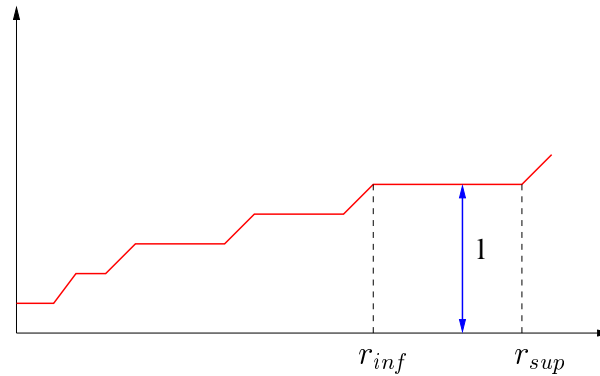


FIG. 5.38 –: *Illustration de la première parabole des cercles arithmétiques et calcul de l'ensemble des cercles de hauteur l .*

Ainsi, l'estimateur basé sur le cercle osculateur que nous proposons utilise l'analogie suivante avec le continu : considérons la tangente en un point d'un cercle de rayon r contenu dans l'anneau $r - 1/2$ et $r + 1/2$. Ce segment a une longueur constante en tout point du cercle de rayon r (voir figure 5.39). Nous partons de l'hypothèse que la discrétisation de ce segment correspond à une tangente discrète d'un cercle arithmétique de rayon r (l'anneau $r - 1/2$, $r + 1/2$ définissant le cercle arithmétique).

Nous émettons donc l'hypothèse que les longueurs des tangentes discrètes en tout point d'un cercle arithmétique sont constantes. Même si cette hypothèse n'est pas vraie dans le cas général, nous montrons que la variation de longueur des tangentes ne perturbe pas l'estimateur de courbure pour des rayons raisonnables.

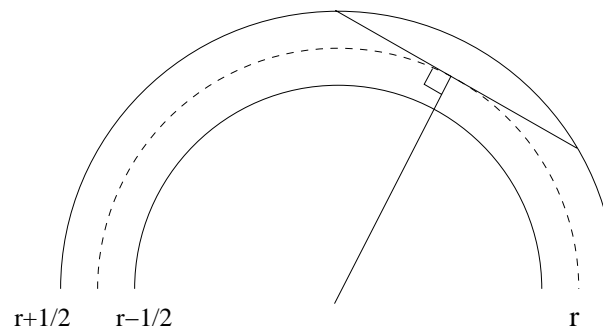


FIG. 5.39 –: *Analogie en l'anneau euclidien $(r - 1/2, r + 1/2)$, la tangente au cercle de rayon r dans cet anneau et la tangente discrète en un point d'un cercle arithmétique.*

Si nous considérons la tangente discrète au point $(r, 0)$ d'un cercle de rayon r , celle-ci correspond exactement à un segment vertical dont la longueur est le double de la hauteur de la première parabole. Ainsi, sous l'hypothèse ci-dessus, nous pouvons associer à chaque demi-longueur l de tangente discrète une estimation des cercles osculateurs discrets donnés par les formules (5.6) et (5.7).

Finalement, à partir de r_{inf} et de r_{sup} , nous estimons la courbure au point p_i comme étant l'inverse du rayon moyen :

$$k^*(p_i) = \frac{2}{r_{inf} + r_{sup}} \quad (5.8)$$

L'algorithme 25 présente l'estimateur complet.

Algorithme 25 Calcul de courbure par estimation de cercles osculateurs discrets

COURBURE_OSCULATEUR(C)

{ C est une courbe discrète}

- 1: Calcul des tangentes discrètes avec l'algorithme de FESCHET et TOUGNE (1999)
 - 2: **pour tout** pixel p_i de C **faire**
 - 3: Soit l la demi-longueur de la tangente en p_i
 - 4: Soit $r_{inf} = \lfloor (l - 1/2)^2 - 1/4 \rfloor$
 - 5: Soit $r_{sup} = \lfloor (l + 1/2)^2 - 1/4 \rfloor$
 - 6: $k^*(p_i) = \frac{2}{r_{inf} + r_{sup}}$
 - 7: **fin du pour**
-

Sur le plan algorithmique, nous pouvons remarquer que l'algorithme 25 s'effectue en $O(n)$ puisque les tangentes discrètes sont calculées en temps optimal (FESCHET et TOUGNE, 1999). Nous obtenons donc une estimation de la courbure très rapide et construite uniquement à partir de rayons entiers de cercles osculateurs. Notons que la valeur de la courbure calculée est, par la suite, signée en analysant la convexité ou concavité locale de la courbe discrète grâce aux paramètres des tangentes des points adjacents à p_i .

La figure 5.40 présente une première comparaison entre cet estimateur et celui proposé par VIALARD (1996a) sur un cercle de rayon 40. Nous pouvons remarquer que, sans introduction de paramètre σ toujours difficile à choisir en fonction de la donnée, nous avons une meilleure estimation de la courbure en tout point. De plus, nous notons que l'hypothèse de travail que nous avons faite, selon laquelle la longueur des tangentes discrètes en tout point d'un cercle arithmétique est constante, reste valide au vue des résultats : la variation de longueur l permet d'avoir de bon résultat sur l'estimation.

De plus, n'ayant pas de processus de lissage dans le coeur même de l'algorithme, nous avons de très bonnes localisations des points de forte courbure sur des courbes non bruitées (voir figure 5.41). Dans le cas de courbes bruitées, nous pouvons utiliser des processus de filtrage en post-traitement mais ceux-ci sont dissociés de la méthode.

Du point de vue de la convergence asymptotique, cet estimateur n'est pas du tout adapté. En effet, nous pouvons remarquer que pour des longueurs l croissantes, l'incertitude sur le rayon croît (le palier de la première parabole est de plus en plus long). Ainsi, cet estimateur permet une estimation de la courbure très rapide, assez efficace sur des objets de taille raisonnable et sans aucun paramètre.

Afin d'avoir non plus un processus d'estimation de cercle osculateur mais un processus de reconnaissance, nous pouvons envisager l'utilisation des différents algorithmes du chapitre 3. L'algorithme 26 présente une solution directe au problème du calcul de courbure par reconnaissance de cercles osculateurs. Cet algorithme repose sur la définition suivante :

Définition 5.4 (Cercle osculateur discret) *Étant donné une courbe discrète C et un point p_i de C . Le cercle osculateur discret à C en p_i est le plus grand morceau de cercle discret qu'il est possible de reconnaître, centré en p_i .*

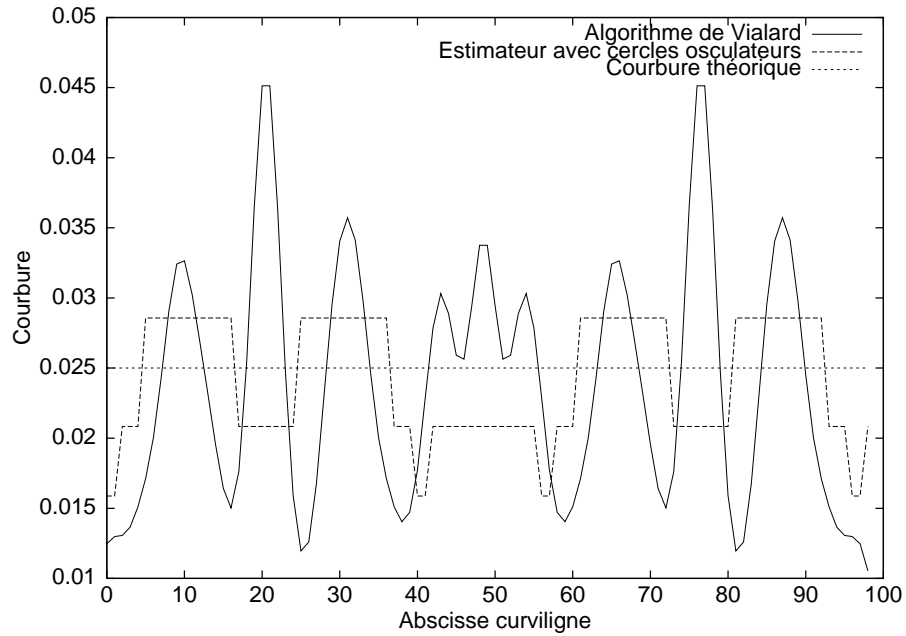


FIG. 5.40 —: Comparaison entre le graphe de courbure de VIALARD (1996a) (avec $\sigma = 1$) et l'estimation de l'algorithme 25 sur un cercle discret de rayon 40.

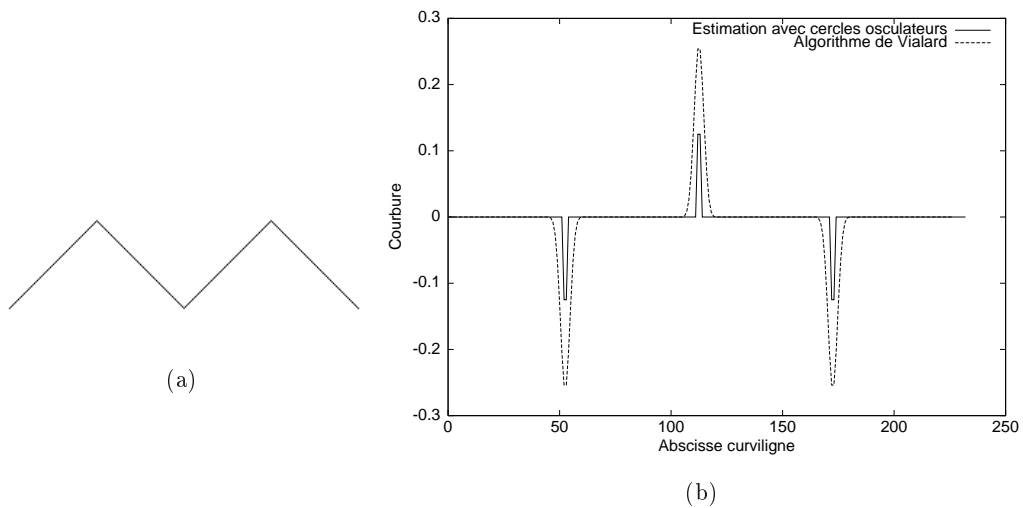


FIG. 5.41 —: Comparaison entre le graphe de courbure de VIALARD (1996a) (avec $\sigma = 1$) et l'estimation de l'algorithme 25 sur un carré déplié.

Cette définition est similaire à celle de la tangente discrète proposée par VIALLARD.

Algorithme 26 Calcul de courbure par reconnaissance de cercles osculateurs discrets

COURBURE_RECONNAISSANCE_OSCULATEUR(C)

{ C est une courbe discrète}

- 1: **pour tout** pixel p_i de C **faire**
 - 2: Reconnaître le plus grand arc de cercle centré en p_i
 - 3: Calculer, par exemple, le cercle (c^*, r^*) avec c^* barycentre du polygone acd résultat de la reconnaissance
 - 4: $k^*(p_i) = \frac{1}{r^*}$
 - 5: **fin du pour**
-

Dans les résultats de segmentation des figures 3.15 et 3.16, nous pouvons observer que la qualité de localisation du centre théorique du cercle est de l'ordre du pixel. Ainsi, ces premiers résultats laissent imaginer un très bon comportement de ce type d'estimateur de courbure.

Un inconvénient majeur de cet algorithme reste sa complexité. En effet, nous avons vu dans le chapitre 3 que la reconnaissance et la segmentation en arcs de cercles discrets d'une courbe discrète de n points ont une complexité en $O(n^{4/3} \log(n))$. Or, une implémentation directe de l'algorithme 26 nous oblige à appliquer un tel calcul en tout point de la courbe ce qui nous donnerait un coût au pire cas en $O(n^{8/3} \log(n))$.

Une perspective de cette approche serait d'analyser le comportement du polygone des solutions du cercle osculateur discret lorsque nous passons d'un point p_i au point suivant de la courbe.

5.5.2.3 Estimation de courbure avec convergence asymptotique

Dans ce paragraphe, nous présentons un estimateur de courbure qui a la propriété de convergence asymptotique. Cet estimateur repose sur une extension de la *courbure normalisée d'ordre m* (voir paragraphe 5.5.2.1 et CHASSERY et MONTANVERT 1991) avec des outils de la géométrie discrète.

L'idée principale est de construire, en un point p_i de la courbe, le plus long segment de droite discrète à droite de p_i (respectivement à gauche de p_i), le pixel extrémité de ce segment est noté P_D (respectivement P_G). Ces deux segments seront appelés respectivement **demi-tangente discrète à droite** et **demi-tangente discrète à gauche** à la courbe en p_i . Ainsi, l'estimation de la courbure au point p_i de la courbe est donnée par l'inverse du rayon du cercle passant par les points P_D , p_i et P_G .

L'estimation de la courbure se ramène donc au calcul du rayon du cercle circonscrit au triangle (P_G, p_i, P_D) . Une idée simple pour implémenter ce calcul se base sur l'intersection des médiatrices de $[P_G p_i]$ et de $[P_D p_i]$. Nous présentons cependant une formule directe permettant d'estimer très simplement le rayon R_c du cercle circonscrit à un triangle donné par les longueurs a , b et c de chacune des arêtes (HARRIS et STOCKER, 1998).

Nous calculons tout d'abord l'aire du triangle donnée par :

$$\mathcal{A} = \frac{\sqrt{(b+c)^2 - a^2} \sqrt{a^2 - (b-c)^2}}{4}$$

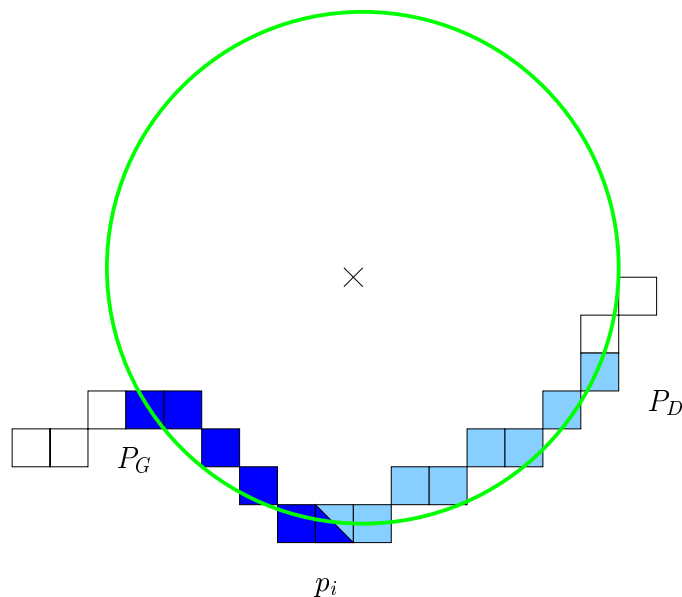


FIG. 5.42 — Illustration de l'estimateur de courbure basé sur les demi-tangentes à gauche et à droite de p_i .

Puis nous avons la formule directe :

$$R_c = \frac{abc}{4\mathcal{A}}$$

Dans notre application, tous les cas dégénérés, qui correspondent à des demi-tangentes discrètes colinéaires par exemple, sont capturés par un simple test de nullité de \mathcal{A} (a , b et c étant non nuls par construction des demi-tangentes). Ce cas particulier nous permet de définir une courbure nulle au point p_i .

L'algorithme 27 présente l'implémentation très simple de cet estimateur de courbure. Le calcul des demi-tangentes discrètes s'effectue grâce aux différents algorithmes présentés dans le chapitre 2. Nous pouvons aussi implémenter dans cet algorithme le calcul du signe de R_c avec une simple extraction du signe du déterminant des vecteurs $\overrightarrow{p_i P_D}$ et $\overrightarrow{p_i P_G}$.

Si maintenant nous regardons le coût de cet algorithme, nous voyons qu'une implémentation directe nous donne une complexité en $O(n^2)$ pour l'étiquetage de chaque pixel par la valeur de la courbure en ce point. Cependant, nous pouvons transformer l'algorithme optimal de calcul des tangentes discrètes de FESCHET et TOUGNE (1999) pour construire en $O(n)$ toutes les demi-tangentes à gauche et à droite en tout point.

En effet, l'algorithme initial de FESCHET et TOUGNE n'impose pas la symétrie complète de la tangente dans le calcul en temps optimal : si l'ajout d'un point à gauche fait échouer la reconnaissance de segment discret, la tangente peut continuer à droite autant que possible, et réciproquement. Pour le calcul des demi-tangentes discrètes à droite, il suffit d'invalider, dès le début, le côté gauche du segment. Nous calculons ainsi en $O(n)$ toutes les demi-tangentes à droite. En appliquant un processus symétrique, nous pouvons calculer les demi-tangentes à gauche dans le même temps.

Par conséquent, l'algorithme 27 est optimal en temps puisqu'il a une complexité en $O(n)$. Notons que cette approche ne nécessite aucun paramètre pour son fonctionnement.

Algorithme 27 Calcul de courbure par demi-tangentes discrètes

COURBURE_DEMI-TANGENTES(C)
 $\{C$ est une courbe discrète}

- 1: **pour tout** pixel p_i de C **faire**
- 2: Calculer la demi-tangente discrète à droite $[p_i P_D]$
- 3: Calculer la demi-tangente discrète à gauche $[P_G p_i]$
- 4: $S = \text{signe}(\det(\overrightarrow{p_i P_D}, \overrightarrow{p_i P_G}))$ {calcul de la convexité ou concavité locale}
- 5: $a = \|\overrightarrow{p_i P_D}\|, b = \|\overrightarrow{p_i P_G}\|, c = \|\overrightarrow{P_D P_G}\|,$
- 6: $\mathcal{A} = \frac{\sqrt{(b+c)^2 - a^2} \sqrt{a^2 - (b-c)^2}}{4}$
- 7: **si** $\mathcal{A} = 0$ **alors**
- 8: $k^*(p_i) = 0$ {Cas dégénéré}
- 9: **sinon**
- 10: $R_c = \frac{abc}{4\mathcal{A}}$
- 11: $k^*(p_i) = S \frac{1}{R_c}$
- 12: **fin du si**
- 13: **fin du pour**

En plus de ce résultat algorithmique, nous avons le théorème suivant :

Théorème 5.10 *Soit Γ la famille des courbes C^2 dans \mathbb{R}^2 à courbure bornée (plus quelques hypothèses techniques), l'estimateur de courbure basé sur les demi-tangentes discrètes converge asymptotiquement.*

La preuve est donnée dans l'annexe (théorème B.4).

En conclusion, nous avons présenté un estimateur de courbure basé sur la notion de demi-tangente discrète en un point d'une courbe qui est optimal en temps et qui à la propriété de convergence asymptotique.

Des analyses expérimentales préliminaires ont été effectuées pour évaluer cet estimateur mais une analyse comparative précise avec les algorithmes présentés ci-dessus est en cours de développement.

5.5.2.4 Courbures de courbes discrètes 3D

Nous présentons ici quelques solutions au problème du calcul de courbure sur des courbes discrètes tridimensionnelles. Dans la littérature, nous n'avons pas trouvé d'estimateurs existants pour un tel calcul. Ceci s'explique par le fait que peu d'applications existent dans la pratique. Récemment, ARONSSON (2002) a montré l'utilité d'une telle mesure dans un projet d'analyse de fibres de papier.

Dans ce qui suit, nous présentons des estimateurs de courbure de courbes discrètes 3D basés sur les algorithmes 25 et 27. Par la suite, nous ne parlerons que de courbure non signée, les notions de convexité ou de concavité locales n'ayant pas de sens en 3D.

Dans le cas de l'estimation du cercle osculateur basée sur la longueur de la tangente discrète (algorithme 25), nous pouvons étendre directement cette approche au cas 3D en considérant la notion de tangente discrète en un point d'une courbe discrète 26-connexe. En effet, si nous utilisons les algorithmes de reconnaissance de droites 3D vus dans le paragraphe 2.3, nous pouvons définir la notion de tangente discrète 3D comme étant le plus long segment de droite, centré au voxel considéré qu'il est possible de reconnaître. Ensuite, une fois la demi-longueur l calculée de

cette tangente, nous pouvons utiliser la formule (5.8) pour estimer la courbure en ce voxel.

D'un point de vue algorithmique, il est possible de rendre ce calcul optimal en temps. En effet, compte tenu du fait que les droites discrètes 3D se projettent sur les plans d'axe de la grille suivant des droites discrètes 2D, nous pouvons utiliser le calcul optimal des tangentes 2D pour construire la tangente 3D. Il suffit pour cela d'effectuer le calcul de tangente en temps optimal dans les trois plans de projections de la grille, puis de définir la tangente 3D à partir des résultats 2D obtenus dans les coupes.

La figure 5.43 présente un exemple de graphe de courbure calculé avec cet estimateur. Dans cet exemple, nous pouvons distinguer sur le graphe : la zone de courbure presque constante qui correspond à l'arc de cercle, la zone de forte courbure qui correspond à la jonction entre l'arc de cercle et le segment de droite et enfin un ensemble de voxels à courbure presque nulle qui correspond au segment.

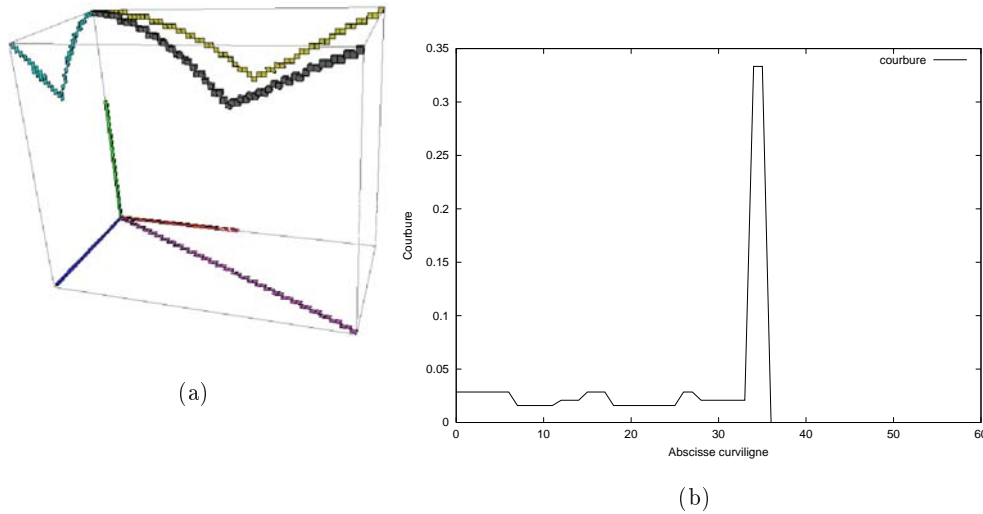


FIG. 5.43 — Exemple de calcul de courbure sur une courbe discrète 3D avec une estimation du cercle osculateur : (a) courbe discrète 26-connexe composée d'un arc de cercle et d'un segment de droite discrète 3D, (b) graphe de courbure obtenu.

Nous pouvons aussi présenter une extension de l'algorithme basée sur les demi-tangentes (algorithme 27). En effet, pour les mêmes raisons que précédemment, nous pouvons définir la notion de demi-tangente discrète 3D en un point d'une courbe discrète 26-connexe. De plus, ce qui est remarquable avec les formules de calcul du rayon R_c (voir ci-dessus), c'est que ce dernier ne nécessite que les distances entre les sommets et est défini quelque soit la dimension de l'espace dans lequel se trouve le triangle. Sur le plan algorithmique, ce calcul se fait aussi en temps optimal, c'est-à-dire en $O(n)$ pour une courbe discrète de n voxels, pour les mêmes raisons que ci-dessus.

De plus, la preuve du théorème 5.10 est valide quelle que soit la dimension de la courbe euclidienne sous-jacente (voir annexe B.) Nous avons donc le théorème suivant :

Théorème 5.11 *Soit Γ la famille des courbes C^2 dans \mathbb{R}^3 à courbure bornée (plus quelques hypothèses techniques), l'estimateur de courbure basé sur les demi-tangentes discrètes converge asymptotiquement.*

En conclusion, l'algorithme 28 est optimal en temps et a la propriété de convergence asymptotique.

Algorithme 28 Calcul de courbure par demi-tangentes discrètes

 COURBURE_DEMI-TANGENTES(C)

 $\{C \text{ est une courbe discrète 3D}\}$

- 1: **pour tout** voxel v_i de C **faire**
 - 2: Calculer la demi-tangente discrète 3D à droite $[v_i V_D]$
 - 3: Calculer la demi-tangente discrète 3D à gauche $[V_G v_i]$
 - 4: $a = \|\overrightarrow{v_i V_D}\|$, $b = \|\overrightarrow{v_i V_G}\|$, $c = \|\overrightarrow{V_D V_G}\|$,
 - 5: $\mathcal{A} = \frac{\sqrt{(b+c)^2 - a^2} \sqrt{a^2 - (b-c)^2}}{4}$
 - 6: **si** $\mathcal{A} = 0$ **alors**
 - 7: $k^*(v_i) = 0$ {Cas dégénéré}
 - 8: **sinon**
 - 9: $R_c = \frac{abc}{4\mathcal{A}}$
 - 10: $k^*(v_i) = \frac{1}{R_c}$
 - 11: **fin du si**
 - 12: **fin du pour**
-

5.5.3 Courbures de surfaces discrètes

Nous nous intéressons maintenant aux différentes courbures sur surfaces discrètes 3D. Sur le plan théorique, ces mesures sont des éléments importants de la signature d'une surface discrète. D'un point de vue pratique, ces mesures ont aussi de nombreuses applications en physique des matériaux (voir annexe A).

5.5.3.1 Courbure Moyenne

Pour l'estimation de la courbure moyenne, deux approches différentes sont possibles : la première se base sur une analyse volumique de l'objet et la seconde repose sur une décomposition en coupes de la surface discrète.

Dans une première approche volumique, nous pouvons utiliser le formalisme associé aux surfaces implicites afin de proposer un estimateur volumique de courbure moyenne. En effet, si nous considérons une surface implicite donnée par l'iso-surface $f(x, y, z) = 0$, nous pouvons caractériser la valeur de la courbure moyenne en un point de cette surface à partir des dérivées partielles premières et secondes de f dans les trois directions (voir par exemple THIRION et GOURDON 1992; MONGA ET AL. 1994). Si nous reprenons l'analyse présentée dans le paragraphe 5.3.2.1 qui consiste à construire la fonction f à partir de la transformée en distance des voxels de la surface discrète, nous pouvons estimer cette courbure en calculant des dérivées partielles dans cette carte de distance.

Une autre approche volumique, moins coûteuse que la précédente, a été proposée par BULLARD ET AL. (1995). L'estimateur est assez simple et permet une mesure linéaire en la courbure moyenne. L'idée est de centrer, en tout point v de la surface discrète, une boule de rayon r . Ensuite, il faut estimer le volume V de l'intersection entre cette boule et l'objet discret en comptant le nombre de voxels de l'objet dans cette boule (la figure 5.44 illustre ce calcul dans le cas 2D). Finalement, BULLARD ET AL. (1995) proposent la formule :

$$H^*(v) = \frac{3.868}{\pi r^4} V - \frac{7.179}{3r}$$

Les différents nombres réels de cette formule ont été choisis de façon à minimiser l'erreur sur un ensemble de sphères discrètes. La figure 5.45 présente quelques résultats d'étiquetage par la courbure moyenne sur différents objets. Le niveau de gris est proportionnel à la valeur $H^*(v)$: les niveaux clairs correspondent aux parties convexes de la surface et les couleurs sombres aux zones concaves. Étant donné que les poids sont optimisés sur des sphères, nous remarquons que les centres des faces des cubes sont étiquetés "faiblement concaves" avec l'estimateur précédent. Bien évidemment, la qualité de l'approximation dépend aussi du choix du rayon de la boule r en fonction des objets à analyser.

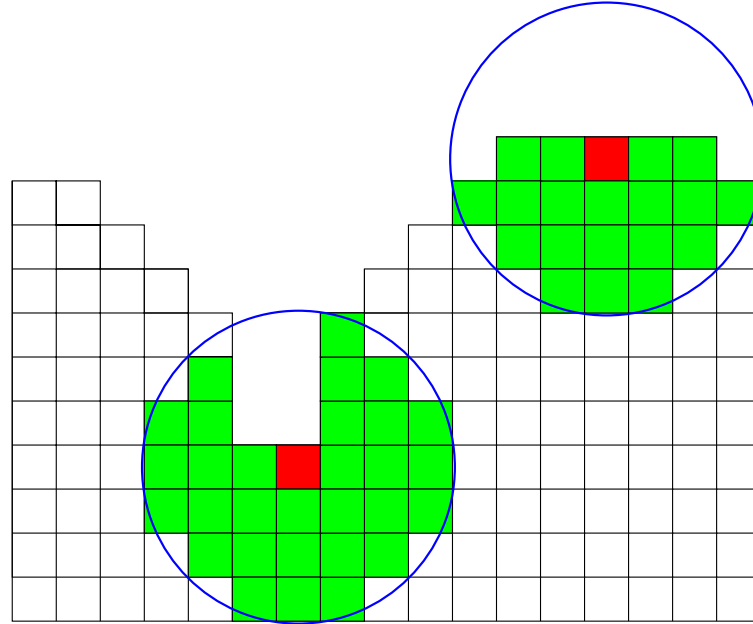


FIG. 5.44 –: *Illustration en 2D de l'estimateur de courbure moyenne de BULLARD ET AL. (1995).*

Dans une approche surfacique, LENOIR (1999) propose une décomposition de la surface discrète en coupes pour le calcul de la courbure moyenne. Ainsi, étant donné un surfel s de la surface discrète, on commence par calculer la valeur de la courbure 2D en s dans les deux coupes associées à ce surfel. Soient $k_1(s)$ et $k_2(s)$ ces deux valeurs, et $\vec{n}(s)$ la normale en s .

Rappelons que nous avons :

$$H(m) = \frac{k_1(m) + k_2(m)}{2}$$

si $k_1(m)$ et $k_2(m)$ sont calculées sur des courbes contenues dans des plans orthogonaux contenant la normale en m .

Dans cette approche par coupes, deux possibilités s'offrent à nous :

- la normale a une ou deux de ses composantes nulles, dans ce cas les deux coupes sont des plans orthogonaux contenant la normale. Nous utilisons donc l'estimateur :

$$H^*(s) = \frac{k_1(s) + k_2(s)}{2}$$

- la normale a ses trois composantes non nulles, les plans de coupe sont orthogonaux mais ne contiennent pas la normale. La formule précédente ne s'applique pas.

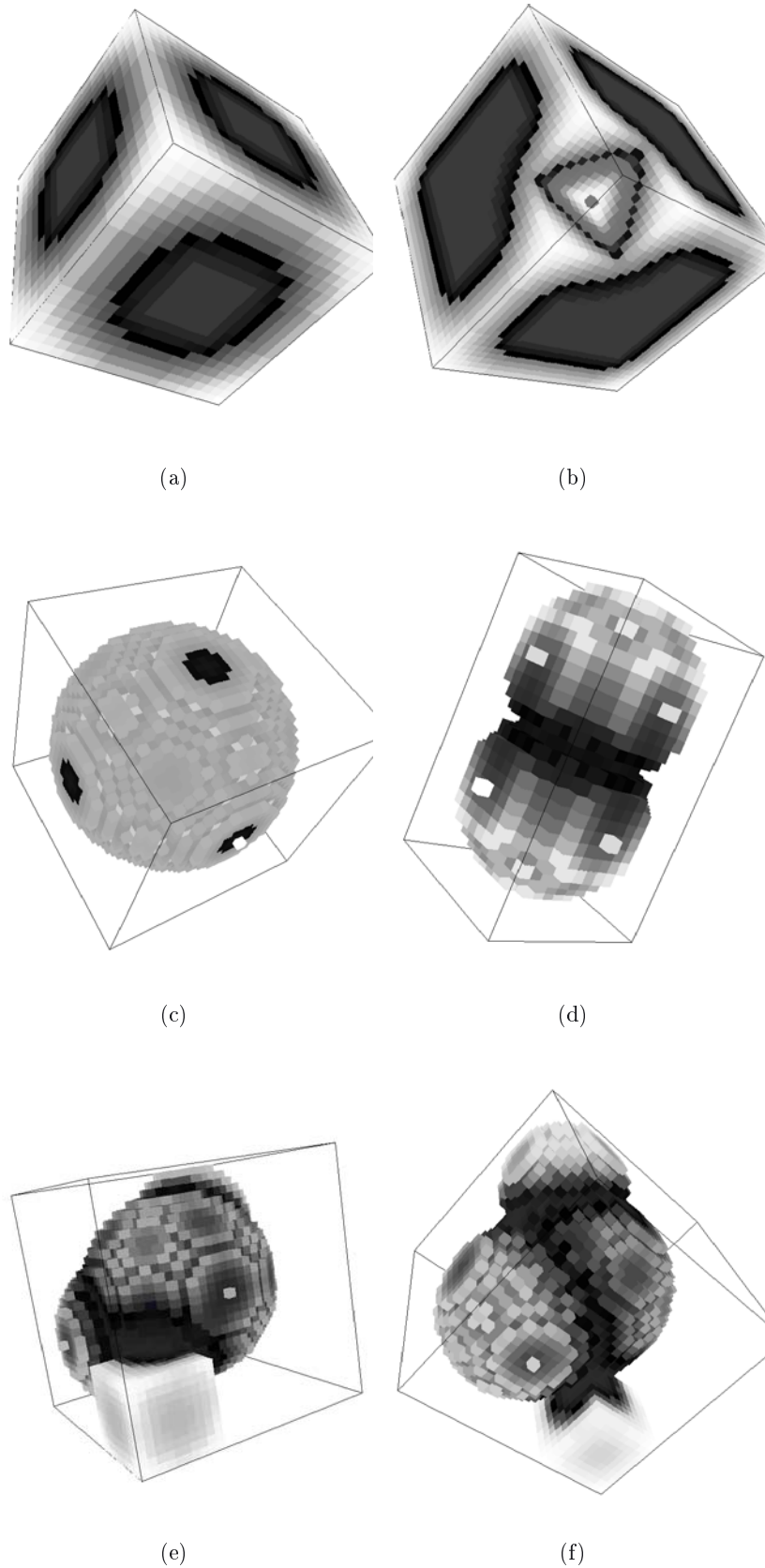


FIG. 5.45 – Exemples de calcul de courbure moyenne avec l'estimateur de BULLARD ET AL. (1995) : (a) sur un cube discret, (b) sur un cube dont un coin est chanfreiné, (c) sur une sphère discrète, (d) sur un objet composé de deux sphères connexes et (e) – (f) sur un objet composite.

Pour résoudre ce dernier point, LENOIR (1999) propose l'utilisation de la formule d'EULER : si nous connaissons les courbures k_1 , k_2 et k_3 en un point m , de trois courbes adjacentes en m , la courbure moyenne est alors une combinaison linéaire de ces courbures :

$$H(m) = w_1 k_1 + w_2 k_2 + w_3 k_3$$

avec $\{w_i\}_{1..3} \in \mathbb{R}$.

En se basant sur cette formule, LENOIR (1999) construit trois courbures, les deux premières correspondent aux courbures dans les coupes $k_1(s)$ et $k_2(s)$, la troisième correspond à une estimation basée sur une analyse locale en s .

5.5.3.2 Courbure Gaussienne

Sur le plan bibliographique, peu de solutions existent pour l'estimation de courbure Gaussienne sur des surfaces discrètes. Dans le cas de la courbure moyenne, différentes recherches ont été menées en physique des matériaux (voir par exemple BULLARD ET AL. 1995) puisque cette information à un sens en terme d'équilibre d'interface.

La courbure Gaussienne quant à elle ne semble pas avoir une signification physique directe. Cependant sur le plan de la description de forme, elle est une mesure prépondérante. Nous proposons donc un estimateur de courbure Gaussienne qui permet un calcul rapide de cette mesure.

Cet estimateur se base sur le calcul de courbure grâce à l'application de GAUSS :

Définition 5.5 (Application de GAUSS) *L'application de GAUSS, notée \mathcal{G} , associe à tout point d'une surface M orientable et au moins C^1 , son vecteur normal orienté. Cette application est illustrée par une injection entre les points de M et les points à la surface de la sphère unité.*

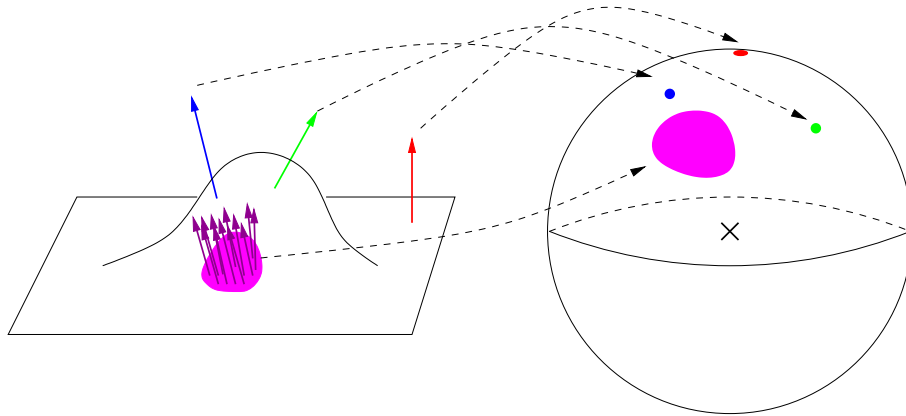


FIG. 5.46 —: *Illustration de l'application de GAUSS dans le cas de surfaces dans \mathbb{R}^3 .*

De plus, nous avons le théorème suivant en géométrie différentielle classique :

$$K(m) = \lim_{v \rightarrow m} \frac{\mathcal{A}(\mathcal{G}(v))}{\mathcal{A}(v)}$$

où v est un ouvert sur la surface contenant m tendant vers ce dernier. $\mathcal{A}(v)$ est l'aire de ce voisinage sur la surface et $\mathcal{A}(\mathcal{G}(v))$ l'aire de l'ouvert sur la sphère unité de l'application de GAUSS \mathcal{G} de v .

Dans le cas des surfaces triangulées, cette relation entre la courbure et l'aire a permis la construction d'estimateurs très simples. Par exemple, la figure 5.47 illustre

un estimateur simple de courbure Gaussienne en un sommet d'une triangulation (voir par exemple BOIX (1995)). Cet estimateur consiste à envoyer les normales des facettes adjacentes au sommet sur la sphère unité et ensuite à faire le rapport entre la somme des aires des triangles et l'aire du polygone sphérique engendré par les normales sur la sphère unité.

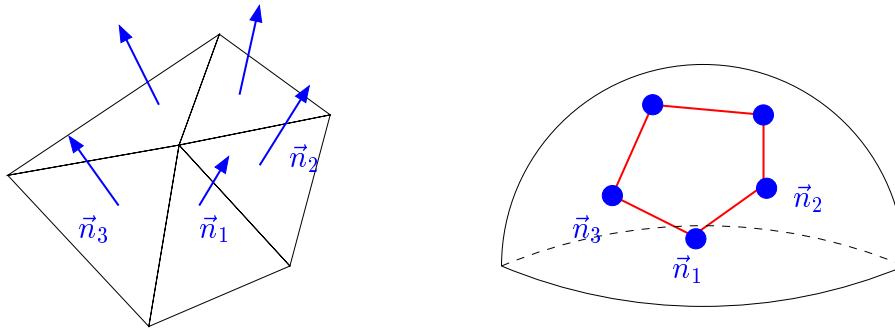


FIG. 5.47 — Application de GAUSS et estimation de courbure Gaussienne dans le cas d'une surface triangulée.

Nous présentons brièvement cet estimateur mais une importante théorie mathématique existe pour prouver une sorte de convergence asymptotique de ce dernier (BOIX, 1995).

Dans le cas discret, une application directe de cet estimateur n'est pas réaliste. En effet, celle-ci consisterait en la projection sur la sphère unité des normales issues des surfels adjacents à un certain pointel de la surface. Étant donné que le nombre de telles configurations locales est faible, la mesure de courbure ainsi obtenue n'aurait pas un très bon comportement.

Cependant le fait d'avoir beaucoup de normales (une par surfel) pour caractériser une petite partie de la surface va nous permettre de proposer des estimateurs très efficaces. L'idée est la suivante : considérons une surface sur laquelle nous avons estimé la normale en tout surfel. Pour un surfel s de la surface, nous nous intéressons au voisinage géodésique de normales autour de s paramétré par une distance d (voir paragraphe 4.4.4). Nous considérons un processus, pour l'instant abstrait noté $\text{AIRE_SPHÈRE}(\mathcal{G}(v))$, qui nous permet d'estimer l'aire associé à un ensemble fini de points sur la sphère unité donné par l'application de GAUSS des normales dans le voisinage v . Nous définirons plus formellement ce processus par la suite. Enfin, nous utilisons l'estimateur de surface basé sur l'intégration de normales du paragraphe 5.4.3 pour estimer l'aire $\text{AIRE}(v)$ de la surface associée au voisinage v . Nous proposons donc l'estimateur :

$$K^*(s) = \frac{\text{AIRE_SPHÈRE}(\mathcal{G}(v))}{\text{AIRE}(v)}$$

Par rapport aux approches basées sur des surfaces triangulées, nous considérons un ensemble de normales dans un voisinage plus grand qu'une simple adjacence avec le surfel s . Le paramètre d , fixant le rayon géodésique de ce voisinage, permet un contrôle de l'estimateur.

Le problème maintenant est donc d'estimer l'aire associée au nuage de points $\mathcal{G}(v)$ sur la sphère unité. La figure 5.48 nous illustre le problème avec les deux solutions que nous avons apporté.

Nous définissons l'aire de ce nuage de points comme étant l'aire du plus petit polygone sphérique convexe contenant ces points. Avant de poursuivre, le choix de

se baser sur la notion d'enveloppe convexe n'est pas anodine. En effet, au sens de la géométrie différentielle, l'application de GAUSS sur un ouvert convexe reste convexe sur la sphère unité. Or, notre définition de disque géodésique surfacique semble proche d'une notion de convexité sur la surface euclidienne sous-jacente (une preuve formelle de ce point est en cours de rédaction).

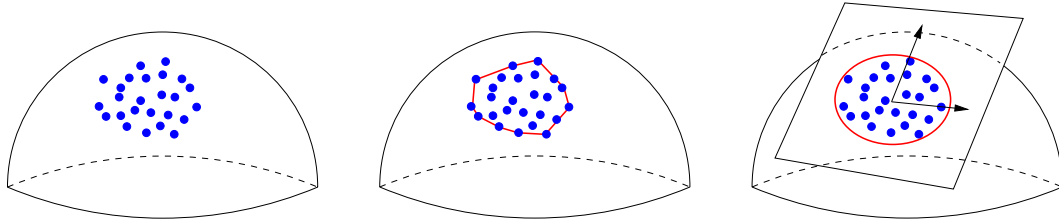


FIG. 5.48 —: *Illustration des différentes solutions pour l'estimation de l'aire d'un nuage de points sur une sphère : (de gauche à droite) les données du problème, solution par calcul d'enveloppe convexe sphérique, solution par calcul d'une ellipse de meilleure approximation.*

La première solution est très géométrique et se base sur la notion d'enveloppe convexe sphérique. Avant cela, rappelons quelques principes de la géométrie sur la surface d'une sphère. On appelle *grand cercle* l'intersection de celle-ci avec un plan contenant son centre. Le rayon d'un grand cercle correspond au rayon de la sphère. Un *triangle sphérique* (ou *triangle d'EULER*) est tel que ses trois arêtes correspondent à des arcs de grands cercles distincts. Enfin, un *polygone sphérique* correspond à une généralisation des triangles sphériques : les arêtes correspondent à des arcs de grands cercles. Un angle d'un triangle ou d'un polygone sphérique en un sommet correspond à l'angle entre les plans générant les grands cercles associés aux arêtes sphériques du sommet.

Finalement, pour ce qui nous intéresse, l'aire d'un polygone sphérique P de n sommets et dont la somme des angles est Θ , est donnée par la formule très simple :

$$\text{AIRE_SPHÈRE}(P) = (\Theta - (n - 2)\pi) R^2$$

où R est le rayon de la sphère (ici $R = 1$).

En se basant sur ces notions, nous avons proposé l'algorithme 29. L'idée est d'utiliser l'algorithme de calcul d'enveloppe convexe 2D de GRAHAM (voir par exemple CORMEN ET AL. 1990; DE BERG ET AL. 2000) pour l'étendre au cas sphérique.

Ainsi, basé sur la notion d'angle présentée ci-dessus, nous pouvons construire un tri polaire sphérique étant donné un centre et surtout, nous pouvons construire un prédicat CONVEXE qui étant donnés trois sommets décide si l'angle sphérique formé est convexe ou concave. Ce prédicat nous permet d'implémenter l'algorithme de GRAHAM. Cet algorithme a pour hypothèse que tous les sommets sont dans la même demi-sphère. Dans le cas contraire nous pouvons avoir des ambiguïtés dans la définition du polygone sphérique.

Cet algorithme est assez simple sur le plan théorique mais son utilisation pour l'estimation de courbure rend son implémentation assez difficile. En effet, dans le cas où les normales sont issues d'un morceau de sphère discrète par exemple, les points $\mathcal{G}(v)$ sont assez bien répartis et l'algorithme peut fonctionner sans difficulté (voir figure 5.49-(a)). Par contre, dans le cas d'une surface plutôt plane ou plutôt cylindrique, la courbure Gaussienne associée est alors nulle et donc les points $\mathcal{G}(v)$

Algorithme 29 Aire d'un nuage de points sur la sphère par enveloppe convexe sphérique

AIRE_SPHERE_ENVELOPPE($\mathcal{G}(v)$)

{ $\mathcal{G}(v)$ est le nuage de points sur la sphère unité}

- 1: Choisir un point p_0 de $\mathcal{G}(v)$ dont les coordonnées sphériques (θ, ϕ) sont minimales (ordre lexicographique)
 - 2: Trier $\mathcal{G}(v)$ dans un ordre polaire sphérique de centre p_0
 - 3: Calculer l'enveloppe convexe sphérique par l'algorithme de GRAHAM
 - 4: Calculer la somme des angles Θ du polygone résultant
 - 5: **retourner** $\Theta - (n - 2)\pi$
-

sont, soit tous regroupés en paquets, soit étirés dans une direction (voir figure 5.49-(b)). Dans ce cas, de nombreux problèmes numériques surviennent dans le calcul du tri polaire et dans le calcul de l'enveloppe convexe sphérique.

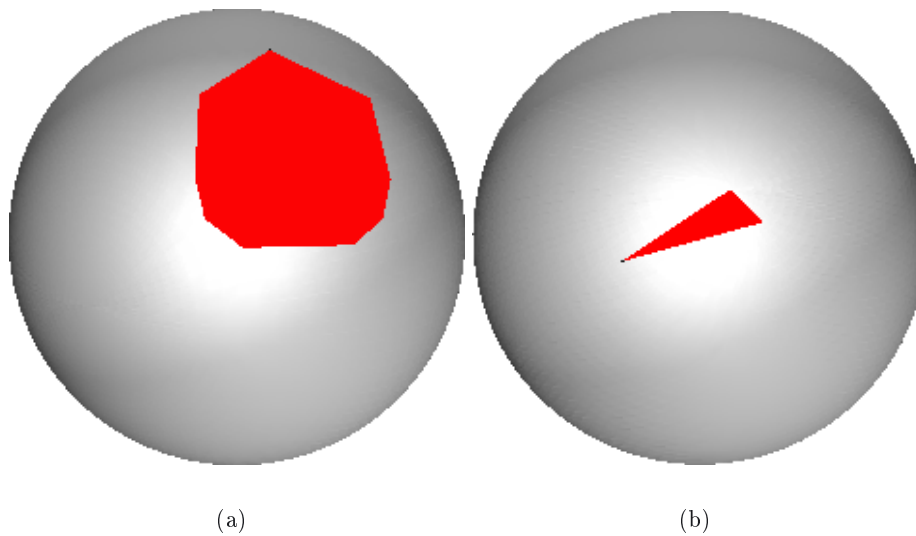


FIG. 5.49 – Exemples de calcul d'enveloppes convexes sphériques : (a) représentation sur la sphère unité du polygone obtenu sur un voisinage v provenant d'une sphère discrète et (b) provenant d'un morceau de cylindre.

Pour résoudre ces problèmes d'instabilité numérique, nous proposons une autre approche basée sur l'ellipse de meilleure approximation. Rappelons le principe de cette approximation dans le cas 2D. Étant donné un ensemble de N points $\{(x_i, y_i)\}_{i=1..N}$ du plan, nous calculons la matrice d'inertie :

$$M = \begin{bmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{bmatrix}$$

où :

$$I_{xx} = \frac{1}{N} \sum_{k=1}^N (x_k - \mu_x)^2$$

$$I_{yy} = \frac{1}{N} \sum_{k=1}^N (y_k - \mu_y)^2$$

$$I_{xy} = \frac{1}{N} \sum_{k=1}^N (x_k - \mu_x)(y_k - \mu_y)$$

$$\mu_x = \frac{1}{N} \sum_{k=1}^N x_k$$

$$\mu_y = \frac{1}{N} \sum_{k=1}^N y_k$$

Ensuite, l'ellipse de meilleure approximation est définie comme étant l'ellipse de centre (μ_x, μ_y) et dont les longueurs des axes sont données par :

$$a_i = \sqrt{\lambda_i}$$

où les λ_i (pour $i = 1, 2$) correspondent aux valeurs propres de la matrice M (voir figure 5.50). Les directions des axes de l'ellipse sont données par les vecteurs propres de M . Enfin, l'aire de cette ellipse est donnée par :

$$\mathcal{A} = \pi \sqrt{\lambda_1 \lambda_2}$$

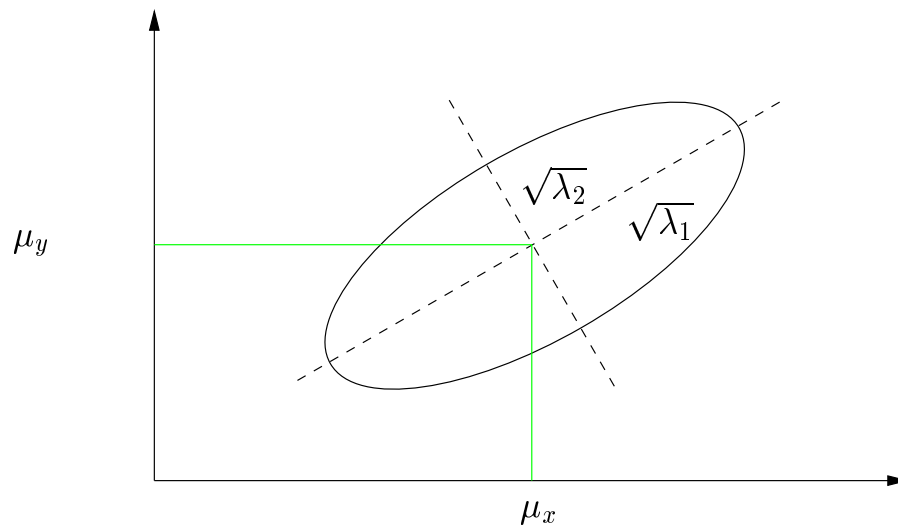


FIG. 5.50 —: *Ellipse de meilleure approximation dans le plan. Les directions des axes principaux sont données par les vecteurs propres de la matrice d'inertie.*

Dans notre cas, l'idée est de résoudre statistiquement notre problème AIRE_SPHERE avec cette notion d'ellipse de meilleure approximation. Cependant, il n'est aisé de calculer des moments de points sur la sphère unité. Nous utilisons pour cela une approximation qui se base sur une projection du nuage de points sur un plan *représentatif*.

Ce plan correspond au plan tangent à la surface de la sphère au point qui est le barycentre du nuage de points. Nous faisons donc une approximation en trois étapes : dans un premier temps, on calcule le barycentre du champ de normales $\mathcal{G}(v)$, ensuite, nous effectuons une rotation de ces normales pour que le barycentre coïncide avec le point $(\pi, \pi/2)$ dans l'espace (ϕ, θ) . Finalement, nous utilisons une projection perspective, dont le centre est le centre de la sphère, sur le plan d'équation $x = -1$ afin d'estimer l'ellipse de meilleure approximation des projections de $\mathcal{G}(v)$. Encore une fois, ce calcul n'est valide que si les points $\mathcal{G}(v)$ sont dans la même demi-sphère. Les deux premières étapes de recentrage des normales nous permettent de simplifier le processus de projection (voir algorithme 30).

Nous obtenons ainsi une estimation de l'aire, notée \mathcal{A}_{proj} , associée au nuage de points $\mathcal{G}(v)$. Ce processus est une approximation d'autant plus juste que les points sont *regroupés* autour du barycentre. Il nous permet une estimation très rapide et avec très peu d'instabilités numériques de la courbure gaussienne.

Nous obtenons donc l'algorithme 30. La figure 5.51 présente quelques résultats de cet estimateurs sur des volumes discrets. Remarquons que dans le cas du cylindre, nous avons effectivement une valeur de la courbure Gaussienne proche de 0 partout sauf sur les bords des bases circulaires (voir tableau 5.2).

Algorithme 30 Aire d'un nuage de points sur la sphère par calcul d'une ellipse de meilleure approximation

AIRE_SPHERE_ELLIPSE($\mathcal{G}(v)$)

{ $\mathcal{G}(v)$ est le nuage de points sur la sphère unité}

- 1: Soit μ le barycentre des normales $\mathcal{G}(v)$
- 2: Appliquer une rotation aux normales pour faire coïncider μ avec $(\pi, \pi/2)$ en coordonnées sphériques
- 3: Projection des normales (x, y, z) en coordonnées sphériques $(\phi, \theta, 1)$ sur le plan $x = -1$ avec la formule :

$$X = \tan(\phi - \pi)$$

$$Y = \tan\left(\theta - \frac{\pi}{2}\right)$$

- 4: Calculer l'ellipse de meilleure approximation (λ_1, λ_2) sur les points $(X_i, Y_i)_{i=1 \dots |\mathcal{G}(v)|}$
 - 5: **retourner** $\mathcal{A}_{proj}(\mathcal{G}(v)) = \pi\sqrt{\lambda_1 \lambda_2}$
-

Dans les estimateurs que nous avons montrés, le signe de la courbure Gaussienne est obtenu en post-traitement. Pour cela, il suffit de considérer une courbe fermée orientée autour du surfel s considéré (la courbure Gaussienne en s doit être non-nulle) et de comparer son sens avec le sens de la courbe obtenue sur la sphère unité par application de GAUSS (voir figure 5.52).

5.6 Conclusion

Dans ce chapitre, nous avons présenté une analyse algorithmique et théorique d'estimations de mesures euclidiennes (longueur, aire, courbure...) sur des objets discrets divers (courbes 2D, 3D, surfaces discrètes...). À chaque fois que cela a été possible, nous nous sommes intéressés à des notions de convergence asymptotique des estimateurs étudiés. Pour résumer brièvement, nous avons montré la convergence asymptotique des estimateurs de longueur de courbes 2D et 3D, d'un estimateur de

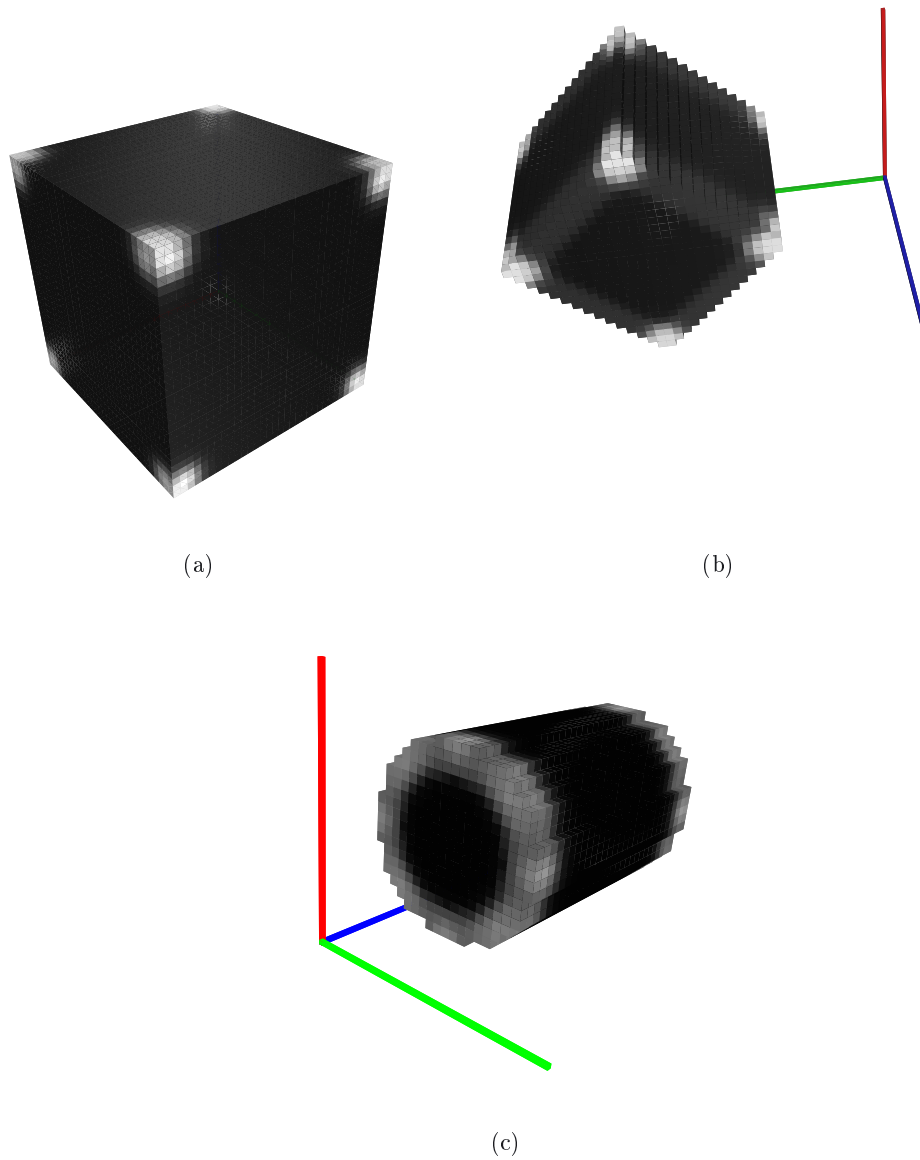


FIG. 5.51 —: Exemples de calcul de courbure Gaussienne : (a) sur un cube dans les axes, (b) sur un cube tourné et (c) sur un cylindre. Les voxels noirs correspondent à des valeurs de courbures nulles ou quasi-nulles.

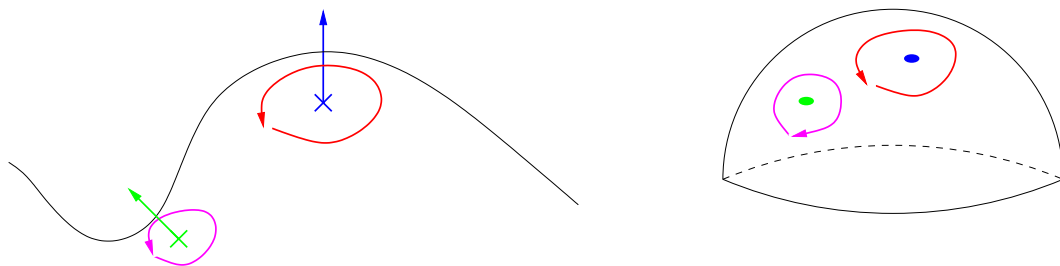


FIG. 5.52 —: Illustration du calcul du signe de la courbure Gaussienne : la courbe rouge et sa transformée par GAUSS sont dans le même sens, le signe est donc positif. Il y a changement de sens pour la courbe magenta, le signe sera négatif.

normales de courbes 2D et de surfaces discrètes, d'un estimateur de courbure de courbes 2D et 3D et enfin d'un estimateur d'aire de surfaces discrètes.

Parmi toutes ces mesures, l'analyse expérimentale de l'estimation de longueur d'une courbe discrète 2D a bénéficié d'une plate-forme de tests conséquente (8 estimateurs implémentés et 5 formes multi-grilles). Une des perspectives majeures de ce chapitre repose donc la création de plates-formes similaires pour l'estimation de courbure et d'aire de surfaces.