



HAL
open science

Diagramme de Voronoi généralisé pour un ensemble de polygones : algorithmes, réalisation et application en analyse de formes

Hai-Tao Hu

► **To cite this version:**

Hai-Tao Hu. Diagramme de Voronoi généralisé pour un ensemble de polygones : algorithmes, réalisation et application en analyse de formes. Modélisation et simulation. Université Joseph-Fourier - Grenoble I, 1991. Français. NNT: . tel-00339655

HAL Id: tel-00339655

<https://theses.hal.science/tel-00339655>

Submitted on 18 Nov 2008

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

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

THESE

présentée par

Hai-Tao HU

POUR OBTENIR LE TITRE DE
DOCTEUR

DE L'UNIVERSITE JOSEPH FOURIER - GRENOBLE 1

(Arrêté ministériel du 23 novembre 1988)

Spécialité : MATHEMATIQUES APPLIQUEES

Diagramme de Voronoï Généralisé Pour un Ensemble de Polygones

algorithmes, réalisation et application en analyse de formes

Soutenue le 1 Juillet 1991
devant le jury composé de :

MM.	Pierre-Jean	LAURENT	<i>Président</i>
	Bernard	LACOLLE	<i>Rapporteur</i>
	Francis	SCHMITT	<i>Rapporteur</i>
	Jean-Marc	CHASSERY	<i>Examineur</i>
	Bernard	PEROCHE	<i>Examineur</i>

Thèse préparée au sein du laboratoire TIM3-IMAG

Remerciements

Je tiens à remercier vivement

Monsieur **Jean-Marc Chassery**, le directeur de cette thèse, pour la grande confiance qu'il m'a accordée tout au long des trois années pendant lesquelles j'ai travaillé à cette thèse, pour m'avoir fait découvrir les joies de la géométrie algorithmique, pour l'aide précieuse qu'il m'a apportée dans tous les aspects, ainsi que pour les corrections de fond et de syntaxe de ma thèse.

Monsieur **Pierre-Jean Laurent**, pour l'honneur qu'il m'a fait en acceptant la présidence du jury de la soutenance de cette thèse.

Monsieur **Bernard Lacolle** et Monsieur **Francis Schmitt**, qui ont accepté d'être les rapporteurs de ma thèse. Je les remercie particulièrement pour la lecture attentive qu'ils ont faite de ce mémoire, ainsi que pour leurs conseils et critiques.

Monsieur **Bernard Péroche**, qui a accepté de participer au jury.

Je remercie également à tous ceux que je ne pourrais pas citer ici qui m'ont aidé au sein de l'Equipe de Reconnaissance des Formes et Microscopie Quantitative du laboratoire TIM3-IMAG.

Je remercie aussi tous mes amis, pour leurs soutiens et aides variés.

A ma femme Zhen REN

献给我亲爱的妻子 任真

Résumé

Ce mémoire présente les problèmes théoriques et pratiques en vue de la réalisation du diagramme de Voronoï généralisé (DVG) pour un ensemble de polygones en métrique euclidienne. Sur la base d'une idée de construction en multi-échelles, le DVG basé sur les objets (polygones) est obtenu d'après le DVG basé sur les éléments (segments, sommets) ; le DVG basé sur les éléments est composé du DVG pour l'intérieur et du DVG pour l'extérieur des polygones.

Pour construire le DVG de l'intérieur d'un polygone, nous avons modifié l'algorithme de Lee [LEE 82a] pour lequel des contre-exemples ont été rencontrés. A partir du concept de visibilité, nous proposons une condition correcte d'arrêt du processus de la fusion de deux DVGs de chaînes.

Pour construire le DVG de l'extérieur d'un ensemble de K polygones, nous avons implémenté un algorithme d'incrémentatation combinée avec "Divide-and-Conquer" dans le cas le plus défavorable en $O(\sum_{i=1}^K n_i \log n_i + \sum_{h=2}^K \sum_{i=1}^h n_i)$ où n_i est le nombre des éléments du i -ème polygone. Une approche de calcul d'un séparateur en temps linéaire à chaque étape d'incrémentatation est proposée. Cette approche constitue la clé de nos algorithmes. Afin de construire efficacement le séparateur, nous avons aussi proposé un algorithme pour trouver en même temps l'enveloppe convexe et l'information d'inclusion de deux polygones convexes d'*intersection simple*.

En développant *a priori* la forme d'un séparateur et en déterminant le nombre des branches (courbes connexes) du séparateur, nous proposons une solution en temps linéaire pour la construction semi-dynamique qui réalise une insertion aléatoire d'un polygone convexe dans un DVG en construction évolutive.

Un exemple des applications du DVG est ainsi présenté. Nous montrons que les squelettes continus de formes polygonales, en analyse de formes, sont des caractéristiques inhérentes au DVG. D'une part, les *axes médians* des formes sont un sous-diagramme du DVG à l'échelle d'éléments pour l'intérieur des formes. D'autre part, les exosquelettes des formes sont aussi un sous-diagramme du DVG de l'extérieur des formes à l'échelle d'éléments, et le SKIZ de formes est identique au graphe du DVG basé sur les objets des formes polygonales.

Mots clés : Géométrie Algorithmique, Diagramme de Voronoï généralisé, Divide-and-Conquer, Incrémentatation, Complexité, Enveloppe convexe, Axe médian, Exosquelette, SKIZ

Abstract

This dissertation presents some theoretical and practical problems for the construction of the generalized Voronoi diagram (GVD) for a set of polygons. The basis of this work is the construction on multi-scales, that is, the GVD under the scale of objects (polygons) is obtained from the GVD under the scale of elements (open segments, vertices) ; the GVD based on elements is composed of the GVD for the interior and the GVD for the exterior of the polygons.

In order to construct the GVD of the interior of a polygon, we have modified Lee's algorithm which meets some counterexamples. The correct condition for terminating a merge process of two subdiagrams based on the concept of visibility between objects is given here.

For the construction of the GVD of the exterior of a set of polygons, we have implemented an incrementation algorithm combined with the Divide-and-Conquer strategy in the worst case in $O(\sum_{i=1}^K n_i \log n_i + \sum_{h=2}^K \sum_{i=1}^h n_i)$, where n_i is the number of the elements decomposed from the contour of the i -th polygon. We propose a method which can compute a separator in linear time at each step of the incrementation. This method constitutes the key of this algorithm. In order to construct a separator efficiently, we propose also an algorithm that can simultaneously find the convex hull and the information related to the inclusion of two convex polygons called simple intersection.

By determining, *a priori*, the shape and number of continuous branches of a separator, we propose a method of semi-dynamic construction which updates in linear time a GVD of a set of polygons when a new convex polygon is randomly inserted into the set of polygons.

The application of the GVD to pattern analysis has also been studied. We show that the continuous skeletons of the polygonal patterns are the inherent features of the GVD. On the one hand, the median axis of a polygonal pattern is a subdiagram of the GVD on the scale of elements for the interior of the pattern. On the other, the exoskeleton among polygonal objects is also a subdiagram of the GVD on the scale of elements for the exterior of the objects, and the SKIZ (SKeleton by Zones of Influence) among polygonal patterns is identical to the graph of the GVD based on objects of the polygonal patterns.

Key words :

Computational geometry, Generalized Voronoi diagram, Divide-and-Conquer, Incrementation, Algorithm, Complexity, Convex hull, Median axis, Exoskeleton, SKIZ

Table des matières

1	Introduction	1
1.1	Géométrie algorithmique.....	1
1.2	Généralisation du diagramme de Voronoï classique.....	1
1.3	Modèles algorithmiques pour le diagramme de Voronoï.....	4
1.4	Travaux antérieurs sur les segments.....	7
1.5	Diagramme de Voronoï de polygones.....	8
1.6	Organisation.....	10
2	Diagramme de Voronoï généralisé: préliminaires géométriques	12
2.1	Définitions et notations de base.....	12
2.2	Bissectrices et séparateurs.....	17
2.3	Intersection entre des arêtes de Voronoï.....	21
2.4	Quelques propriétés du diagramme de Voronoï généralisé.....	24
2.5	Structure de graphe du diagramme de Voronoï généralisé.....	28
2.6	Localisation d'un point dans une région de Voronoï généralisée.....	29
3	Diagramme de Voronoï généralisé d'un polygone basé sur les éléments	36
3.1	Introduction.....	36
3.2	Représentation et décomposition de la frontière d'un polygone.....	37
3.2.1	Représentation basée sur les sommets.....	37
3.2.2	Décomposition basée sur les chaînes de sommets.....	38
3.2.3	Décomposition basée sur les chaînes d'éléments.....	39
3.3	Diagramme de Voronoï pour l'intérieur d'un polygone.....	41
3.3.1	Diagramme de Voronoï d'une chaîne d'éléments.....	41
3.3.2	Approche Divide-and-Conquer pour l'intérieur d'un polygone.....	43
3.4	Diagramme de Voronoï pour l'extérieur d'un polygone.....	52
3.5	Exemples de résultats.....	55
4	Enveloppe convexe de polygones	64
4.1	Introduction.....	64
4.2	Enveloppe convexe d'un polygone simple.....	65
4.2.1	Introduction.....	65
4.2.2	Préliminaires.....	66
4.2.3	Algorithme.....	69
4.3	Enveloppe convexe de deux polygones convexes.....	70
4.3.1	Deux méthodes pour deux polygones convexes.....	71
4.3.1.1	Méthode basée sur le balayage de Graham.....	71

4.3.1.2	Méthode basée sur les lignes d'appui	71
4.3.2	Un nouvel algorithme — libre de contraintes	74
4.3.2.1	Préliminaires	74
4.3.2.2	Raffinement de l'algorithme.....	76
4.4	Enveloppe convexe de deux polygones disjoints	79
4.4.1	Deux polygones convexes d'intersection simple	79
4.4.2	Enveloppe convexe de polygones convexes d'intersection simple.....	80
4.4.3	Enveloppe convexe de deux polygones simples disjoints	82
4.5	Enveloppe convexe d'un ensemble de polygones disjoints.....	85
4.5.1	Prétraitement.....	85
4.5.2	Algorithme	86
5	Construction d'un séparateur	89
5.1	Introduction	89
5.2	Forme d'un séparateur	90
5.3	Calcul de la bissectrice de départ et de la bissectrice d'arrivée.....	95
5.3.1	Cas des séparateurs fermés.....	95
5.3.2	Cas des séparateurs ouverts	99
5.4	Tracé d'un séparateur en temps linéaire.....	103
5.5	Fusion de deux sous-diagrammes de Voronoï	109
5.6	Illustration.....	112
6	Diagramme de Voronoï Généralisé d'un ensemble de polygones	121
6.1	Introduction	121
6.2	Schéma de calcul	122
6.3	Choix d'un modèle algorithmique	124
6.4	Modèle incrémental de construction	127
6.5	Cas particulier — diagramme de Voronoï de polygones convexes	133
6.6	Construction semi-dynamique : incrémentation aléatoire d'un objet.....	136
6.6.1	Introduction	136
6.6.2	Insertion aléatoire d'un point.....	137
6.6.3	Insertion aléatoire d'un segment.....	138
6.6.4	Insertion aléatoire d'un polygone convexe	142
6.7	Exemples de diagrammes de Voronoï généralisés	147
7	Axe médian, exosquelette et SKIZ	156
7.1	Description de formes.....	156
7.2	Définitions existantes de squelette	156
7.3	Approximation de squelettes continus.....	160

7.3.1	Squelettes par la triangulation de Delaunay.....	161
7.3.2	Squelettes par diagramme de Voronoï classique.....	162
7.4	Squelettes continus d'après le diagramme de Voronoï généralisé.....	163
7.4.1	Axe médian.....	163
7.4.2	Exosquelette.....	168
7.4.3	SKIZ.....	170
7.4.4	Résultats et discussions.....	170
Conclusion et perspectives		175
Bibliographie		178

Liste des figures

1.1	Diagramme de Voronoï classique.....	2
2.1	Eléments, projections, images et distances.....	13
2.2	Les bissectrices de deux éléments.....	18
2.3	Séparateurs entre deux ensembles d'éléments.....	19
2.4	Deux exemples simples du diagramme de Voronoï généralisé.....	21
2.5	Intersections des deux paraboles ayant un foyer commun.....	22
2.6	Intersections des deux paraboles ayant une directrice commune.....	23
2.7	Illustration pour lemme 2.6.....	24
2.8	Illustration de lemme 2.12.....	26
2.9	Si $\text{Sep}(S_1, S_2) \cap V(O_1, S_1-O_1) \cap V(O_2, S_2-O_2) \neq \emptyset$, alors $V(O_1, S_1-O_1)$ et $V(O_2, S_2-O_2)$ partagent une arête de Voronoï.....	27
2.10	Division d'un polygone convexe en secteurs.....	31
2.11	Chaîne polygonale monotone par rapport à la droite l.....	32
2.12	La chaîne monotone d'une s-région de Voronoï.....	32
2.13	La frontière d'une s-région de Voronoï est monotone.....	33
2.14	Localisation d'un point dans une p-région de Voronoï.....	34
3.1	Représentation d'un polygone par une liste doublement chaînée.....	37
3.2	Sommet concave (a) et sommet convexe (b).....	38
3.3	Représentation d'un polygone sous forme de chaînes basées sur les sommets.....	38
3.4	Exemples des chaînes.....	40
3.5	Quand on ne s'intéresse qu'à l'extérieur d'un polygone convexe, la chaîne décomposée sera une boucle.....	40
3.6	Diagramme de Voronoï d'un côté d'un segment fermé.....	42
3.7	Diagramme de Voronoï d'un côté pour une chaîne de longueur supérieure à un.....	43
3.8	Chaînes successives et ensembles de chaînes voisins.....	44
3.9	G_L et G_R sont les divisions premières du polygone G.....	46
3.10	G_L et G_R sont les divisions secondaires du polygone G.....	46
3.11	Un exemple ne convenant pas aux conditions d'arrêt de Lee [LEE 82a].....	47
3.12	Visibilité à partir d'un point q.....	48
3.13	Visibilité partielle (a) et entière (b).....	49
3.14	Visibilité intérieure et extérieure.....	49
3.15	Illustration pour le lemme 3.2.....	49
3.16	Un sens spécifié de balayage de la frontière d'un polygone définit	

l'intérieur (a) ou l'extérieur du polygone (b).....	52
3.17 Un polygone ayant quatre chaînes.....	53
3.18 Diagramme de Voronoï pour l'extérieur d'un polygone convexe. Le polygone Ω de N sommets définit, à son extérieur, $2N$ régions de Voronoï dont chacune est numérotée	54
3.19 Diagrammes de Voronoï généralisés pour l'intérieur d'un polygone	
(a) et (b)	55
(c) et (d)	56
(e) et (f).....	57
(g) et (h)	58
(i) et (j).....	59
3.20 Diagrammes de Voronoï généralisés pour l'extérieur d'un polygone	
(a)	60
(b).....	61
(c)	62
(d).....	63
4.1 Un test simple de convexité ne réussit pas pour le polygone donné.....	66
4.2 Illustration d'une poche.....	67
4.3 Une poche maximale.....	68
4.4 L'enveloppe convexe $CH(P \cup Q)$ de deux polygones simples disjoints P et Q . Intersection peut avoir lieu entre $CH(P)$ et $CH(Q)$	71
4.5 "Left tangent "et "right tangent " de deux polygones convexes	72
4.6 Rotation de 90° de la figure 4.5, deux chaînes $CHN1$ et $CHN2$ doivent être retrouvées par y-coordonnée au lieu de x-coordonnée.....	74
4.7 Illustration de la définition 4.8.....	75
4.8 Le graphe de $L_Q(v)$ correspondant aux polygones dans figure 4.7.....	76
4.9 Seul un point d'appui dans le polygone P qui ne peut pas être trouvé par la séquence différentielle	77
4.10 Polygones convexes d'intersection simple	80
4.11 (a) Polygone P est inclus dans polygone Q ; (b) les graphes des séquences $L_Q(v) v \in P$ et $L_P(v) v \in Q$	81
4.12 L'intersection de deux polygones convexes a lieu par couples.....	83
4.13 Intersection de $CH(P)$ et $CH(Q)$ où P et Q sont deux polygones disjoints (a) intersection impossible; (b) intersection possible	84
4.14 Illustration du lemme 4.10.....	85
4.15 Exemple de sortie	
(a) un ensemble de polygones	87

	(b) son enveloppe convexe basé sur les segments d'appui orientés.....	88
5.1	Le séparateur entre $S_1=\{1, 2\}$ et $S_2=\{3, 4\}$ est composé de deux parties	89
5.2	Si les quatre polygones sont divisés en deux sous-ensembles, $S_1=\{P_1, P_2\}$ et $S_2=\{P_3, P_4\}$, alors le séparateur $Sep(S_1, S_2)$ est discontinu	90
5.3	Illustration du lemme 5.1	92
5.4	(a) Le séparateur $Sep(L, R)$ est fermé quand $CH(R)$ est inclus dans $CH(L)$	93
	(b) Aucun cercle centré dans la région ombrée n'inclut e qui est un point de ∂R	94
5.5	Illustration du lemme 5.2 (b).....	94
5.6	Construction d'une demi-droite l_v	96
5.7	Un pont $[b_l, b_r]$ entre deux polygones	100
5.8	Lignes d'appui communes de deux polygones	101
5.9	Illustration du lemme 5.6.....	102
5.10	Séparateur entre deux polygones convexes.....	104
5.11	Quand un séparateur tourne plusieurs fois dans une même région de Voronoï, les intersections de la frontière de la région avec les bissectrices correspondant à chaque composante du séparateur sont ordonnées suivant la frontière de la région	108
5.12	Renouvellement d'une région de Voronoï.....	111
5.13	(a) Trois polygones initiaux P_1, P_2 et P_3	113
	(b) $VOD_e(P_1)$ — DVG basé sur les éléments pour l'extérieur du polygone P_1	114
	(c) $VOD_e(P_2)$ — DVG basé sur les éléments pour l'extérieur du polygone P_2	115
	(d) Superposition de $VOD_e(P_1)$ et $VOD_e(P_2)$, et représentation du séparateur $Sep(P_1, P_2)$ (fermé)	116
	(e) $VOD_e(P_1 \cup P_2)$	117
	(f) $VOD_e(P_3)$	118
	(g) Superposition de $VOD_e(P_1 \cup P_2)$ et $VOD_e(P_3)$, et représentation du séparateur $Sep(P_1 \cup P_2, P_3)$ (ouvert)	119
	(h) $VOD_e(P_1 \cup P_2 \cup P_3)$ — le diagramme final basé sur les éléments.....	120
6.1	Schéma global pour calculer $VOD_o(G)$	122
6.2	Distance entre un point p et un polygone G	123
6.3	Les positions relatives des polygones sont ambiguës.....	125
6.4	Le séparateur est composé de deux morceaux	126
6.5	N polygones peuvent apporter un séparateur de N/2 courbes discontinues	126
6.6	Illustration pour l'algorithme FINDKEYS	130
6.7	Un segment de droite est un polygone le plus simple. Il est composé de quatre éléments différents	135
6.8	L divise S en deux sous-ensembles d'objets invisibles.....	139

6.9	Illustration pour lemme 6.8	140
6.10	Les intersections positives et l'intersections négatives de deux polygones convexes.....	143
6.11	Chaque point d'intersection de P avec $CONV(S)$ est partagé par deux faucilles; l'une des faucilles est dans P et l'autre est dans $CONV(S)$	144
6.12	Illustration du lemme 6.12	147
6.13	(a) Diagramme de Voronoï généralisé d'un ensemble de segments basé sur les éléments	148
	(b) Diagramme de Voronoï généralisé d'un ensemble de segments basé sur les objets.....	148
6.14	(a) Diagramme de Voronoï généralisé d'un ensemble de polygones convexes basé sur les éléments.....	149
	(b) Diagramme de Voronoï généralisé d'un ensemble de polygones convexes basé sur les objets.....	149
6.15	(a) - (h) Des diagrammes de Voronoï généralisés de polygones basé sur les éléments	
	(a)	150
	(b)	150
	(c)	151
	(d)	151
	(e)	152
	(f).....	152
	(g)	153
	(h)	153
6.15	(i) - (l) Des diagrammes de Voronoï généralisés de polygones basé sur les objets	
	(i).....	154
	(j).....	154
	(k)	155
	(l).....	155
7.1	Squelette fondé sur la simulation de la propagation d'une onde	157
7.2	Squelette par boules maximales.....	158
7.3	Trois catégories des points du squelette	159
7.4	Le diagramme de Voronoï d'un ensemble de points et la triangulation de Delaunay correspondante.....	161
7.5	Un point de l'axe médian est déterminé par trois quantités égales de distance	164

7.6	Un point sur le graphe du diagramme de Voronoï est contraint par trois quantités égales de distance	165
7.7	Les points intérieurs dans l'arête $b(p_i, s_j)$ ne sont pas des centres de boules maximales où p_i est un sommet concave et s_j est un segment voisin de p_i	167
7.8	Exemples de l'axe médian	
	(a)	172
	(b)	172
7.9	Exemples de l'exosquelette	
	(a)	173
	(b)	173
7.10	Exemples de SKIZ	
	(a)	174
	(b)	174

Chapitre 1

Introduction

1.1 Géométrie algorithmique

La Géométrie Algorithmique (en anglais *Computational Geometry*) est une discipline relativement nouvelle et florissante en science informatique qui concerne l'analyse et la conception d'algorithmes efficaces pour résoudre des problèmes géométriques. Les études systématiques en géométrie algorithmique ont été initialisées autour de 1975 [SHAMOS 75a] par les travaux de la thèse doctorale de Shamos. Depuis, les recherches croissantes en géométrie algorithmique sont issues non seulement de son intérêt propre mais encore de son étroite relation avec de nombreux domaines appliqués. Par exemple, la conception de circuits VLSI (Very Large Scale Integrated) nécessite des algorithmes rapides pour résoudre des problèmes à base de manipulation de rectangles; en robotique, on a besoin d'algorithmes pour trouver un chemin d'un robot sans collision entre deux positions dans un environnement en présence d'objets géométriques; l'infographie demande des algorithmes performants pour résoudre des problèmes géométriques associés, l'intersection, la visibilité etc. Généralement, on peut classer les problèmes de la géométrie algorithmique selon les axes suivants : convexité, intersection, recherche géométrique, proximité et optimisation, etc... La description et les détails des principaux problèmes de la discipline sont fournis dans l'ouvrage "Computational Geometry" [PREPARATA 88] dont la maquette est la thèse de Shamos [SHAMOS 78].

1.2 Généralisation du diagramme de Voronoï classique

Le diagramme de Voronoï est l'un des sujets très actifs et une des structures de données très puissantes en géométrie algorithmique. Il existe beaucoup de littérature sur le diagramme de Voronoï depuis qu'il a été introduit dans le domaine informatique en 1975 par Shamos et Hoey [SHAMOS 75b]. On a trouvé des applications du diagramme de Voronoï dans plusieurs secteurs: biologie, physique, reconnaissance des formes, vision par ordinateur, robotique etc.

Formellement, le diagramme de Voronoï est un complexe de *cellules* (ou *régions*) définies sur un ensemble d'*objets* (ou *germes*) dans l'espace euclidien. En général, si un ensemble S d'objets o_i est donné dans E^d , la construction du diagramme de Voronoï de S implique la recherche d'une partition de l'espace E^d en cellules $V(o_i)$ de manière à ce que chaque cellule contienne un objet et que tous les points d'une même cellule soient plus proches de l'objet associé que de n'importe quel autre objet. Evidemment, cette définition doit être précisée sur trois points: (i) *dimension* de l'espace d ; (ii) *type d'objets* géométriques concernés; (iii) *métrique* utilisée dans E^d pour la mesure de distance.

En général, on traite des objets dans le plan ($d=2$). Mais le diagramme de Voronoï dans les espaces de dimension supérieure ($d \geq 3$) est aussi considéré [BOWYER 81][WATSON 81] [DWYER 88].

Dans le plan, lorsque les germes sont des points et que la métrique correspond à la distance euclidienne, on a alors le *diagramme de Voronoï classique* [VORONOI 08]. Ceci est illustré en figure 1.1. Cependant, les germes dans l'espace E^d ne sont pas toujours des points. Ils peuvent être associés à des segments, des polygones, des cercles, des portions de courbe, et des objets géométriques dans le cas le plus général.

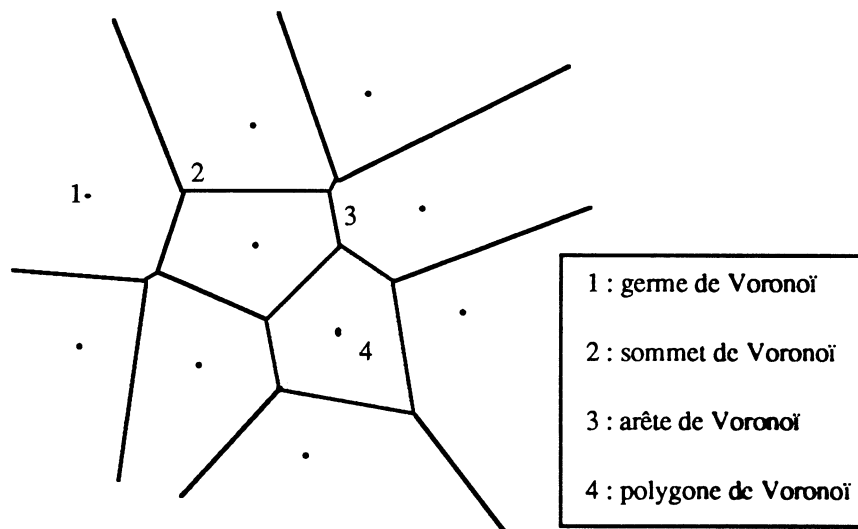


Figure 1.1 Diagramme de Voronoï classique (d'après [CHASSERY 91])

Quant à la métrique utilisée, la mesure de distance n'est pas toujours la distance euclidienne. En fait la mesure de distance dépend souvent de l'environnement où on se trouve. Soit S un ensemble de germes en général. Pour chaque germe g de S , nous définissons une transformation qui associe à un point p de E^d , un nombre réel positif noté $d_g(p)$. Cette transformation est appelée *fonction de distance* de g ou *transformation de distance*. Remarquons que la transformation de distance est en fait une reconstruction de l'espace: reconstruire l'espace

des distances à partir de l'espace des objets. Nous pouvons concrétiser la fonction de distance dans le plan euclidien par quelques exemples suivants.

(1) Fonction de distance d'un point.

Si S est un ensemble de points dans le plan, alors la transformation de distance euclidienne sur S est :

$$d_g(p) = d(g, p) \quad (1.1)$$

En fait, elle appartient à la famille issue d'une transformation de distance plus générale associée à la métrique de *Minkowski* :

$$d_g(p) = (|g_x - p_x|^k + |g_y - p_y|^k)^{1/k} \quad (1.2)$$

Une application de cette transformation de distance est utilisée dans les circuits VLSI où la distance entre deux points a et b est mesurée avec la métrique $L_1(a, b) = |g_x - p_x| + |g_y - p_y|$.

(2) Fonction de distance d'un sous-ensemble de taille fixée.

Si S est composé de tous les sous-ensembles T de points ayant un cardinal k , la fonction de distance d'un sous-ensemble T est alors définie par:

$$d_T(p) = \max \{d(p, t) \mid t \in T\} \quad (1.3)$$

(3) Dans certains cas, chaque point g d'un ensemble S est associé à une pondération w_g réelle non négative. On peut définir [AURENHAMMER 84] [AURENHAMMER 86] la fonction de distance d'un point pondéré g par :

$$d_g(p) = d(p, g) / w_g \quad (1.4)$$

(4) Fonction de distance d'un objet géométrique.

Si S est un ensemble d'objets géométriques, la fonction de distance d'un objet est définie par :

$$d_o(p) = \min \{d(p, q) \mid q \in \partial o\} \quad (1.5)$$

où ∂o représente la frontière de l'objet o .

En résumé, en intégrant les trois aspects de la définition du diagramme de Voronoï, on peut donner ci-dessous une définition généralisée pour le diagramme de Voronoï.

Soit S un ensemble dénombrable de sous-ensemble T_i d'objets dans E^d . L'ensemble de points $\{q \in E^d \mid d_{T_i}(q) < d_{T_j}(q), T_i \in S - \{T_j\}\}$ s'appelle cellule de Voronoï associée à T_i . Le complexe des cellules de Voronoï associées à tous

les sous-ensembles dans S est appelé diagramme de Voronoï de S , noté par $VOD(S)$.

En association avec ce qui a été dit précédemment, le diagramme de Voronoï classique est celui défini sur un ensemble de points sous la métrique euclidienne. *Les diagrammes de Voronoï généralisés* sont les diagrammes de Voronoï différents du diagramme de Voronoï classique selon certains sens. Les diagrammes de Voronoï généralisés sont souvent étudiés selon le sens concret de la généralisation.

Pour un ensemble de germes ponctuels du plan, le diagramme de Voronoï classique a été généralisé avec les métriques L_1 et L_∞ [LEE 80a] [LEE 80b] quand la fonction de distance est définie par (1.2). Si la fonction de distance vérifie (1.3), on a alors *le diagramme de Voronoï d'ordre k* [LEE 82b] [CHAZELLE 87] [ROSENBERGER 88]. Tout point dans une cellule du diagramme de Voronoï d'ordre k est plus proche des k germes que de n'importe quel autre germe. En outre, si à chaque germe est associée une pondération positive et la fonction de distance donnée par (1.4), on a alors *le diagramme de Voronoï pondéré* [AURENHAMMER 84] [ROSENBERGER 88].

Au lieu d'un ensemble de points, lorsque les germes sont des objets géométriques comme par exemple des segments de droite, des polygones, et si la fonction de distance vérifie (1.5), alors le diagramme de Voronoï généralisé obtenu est basé sur les objets du plan. On verra au chapitre 2 qu'une cellule généralisée de Voronoï dans le plan n'est pas un polygone convexe comme dans le cas du diagramme de Voronoï classique. Dans le cadre de cette thèse, nous discutons du diagramme de Voronoï où les germes sont des polygones du plan et la métrique utilisée est la métrique euclidienne. Dans la suite, nous appellerons un tel diagramme de Voronoï tout simplement *diagramme de Voronoï généralisé*.

1.3 Modèles algorithmiques pour le diagramme de Voronoï

Il existe différents modèles algorithmiques pour construire le diagramme de Voronoï. Pour simplifier la présentation des modèles, nous ne considérons ici qu'un ensemble de germes ponctuels, $S = \{p_1, p_2, \dots, p_N\}$. Il faut préciser que les modèles peuvent aussi s'appliquer à un ensemble de germes plus généralisés.

Modèle "Divide-and-conquer"

"Divide-and-conquer" est un modèle algorithmique pour la conception d'un algorithme rapide [BENTLEY 78] [BENTLEY 80]. L'idée essentielle est résumée dans les trois opérations suivantes:

- (1) Opération division : diviser un problème original en plusieurs sous-problèmes ;
- (2) Opération récurrence : résoudre chacun des sous-problèmes avec la même stratégie ;

(3) Opération fusion : fusionner les solutions de tous les sous-problèmes afin d'obtenir la solution du problème original.

Une telle stratégie de "divide-and-conquer" a été largement appliquée à la construction des diagrammes de Voronoï [PREPARATA 88]. L'algorithme est décrit ci-dessous:

Algorithme VORONOI_DAC(S);

Début

1. Si $Card(S)=2$ alors construire le diagramme de Voronoï des deux points;
2. sinon
 - début
 - a. Soit $k=int(N/2)$. Diviser S en S_1 et S_2 où $S_1=\{p_1, p_2, \dots, p_k\}$ et $S_2=\{p_{k+1}, p_{k+2}, \dots, p_N\}$;
 - b. $VOD_1 \leftarrow VORONOI_DAC(S_1)$;
 $VOD_2 \leftarrow VORONOI_DAC(S_2)$;
 - c. fusionner VOD_1 et VOD_2 pour obtenir $VOD(S)$;
 - fin;

Fin;

L'étape cruciale dans l'algorithme est la fusion des deux sous-diagrammes de Voronoï VOD_1 et VOD_2 . En général, le temps de calcul de l'algorithme peut s'exprimer par $T(N)=2T(N/2) + t_m$ où t_m représente le temps de la fusion. Shamos [SHAMOS 78] a montré que t_m peut être linéaire si les germes originaux sont ordonnés de sorte que les deux sous-ensembles divisés soient séparables linéairement. Dans ce cas, le diagramme de Voronoï d'un ensemble de points peut être construit en temps $O(N \log N)$ [PREPARATA 88]. Ceci est une solution optimale. On remarque que le modèle "Divide-and-Conquer" convient bien au *calcul parallèle* dans un environnement de multiprocesseurs [ATALLAH 89] [PREILOWSKI 88][SCHWARZKOPF 89].

Modèle d'incrémentalation

Un ensemble d'objets est dit être traité de façon *incrémentale* si on effectue le calcul étape par étape en introduisant un objet à chaque fois. Notons $S_N=\{p_1, p_2, \dots, p_N\}$. Pour la construction du diagramme de Voronoï, l'algorithme d'incrémentalation est décrit comme suit:

Algorithme INCREMENTATION;

Début

1. Initialisation. calculer $VOD_2=VOD(\{p_1, p_2\})$ pour les deux premiers germes ;
2. pour $k=3$ à N faire
 - début

- a. ajouter p_k à S_{k-1} ;
 - b. modifier localement VOD_{k-1} pour obtenir VOD_k ;
- fin ;

Fin ;

La modification locale de VOD_{k-1} , pour obtenir VOD_k , diagramme de Voronoï des k germes, constitue la clé de l'algorithme.

Pour effectuer la modification locale, il faut calculer un *séparateur* (voir chapitre 5) entre $S_{k-1} = \{p_1, p_2, \dots, p_{k-1}\}$ et $\{p_k\}$. Ici le séparateur est formé des arêtes de Voronoï de la frontière de la nouvelle région de Voronoï du germe p_k sur l'ensemble S_{k-1} complété de p_k , notée $V(p_k, S_{k-1})$. Une fois le séparateur calculé, la modification locale peut alors se faire en trois étapes : (a) créer une nouvelle région de Voronoï associée à p_k , $V(p_k, S_{k-1})$; (b) modifier les régions de VOD_{k-1} par lesquelles passe le séparateur si bien que $V(p_i, S_{k-1} - p_i) \cap V(p_k, S_{k-1}) = \emptyset$ ($i \in \mathbb{I}[1, k-1]$) où $V(p_i, S_{k-1} - p_i)$ désigne une région de Voronoï modifiée; (c) lier $V(p_i, S_{k-1} - p_i)$ et $V(p_k, S_{k-1})$ pour obtenir VOD_k . On remarque que le modèle d'incrémental a généralement une complexité de temps de calcul en $O(n^2)$ dans le cas le plus défavorable, mais en réalité il se comporte en pratique avec une complexité en $O(n)$ [OHYA 84] [SUGIHARA 89]. Mentionnons aussi que le modèle d'incrémental est adapté à l'environnement de *construction dynamique* où on a besoin d'ajouter aléatoirement un germe dans un ensemble de germes existants.

Modèle de "Sweeping"

Une autre méthode intéressante pour construire le diagramme de Voronoï est fondée sur une technique appelée *ligne de balayage* (sweeping). La méthode a été proposée par Fortune en 1986 [FORTUNE 86]. L'algorithme résultant fonctionne en $O(N \log N)$ pour un ensemble de points, un ensemble de segments et un ensemble de points pondérés. L'idée essentielle consiste à transformer le diagramme de Voronoï en telle forme que la technique du "sweeping" puisse s'appliquer. Une ligne de balayage est une ligne horizontale se déplaçant de bas en haut dans le plan. A chaque instant, un état de la ligne de balayage est caractérisé par une séquence d'événements ordonnés de gauche à droite le long de la ligne de balayage. Les événements sont les régions de Voronoï et les arêtes de Voronoï intersectées par la ligne de balayage. Une fois que l'état de la ligne de balayage change, la ligne de balayage doit s'arrêter pour mettre à jour la séquence d'événements ordonnés.

Soit $S = \{p_i \mid i=1, \dots, N\}$ un ensemble de points. Considérons une transformation $*_S : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ définie par :

$$*_S(q) = *_S(q_x, q_y) = (q_x, q_y + \min_{p \in S} d(p, q)) \quad (1.6)$$

On peut voir quelques propriétés de la transformation $*_S$:

(i) $*_S(p_i) = p_i$, pour $p_i \in S$;

(ii) pour un point q de la région de Voronoï originale (non transformée) $V(p)$

$$q^* = *_S(q) = *_p(q),$$

on a :

$$q^*_y \geq p^*_y (=p_y)$$

C'est-à-dire, avec la transformation $*_S$, le point de la région de Voronoï transformée $V^*(p)$ se situe toujours au dessus du germe associé à p .

Soit $VOD^*(S)$ le diagramme de Voronoï transformé. Ces deux dernières propriétés impliquent que : quand la ligne horizontale L balaie le plan de bas en haut, alors (a) la première intersection de L et de $V^*(p_i)$, aura lieu au germe p_i (L entre dans $V^*(p_i)$ d'abord en p_i) ; (b) la dernière intersection de L et de $V^*(p_i)$ aura lieu en un sommet de Voronoï transformé (L quitte $V^*(p_i)$ à l'intersection de deux arêtes de Voronoï transformées). Par conséquent, en utilisant la technique de "sweepline", le diagramme de Voronoï transformé $VOD^*(S)$ peut être construit en $O(N \log N)$. Cependant, la méthode peut fonctionner directement sur les bissectrices non transformées. Ceci est dû au fait que l'intersection (un sommet de Voronoï) de deux bissectrices transformées peut être obtenue par la transformation de l'intersection des deux bissectrices (cf. Déf. 2.6) non transformées. Donc, avec des structures de données appropriées associées à la ligne de balayage L , les arêtes et les sommets du diagramme de Voronoï original peuvent être calculés pour obtenir $VOD(S)$ sans recours de la transformation inverse de $*_S$.

1.4 Travaux antérieurs sur les segments

Rappelons ici quelques travaux antérieurs sur le diagramme de Voronoï d'un ensemble de segments de droite. Le premier travail sur le diagramme de Voronoï généralisé d'un ensemble de segments est la thèse de Drysdale [DRYSDALE 79]. Il avait donné un algorithme avec un temps de calcul en $O(nc^{\sqrt{\log n}})$ où c est une constante positive. Un algorithme simplifié avec une complexité de temps en $O(n \log^2 n)$ a été par la suite amélioré par Lee et Drysdale [LEE 81]. En 1979, Kirkpatrick [KIRKPATRICK 79] avait proposé un algorithme en $O(n \log n)$, mais la technique semble trop compliquée et l'exactitude reste à démontrer [YAP 87]. Yap [YAP 87] a établi un algorithme en détails avec une complexité en $O(n \log n)$ pour un ensemble de n portions de courbe. Indépendamment Fortune a donné un autre algorithme en $O(n \log n)$ en utilisant la technique de "sweepline" (mentionnée précédemment) [FORTUNE 86].

On remarque que tous les algorithmes cités ci-dessus excepté celui de Fortune sont basés sur le modèle de calcul “divide-and-conquer”. La différence entre eux consiste dans la réalisation d’une division d’un ensemble de segments de droite initial. Par exemple, Lee et Drysdale [LEE 81] divisent les segments en S_1 et S_2 tels que S_1 et S_2 sont disjoints et aucun segment n’est coupé en plusieurs morceaux. Par conséquent, la courbe de fusion à chaque étape de la récurrence peut être déconnectée (voir figure 5.1 par exemple). De son côté, Yap [YAP 87] trace dans le plan les droites verticales passant par chaque extrémité de segments donnés. Ces droites verticales partitionnent le plan en “slabs”. L’algorithme de Yap effectue alors récursivement une division des “slabs” en deux moitiés, l’une à gauche et l’autre à droite. Le processus de fusion se fait en deux étapes: la fusion verticale pour former le diagramme de Voronoï dans chacun des “slabs” et la fusion horizontale pour obtenir le diagramme de Voronoï d’objets (portions de segments coupées par “slabs”) dans les deux “slabs” concernés.

1.5 Diagramme de Voronoï de polygones

Quand les segments de droite forment un polygone simple, on a affaire à deux genres de diagrammes de Voronoï généralisés : celui situé à l’intérieur du polygone et celui situé à l’extérieur du polygone. Lee [LEE 82a] a donné un algorithme en $O(n \log n)$ de type “divide-and-conquer” pour le diagramme de Voronoï généralisé de l’intérieur d’un polygone. Cependant, comme nous le verrons au chapitre 3, des contre-exemples mettent en cause son algorithme. Pour le cas d’un ensemble de polygones simples, Srinivasan et al n’ont traité qu’un cas particulier où les polygones forment un domaine polygonal multiple-connecté qui est défini par l’intérieur d’un polygone moins les trous inclus dans ce polygone [SRINIVASAN 87] [MESHKAT 87] .

Cette thèse s’intéresse à quelques problèmes théoriques et à la réalisation du diagramme de Voronoï généralisé dans le cas de la présence de polygones quelconques. Une idée de calcul en *mode hiérarchique* ou *multi-échelles* est appliquée. Le concept de multi-échelles doit être compris au sens où l’on utilise différents niveaux de représentation géométrique à savoir: le point, le segment, la ligne polygonale et la forme polygonale. Pour calculer le diagramme de Voronoï généralisé de polygones quelconques, nous construisons respectivement le diagramme de Voronoï de *l’intérieur* des polygones et celui de *l’extérieur* des polygones. Pour construire le diagramme de Voronoï de l’intérieur ou de l’extérieur d’un polygone, nous fusionnons les diagrammes de Voronoï de *chaînes polygonales* décomposées à partir de la frontière du polygone correspondant. Pour obtenir le diagramme de Voronoï d’une chaîne polygonale, nous décomposons la chaîne polygonale en *éléments* qui sont soit des sommets de la chaîne soit une arête de la chaîne exceptée les sommets associés. D’autre part, pour la construction de multi-échelles, le diagramme de Voronoï généralisé à l’échelle du polygone est obtenu à l’aide de celui à l’échelle d’éléments.

Une telle idée de calcul en multi-échelles nous permet de construire le diagramme de Voronoï généralisé au niveau de polygones, généralisant celui de segments de droite comme dans [YAP 87] [LEE 81]. Comme nous manipulons des polygones, nous traitons un ensemble de segments comme un ensemble de polygones dégénérés. La construction en multi-échelles fournit un moyen pour partitionner et représenter un espace en différents niveaux. Un des exemples se présente lorsque l'on veut analyser des formes polygonales. Dans ce cas, on peut obtenir directement les *axes médians* [BLUM 64] des formes, à partir du diagramme de Voronoï généralisé à l'échelle d'éléments pour l'intérieur des formes. D'autre part, Le SKIZ (*SKEleton by Zones of Influence*) [SERRA 82] d'un ensemble de formes peut être déduit directement du diagramme de Voronoï généralisé à l'échelle de polygones. La squelettisation continue de formes polygonales à partir du diagramme de Voronoï généralisé est détaillée au chapitre 7. Sans doute une telle partition et représentation de l'espace fournit un outil efficace pour la planification d'un robot [TAKAHASHI 89] [LEVEN 87a] [LEVEN 87b] [SCHWARTZ 89] [SCHWARTZ 90].

Les polygones concernés dans cette thèse sont *simples et disjoints*, c'est-à-dire qu'il n'y a pas d'intersection entre deux arêtes non voisines d'un polygone, et qu'il n'y a pas non plus, d'intersection d'arêtes appartenant à des polygones différents. Ces contraintes sont en pratique très naturelles et raisonnables. Sous ces contraintes, on voit qu'un polygone contenant des trous polygonaux [SRINIVASAN 87] est aussi un cas particulier parmi ceux que nous traitons. Pour construire le diagramme de Voronoï de chaque polygone, nous utilisons la stratégie "divide-and-conquer" sur la représentation en *chaînes d'éléments* décomposés à partir de la frontière du polygone. Cependant pour un ensemble de polygones, la stratégie d'incrémentation est appliquée au sens que *l'entité* introduite à chaque étape du calcul est un polygone. Les algorithmes correspondants peuvent s'appliquer pratiquement à un ensemble d'objets hybrides (points, segments, polygones).

Le problème de la construction *semi-dynamique* est aussi discuté, c'est-à-dire qu'à partir du diagramme de Voronoï d'un ensemble d'objets existants, quand un nouvel objet est inséré *aléatoirement* dans l'ensemble, il faut construire un nouveau diagramme de Voronoï correspondant. Trois types d'insertion sont envisagés : insertion d'un point, insertion d'un segment de droite, et insertion d'un polygone convexe.

L'analyse des performances des algorithmes proposés est toujours fondée sur la notion de *la complexité* du temps de calcul *du cas le plus défavorable*. La complexité du cas le plus défavorable est le maximum d'une mesure d'un algorithme donné pour tous les cas possibles des données avec une taille fixée. *La complexité du cas moyen* est une mesure au sens statistique. L'analyse de la complexité du cas moyen est généralement plus compliquée. Bien

que nous n'effectuons pas une telle analyse, la complexité du cas moyen de notre algorithme pour construire le diagramme de Voronoï généralisé de polygones pourrait être attractive.

1.6 Organisation

Cette thèse est organisée en 7 chapitres.

Après cette introduction, des préliminaires sur le diagramme de Voronoï généralisé en la métrique euclidienne sont présentés dans le chapitre 2. Il comporte des notations de base, des définitions, des propriétés, des structures de données pour le diagramme de Voronoï généralisé etc... En outre on décrit des calculs souvent rencontrés dans la construction du diagramme de Voronoï, par exemple, l'intersection entre deux bissectrices, et la localisation d'un point dans le graphe du diagramme de Voronoï généralisé.

Le chapitre 3 détaille le problème de la construction du diagramme de Voronoï d'un seul polygone. A partir du concept de visibilité, une modification de l'algorithme de Lee [LEE 82a] est présentée. En conséquence une version corrigée de cet algorithme est proposée pour construire respectivement le diagramme de Voronoï de l'intérieur d'un polygone et celui de l'extérieur d'un polygone.

Le chapitre 4 concerne l'enveloppe convexe d'un ensemble de polygones quelconques. Malgré que ceci soit un problème indépendant du contexte du diagramme de Voronoï, il est en fait en étroite relation avec le diagramme de Voronoï généralisé. Le concept de polygones d'intersections simples y est introduit. Le chapitre introduit aux méthodes de calcul d'un séparateur.

Dans le chapitre 5, nous montrons comment construire un séparateur entre deux sous-ensemble d'objets. C'est un problème crucial pour tous les algorithmes associés à la construction du diagramme de Voronoï. Notamment, dans le cas où la stratégie d'incrémentatation est appliquée à la construction du diagramme de Voronoï de polygones, nous pouvons *a priori* déterminer la forme d'un séparateur qui est soit fermé soit ouvert selon la situation d'inclusion de l'enveloppe convexe du polygone à insérer dans l'enveloppe convexe de l'ensemble des polygones existants.

En utilisant les résultats des chapitres 2, 3 et 5, nous présentons dans le chapitre 6 les détails d'algorithmes pour construire le diagramme de Voronoï généralisé d'un ensemble de polygones. L'algorithme est de type incrémental. Pour appliquer la propriété de continuité d'un séparateur exploitée dans le chapitre 5, un prétraitement des polygones est nécessaire. Le cas d'un ensemble de segments de droite disjoints y est traité comme celui de polygones dégénérés. De plus, la construction semi-dynamique est mentionnée. Evoquons le cas typique de l'insertion aléatoire d'un objet convexe.

Etant indépendant des autres chapitres, le chapitre 7 traite du problème des *squelettes continus* à l'aide de la notation et de la structure du diagramme de Voronoï généralisé. Nous montrons que les squelettes continus d'objets polygonaux dans le plan continu sont une caractéristique *inhérente* au diagramme de Voronoï généralisé correspondant. Trois types de squelettes continus peuvent être obtenus directement d'après le diagramme de Voronoï généralisé. Ce sont: l'axe médian pour l'intérieur d'un objet, l'exosquelette d'un ensemble d'objets et le SKIZ.

Chapitre 2

Diagramme de Voronoï Généralisé : Préliminaires Géométriques

2.1 Définitions et notations de base

Dans cette partie, nous allons introduire quelques définitions et notations souvent utilisées pour le diagramme de Voronoï généralisé.

Définition 2.1 Notons par $d(p, q)$ la *distance euclidienne* entre un point p et un point q . La distance entre un point p et un ensemble non vide S , notée $d(p, S)$, est définie par:

$$d(p, S) = \min_{q \in S} \{d(p, q)\} \quad (2.1)$$

De façon similaire, la distance euclidienne entre deux ensembles d'objets non vides disjoints S_1, S_2 , notée par $d(S_1, S_2)$, est définie par

$$d(S_1, S_2) = \min_{q_1 \in S_1, q_2 \in S_2} \{d(q_1, q_2)\} \quad (2.2)$$

Définition 2.2 Un *segment de droite fermé* $[a, b]$ est formé de la réunion des deux points d'extrémité a et b , et du segment de droite ouvert (a, b) . On désigne par *élément* un point ou un segment de droite ouvert. Quelquefois, nous désignons un point par *p-élément* et un segment de droite ouvert par *s-élément*.

Définition 2.3 La *projection* d'un point q sur un segment de droite fermé $[a, b]$, notée $P(q, [a, b])$ est la projection orthogonale du point q sur la droite supportant le segment $[a, b]$. Une définition similaire tient pour $P(q, (a, b))$.

Définition 2.4 L'*image* $I(q, [a, b])$ d'un point q sur un segment de droite fermé $[a, b]$ est la suivante:

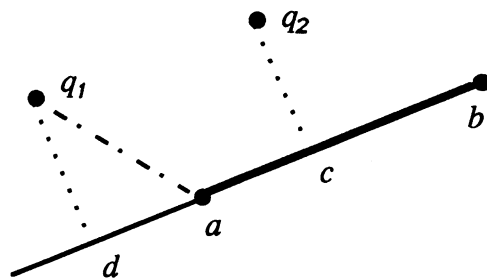
$$I(q, [a, b]) = \begin{cases} P(q, [a, b]) & \text{si } P(q, [a, b]) \text{ est située sur } [a, b] \\ \text{le point extrême (a ou b) le plus proche de la projection sinon} \end{cases} \quad (2.3)$$

D'après (2.1) et (2.3), la distance entre un point q et un segment de droite fermé $[a, b]$ s'obtient selon la formule ci-dessous:

$$d(q, [a, b]) = d(q, I(q, [a, b])) \quad (2.4)$$

C'est-à-dire, la distance entre un point q et un segment $[a, b]$ est la distance entre le point q et l'image de q sur le segment.

La figure 2.1 illustre les définitions précédentes.



$$\begin{aligned} P(q_1, [a, b]) &= a & P(q_2, [a, b]) &= c \\ I(q_1, [a, b]) &= a & I(q_2, [a, b]) &= c \\ d(q_1, [a, b]) &= d(q_1, a) & d(q_2, [a, b]) &= d(q_2, c) \end{aligned}$$

Figure 2.1 Eléments, projections, images et distances

Définition 2.5 Un objet O est un ensemble connexe de points du plan \mathbb{R}^2 . Quand la frontière d'un objet, notée par ∂O , est un polygone fermé, l'objet est alors appelé *objet polygonal*.

Dans le cadre de cette thèse, un objet est toujours supposé comme étant un objet polygonal. Donc la frontière d'un objet est composé d'éléments (points ou segments ouverts). Ainsi, les symboles ζO et εO représentent respectivement l'intérieur et l'extérieur de l'ensemble O . Dans les cas sans ambiguïté, l'intérieur d'un objet est appelé simplement objet. Pour préciser une orientation de parcours de la frontière d'un objet, nous allons utiliser CW pour le sens des aiguilles d'une montre et CCW pour le sens inverse des aiguilles d'une montre.

Définition 2.6 La *bissectrice* $B(e_i, e_j)$ entre deux éléments e_i et e_j est formée de l'ensemble des points équidistants de e_i et de e_j :

$$B(e_i, e_j) = \{ q \mid d(q, e_i) = d(q, e_j) \} \quad (2.5)$$

Si on impose une direction sur les bissectrices, on parle de *bissectrice orientée*. Dans ce qui suit, une orientation est toujours appliquée, c'est-à-dire, $B(e_i, e_j)$ signifie que les éléments e_i et e_j seront répartis respectivement à gauche et à droite de celle-ci.

Dans le cas plus général où on s'intéresse à 2 ensembles d'objets, le concept de bissectrice est généralisé par celui de séparateur:

Définition 2.7 Le *séparateur* de deux objets différents O_1 et O_2 , noté par $Sep(O_1, O_2)$, est défini comme le lieu des points équidistants de O_1 et O_2 où la distance entre un point et un ensemble d'éléments est donnée dans la définition 2.1. A partir de cette définition nous avons:

$$Sep(O_1, O_2) = \{ q \mid \min_{e_i \in O_1} d(q, e_i) = \min_{e_j \in O_2} d(q, e_j) \} \quad (2.6)$$

De même, une orientation est toujours imposée sur un séparateur $Sep(O_1, O_2)$ si bien que O_1 est situé à gauche du séparateur et O_2 est situé à droite de celui-ci quand on parcourt un séparateur selon son orientation.

Définition 2.8 Le *demi-plan* $h(e_i, e_j)$ associé à deux éléments e_i et e_j est l'ensemble des points les plus proches de l'élément e_i vis à vis de l'élément e_j , c'est-à-dire:

$$h(e_i, e_j) = \{ q \mid d(q, e_i) < d(q, e_j) \} \quad (2.7)$$

La fermeture du demi-plan $h(e_i, e_j)$, notée $*h(e_i, e_j)$, est la réunion du demi-plan $h(e_i, e_j)$ et de la bissectrice des deux éléments e_i et e_j .

Le demi-plan associé à deux ensembles d'objets S_1 et S_2 , noté par $H(S_1, S_2)$, peut être défini de façon similaire:

$$H(S_1, S_2) = \{ q \mid \min_{e_i \in S_1} d(q, e_i) < \min_{e_j \in S_2} d(q, e_j) \} \quad (2.8)$$

Lemme 2.1 Soit e_i et e_j deux éléments différents. Alors

$$B(e_i, e_j) = \mathbf{R}^2 - (h(e_i, e_j) \cup h(e_j, e_i)) \quad (2.9)$$

Preuve. Immédiate.

Lemme 2.2 Si S_1 et S_2 désignent deux ensembles d'objets non vides et disjoints, on a:

$$(a). H(S_1, S_2) = \bigcup_{e_i \in S_1, e_j \in S_2} \bigcap h(e_i, e_j) \quad (2.10.a)$$

$$(b). H(S_2, S_1) = \bigcup_{e_j \in S_2, e_i \in S_1} \cap h(e_j, e_i) \quad (2.10.b)$$

$$(c). Sep(S_1, S_2) = \mathbb{R}^2 - (H(S_1, S_2) \cup H(S_2, S_1)) \quad (2.10.c)$$

Preuve.

(a) Supposons que $q \in H(S_1, S_2)$. Selon la définition, q est le point plus proche d'un élément e_i de S_1 que de n'importe quel élément de S_2 , c'est-à-dire, $q \in \cap h(e_i, e_j)$ (pour $e_j \in S_2$). Donc, on a :

$$q \in \bigcup_{e_i \in S_1, e_j \in S_2} \cap h(e_i, e_j)$$

Par un argument similaire, l'inverse peut être montré facilement.

(b). Similaire à (a)

(c). Immédiate.

Ce lemme montre que étant donné deux ensembles d'objets S_1 et S_2 , le plan euclidien \mathbb{R}^2 se divise alors en trois parties: les demi-plans $H(S_1, S_2)$ et $H(S_2, S_1)$, et le séparateur $Sep(S_1, S_2)$.

Définition 2.9 La région de Voronoï d'un élément e_i sur un ensemble S est l'ensemble des points plus proches de e_i que de n'importe quel autre élément de S . La frontière d'une région de Voronoï est composée d'une suite d'arêtes de Voronoï. L'intersection de deux arêtes de Voronoï est appelée *sommet de Voronoï*.

Cette région est notée par $V(e_i, S - e_i)$. Un tel symbole nous permet de préciser que cette région est définie dans le cadre de S et elle se rapporte au sous-ensemble $S - e_i$ de S .

$$\text{Lemme 2.3} \quad V(e_i, S - e_i) = \bigcap_{e_j \in S - e_i} h(e_i, e_j) \quad (2.11)$$

Preuve. Immédiate.

Définition 2.10 La région de Voronoï d'un objet O_i d'un ensemble d'objets S est l'ensemble des points plus proches de O_i que de n'importe quel autre objet dans S . De même, cette région est notée par $V(O_i, S - O_i)$.

$$\text{Corollaire 2.1 } V(O_i, S - O_i) = H(O_i, S - O_i) = \bigcap_{O_j \in S - O_i} H(O_i, O_j) = \bigcup_{e_i \in O_i} \bigcap_{e_j \in S - O_i} h(e_i, e_j) \quad (2.12)$$

Preuve.

$$\begin{aligned} V(O_i, S - O_i) &= \bigcup_{e_i \in O_i} V(e_i, S - O_i) \\ &= \bigcup_{e_i \in O_i} \bigcap_{e_j \in S - O_i} h(e_i, e_j) \end{aligned}$$

Définition 2.11.a *Le diagramme de Voronoï généralisé, $VOD(S)$, d'un ensemble d'objets S est la réunion de toutes les régions de Voronoï associées à chaque objet de l'ensemble.*

Selon cette définition, on a:

$$VOD(S) = \bigcup_{O \in S} V(O, S - O) \quad (2.13.a)$$

De plus l'utilisation d'entités différentes va spécifier le diagramme de Voronoï généralisé défini ci-dessus. En général S est un ensemble dénombrable, c'est-à-dire,

$$S = \{u \mid u \text{ est unité élémentaire}\}$$

Comme cas particuliers, nous avons:

(1) Si u est un point, on a alors *le diagramme de Voronoï ponctuel*. C'est le *diagramme de Voronoï classique* [VORONOI 08], noté par $VOD_p(S)$, et

$$VOD_p(S) = \bigcup_{p \in S} V(p, S - p) \quad (2.13.b)$$

(2) Si u est un élément comme ceux donnés par la définition 2.2, le diagramme de Voronoï obtenu est désigné comme *diagramme de Voronoï d'éléments*, noté par $VOD_e(S)$, et

$$VOD_e(S) = \bigcup_{e \in S} V(e, S - e) \quad (2.13.c)$$

(3) Si u est composé d'objets polygonaux disjoints, nous avons alors le *diagramme de Voronoï d'objets*, noté par $VOD_o(S)$:

$$VOD_o(S) = \bigcup_{o \in S} V(o, S - o) \quad (2.13.d)$$

2.2 Bissectrices et séparateurs

Dans le précédent paragraphe, nous avons défini les bissectrices et les séparateurs. Dans cette partie, nous allons montrer les formes concrètes de bissectrice et de séparateur. Dans un premier temps, nous considérons des configurations d'objets comportant quelques éléments. Les séparateurs correspondants sont simples. Nous reviendrons sur l'étude du séparateur généralisé dans le chapitre 5.

Etant donné 2 éléments e_1 et e_2 on peut identifier trois types de bissectrices en tenant compte de la combinaison de la nature des éléments (point ou segment ouvert). Ces types des bissectrices sont étiquetés par:

— $[pp]$ Si les éléments e_1 et e_2 sont des points, la bissectrice correspondante est alors une droite (cf. figure 2.2.a).

— $[ps]$ Si e_1 est un point et e_2 est un segment de droite ouvert, la bissectrice est alors une courbe composée de trois parties, une partie parabolique (dont le foyer et la directrice sont respectivement le point et le segment) et deux demi-droites déterminées respectivement par le point e_1 et une extrémité de la fermeture du segment de droite ouvert e_2 (cf. figure 2.2.b).

Un cas de dégénérescence de la bissectrice de type $[ps]$ se rencontre quand le point e_1 coïncide avec une extrémité de la fermeture du segment ouvert. Dans ce cas la bissectrice de type $[ps]$ devient une droite passant par le point et orthogonale au segment (cf. figure 2.2.c).

— $[ss]$ Si les éléments e_1 et e_2 sont de type segments ouverts, la bissectrice est alors une courbe qui peut se décomposer au maximum en sept parties qui sont soit des segments de droite, soit des demi-droites, soit des arcs paraboliques. Le foyer et la directrice d'un arc parabolique sont toujours relatifs à l'extrémité de la fermeture d'un segment et à l'autre segment. La figure 2.2.d illustre la bissectrice des deux segments ouverts disjoints et la figure 2.2.e. illustre la bissectrice entre deux segments ouverts dont les segments fermés associés ont une extrémité commune.

Maintenant étudions les configurations de séparateurs entre deux ensembles d'éléments, S_1 et S_2 . Dans le cas qui nous intéresse, le séparateur le plus simple est celui qui apparaît quand S_1 est un point et S_2 un segment de droite fermé. Un tel séparateur peut être composé au maximum de trois parties qui sont identiques à la bissectrice établie pour le point et le segment ouvert correspondant au segment fermé (cf. figure 2.3.a).

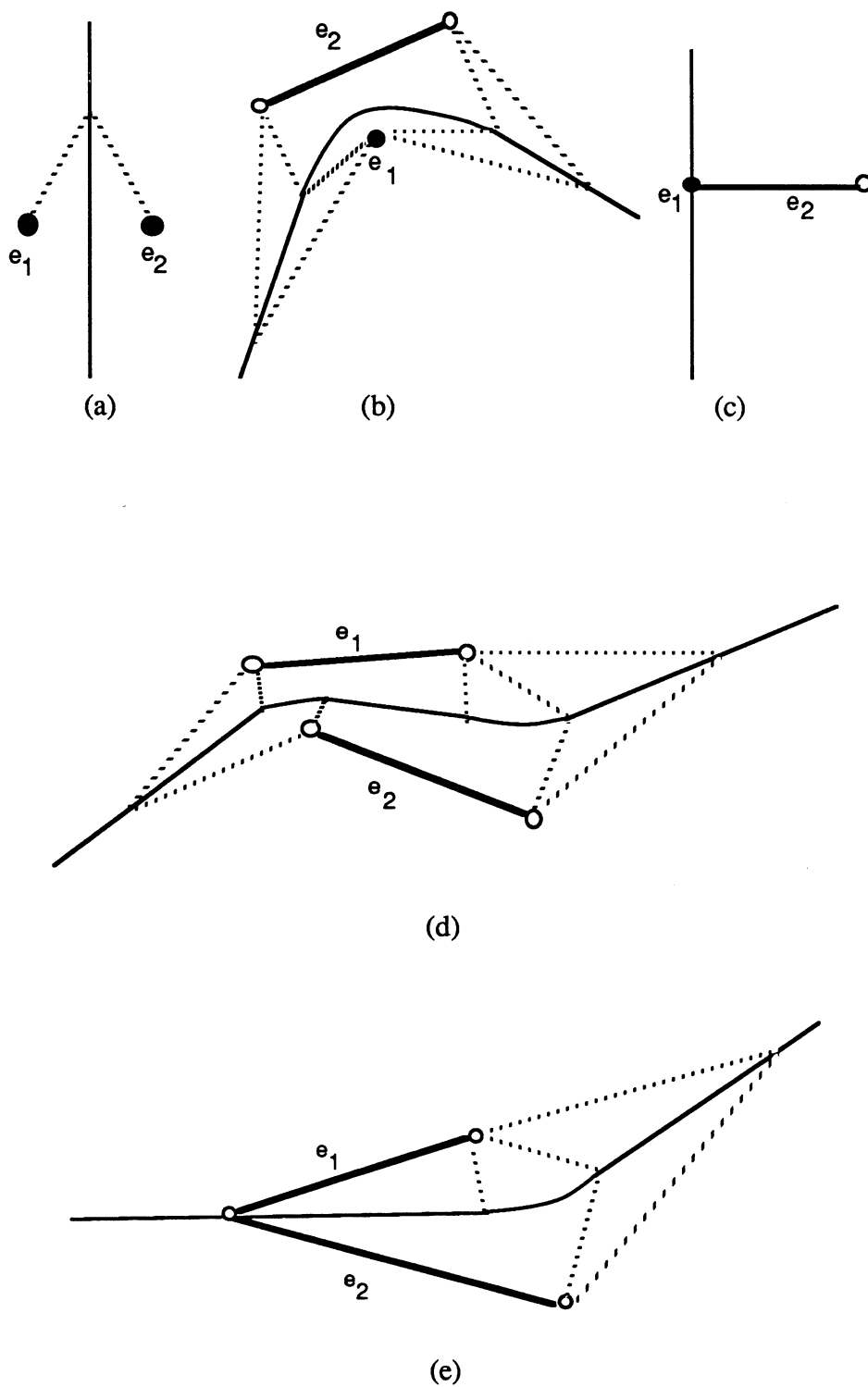


Figure 2.2 Les bissectrices de deux éléments de type [pp] (a); de type [ps] (b); de type [ps] en cas de dégénérescence (c); de type [ss] de deux segments ouverts disjoints (d); et de type [ss] de deux segments ouverts ayant une “extrémité commune” (e)

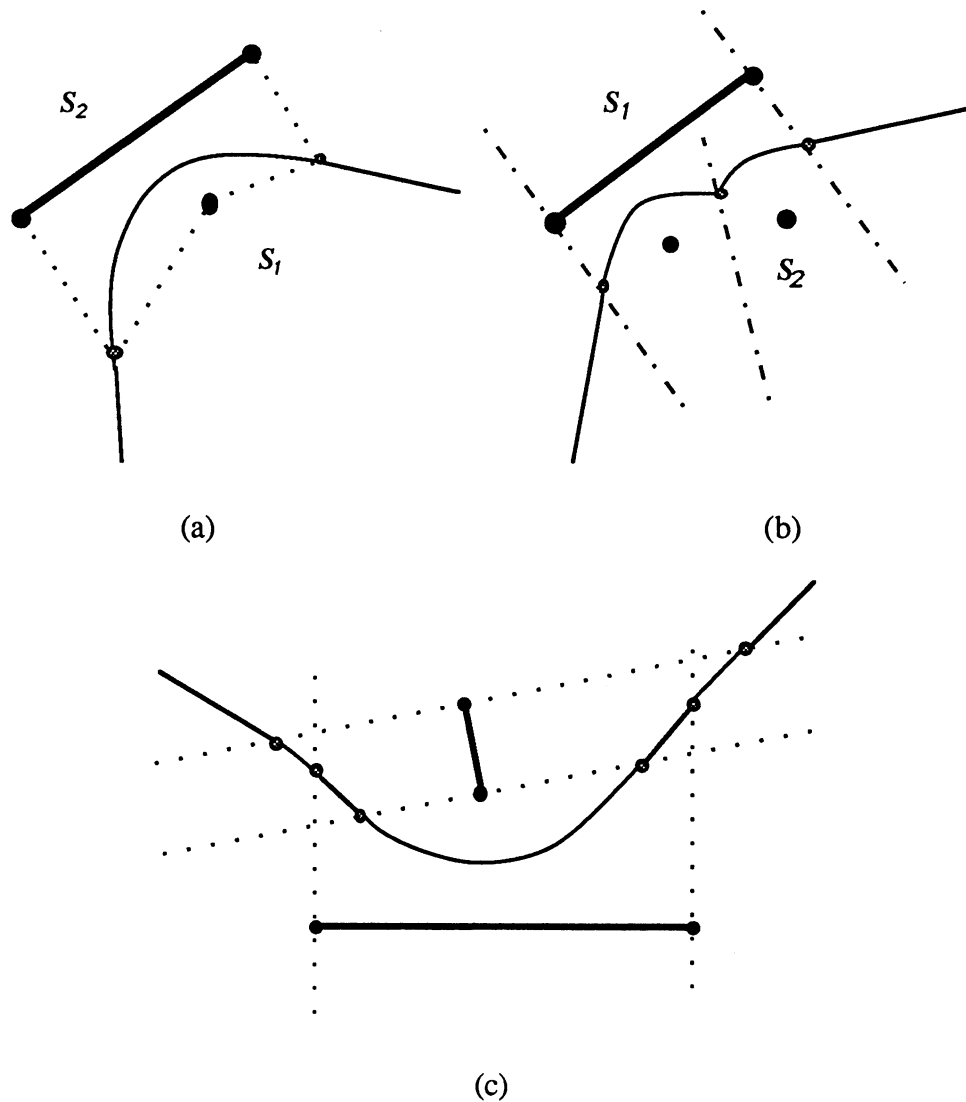


Figure 2.3. Séparateurs entre deux ensembles d'éléments composés
 (a) d'un point et d'un segment fermé; (b) de deux points
 et d'un segment fermé; (c) de deux segments fermés

Le séparateur entre un segment fermé et deux points est montré en figure 2.3.b. La figure 2.3.c représente un exemple de séparateur entre deux segments de droite fermés. Dans ce dernier cas, le séparateur se compose au maximum de sept parties (segments de droite ou arcs paraboliques). Bien que la forme géométrique de la bissectrice illustrée dans la figure 2.2.b est identique à celle du séparateur illustrée dans la figure 2.3.a, la première est issue de la frontière d'un demi-plan $h(e_1, e_2)$ et la seconde est formée de l'intersection des frontières de plusieurs demi-plans $h(e_i, e_j)$ avec $e_i \in S_1$ et $e_j \in S_2$. Donc, ils portent des significations différentes.

Notons que les points du plan peuvent se classer en trois catégories:

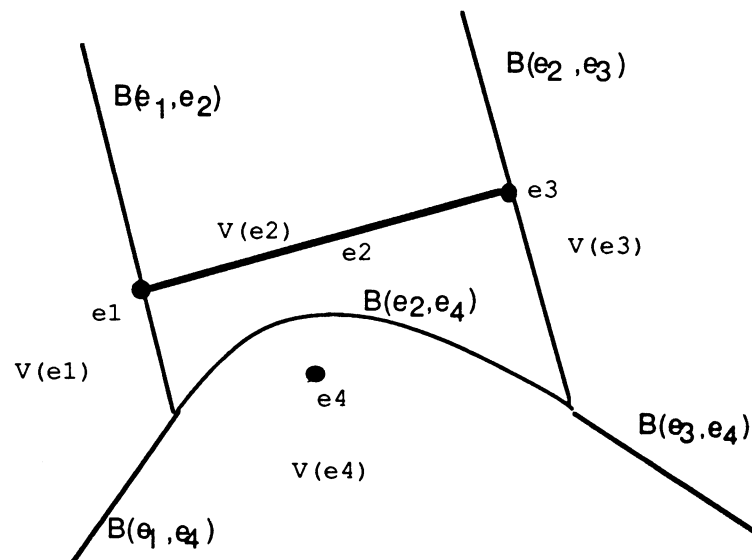
- (a) les points appartenant aux objets donnés;
- (b) ceux situés dans une région de Voronoï et;

(c) ceux appartenant aux séparateurs entre les objets.

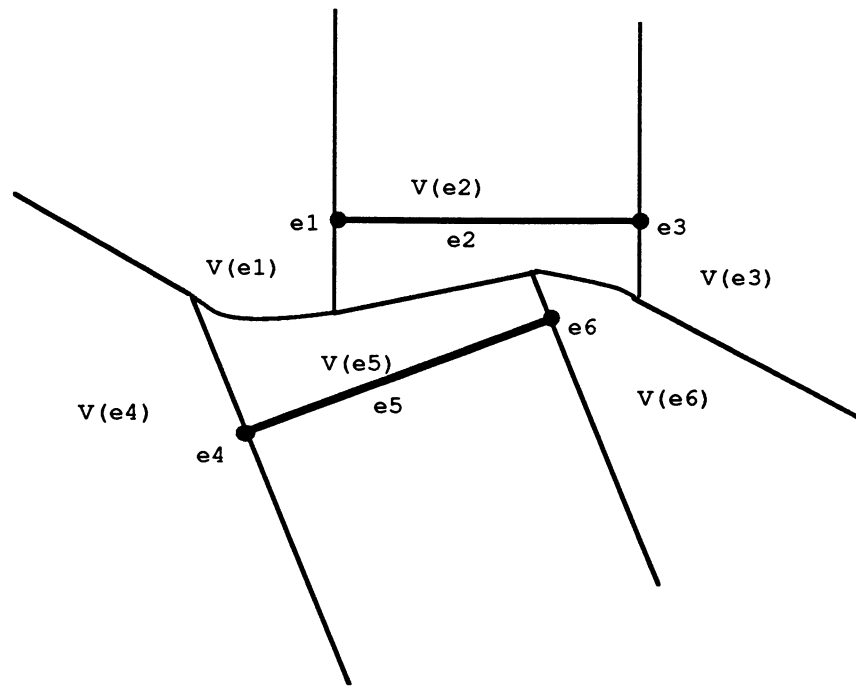
Cette dernière catégorie peut être prise comme la définition pour le diagramme de Voronoï. Cette définition est exprimée ci-dessous:

Définition 2.11.b *Le diagramme de Voronoï généralisé* d'un ensemble d'objets, O , noté $VOD(O)$, est un ensemble de points p tels qu'il existe deux objets O_i et O_j différents, appartenant à O points desquels p est équidistant.

En fait, les points de la catégorie (c) forment le graphe du diagramme de Voronoï. La figure 2.4 montre le diagramme de Voronoï généralisé d'un ensemble d'éléments. On peut remarquer qu'une région du diagramme de Voronoï généralisé n'est plus nécessairement un polygone convexe.



(a)



(b)

Figure 2.4. Deux exemples simples du diagramme de Voronoï généralisé d'ensemble d'éléments S composé: (a) d'un point et d'un segment fermé; et (b) de deux segments fermés. $B(e_i, e_j)$ désigne la bissectrice entre e_i et e_j ; $V(e_i)$ désigne la région de Voronoï généralisée associée à e_i

2.3 Intersection entre des arêtes de Voronoï

L'intersection entre arêtes de Voronoï (ou bissectrices) est un calcul rencontré fréquemment dans le processus de la construction du diagramme de Voronoï. En fait, chaque sommet de Voronoï est une intersection d'arêtes de Voronoï. Nous avons indiqué dans le paragraphe précédent qu'il y a deux types d'arêtes de Voronoï dans le cas où l'ensemble est composé de points et de segments: les segments de droite (droites et demi-droites incluses) et les arcs paraboliques. Donc, le problème de l'intersection entre les bissectrices est ramené à celui de l'intersection de deux segments de droite, ou d'un segment de droite et d'un arc parabolique ou de deux arcs paraboliques.

L'intersection de deux droites peut se calculer simplement en résolvant un système linéaire à deux équations.

L'intersection d'une droite et d'une parabole implique de résoudre une équation d'ordre deux. L'intersection de deux paraboles nécessite le calcul des racines d'une équation d'ordre quatre.

Cependant, nous présentons ci-dessous les lemmes suivants exposant des propriétés intéressantes. Nous montrons que grâce à ces propriétés la solution d'une équation d'ordre quatre se ramène à celle d'une équation d'ordre 2. Ceci simplifie le calcul de l'intersection de deux arêtes de Voronoï de type parabolique.

Lemme 2.4 Les intersections, si elles existent, de deux paraboles sont situées sur la bissectrice des deux directrices correspondantes si ces deux paraboles partagent un foyer commun.

Preuve. Considérons la figure 2.5 . Soit I_1, I_2 les points d'intersection de deux paraboles ayant respectivement les directrices L_1 et L_2 . Soit f le foyer commun des deux arcs de parabole. Nous avons alors:

$$B(L_1, L_2) = \{ p=(x, y) \mid d(p, L_1) = d(p, L_2) \} \quad (2.14)$$

D'après la définition d'une parabole, on a :

$$d(I_j, L_1) = d(I_j, f) \quad (2.15)$$

et

$$d(I_j, L_2) = d(I_j, f) \text{ avec } j=1,2 \quad (2.16)$$

Alors

$$d(I_j, L_1) = d(I_j, L_2) \quad (2.17)$$

Donc

$$I_j \in B(L_1, L_2) \text{ où } j=1,2.$$

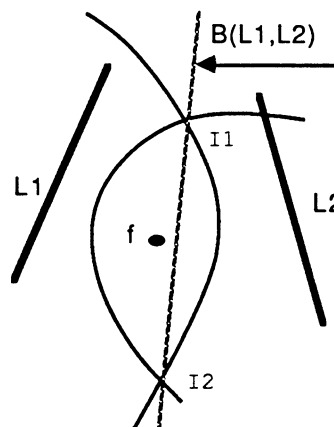


Figure 2.5 Intersections des deux paraboles ayant un foyer commun

Lemme 2.5 Les intersections (si elles existent) de deux paraboles sont situées sur la bissectrice des deux foyers correspondants si les deux paraboles ont une directrice commune.

Preuve. Soit I une intersection des deux paraboles ayant une directrice commune (voir figure 2.6), f_1 et f_2 les foyers de ces deux paraboles respectivement. D'après la définition d'une parabole, on a les relations suivantes:

$$d(I, f_1) = d(I, L) \tag{2.18}$$

et

$$d(I, f_2) = d(I, L) \tag{2.19}$$

Donc

$$d(I, f_1) = d(I, f_2) \tag{2.20}$$

Mais

$$B(f_1, f_2) = \{ p=(x, y) \mid d(p, f_1) = d(p, f_2) \} \tag{2.21}$$

Donc

$$I \in B(f_1, f_2) .$$

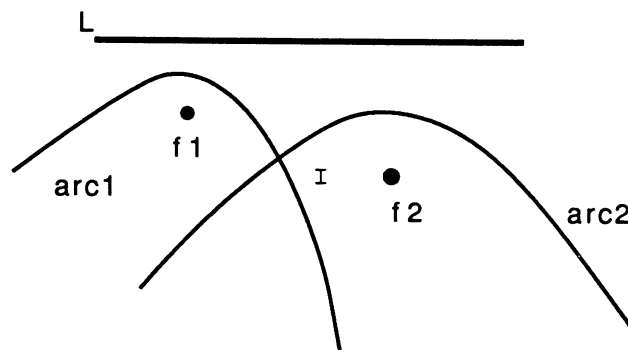


Figure 2.6 Intersections des deux paraboles ayant une directrice commune

Lemme 2.6 S'il existe une intersection entre deux arêtes de Voronoï de type parabolique dans un diagramme de Voronoï, alors ces deux arêtes partagent soit un foyer, soit une directrice commune.

Preuve. Soit C_i et C_j deux arêtes de type arc de parabole ayant une intersection commune I (donc elles sont voisines) dans le diagramme de Voronoï (voir Figure 2.7). Selon la définition du diagramme de Voronoï, la frontière de la région $V(e_i)$ associée à l'élément e_i , noté par $\partial V(e_i)$, est composée de segments de bissectrices entre e_i et d'autres éléments voisins. Donc C_i et C_j , les deux arêtes de type arc de parabole, partagent un élément commun noté e_i , qui est soit un segment de droite (la directrice) soit un point (foyer).

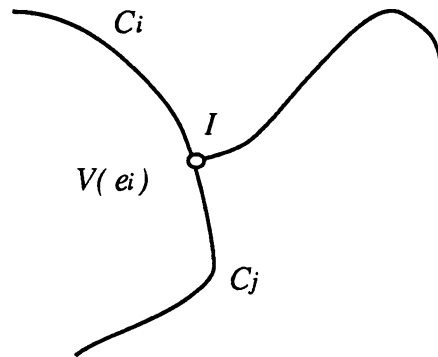


Figure 2.7 illustration pour lemme 2.6

Le lemme 2.6 implique que dans le diagramme de Voronoï généralisé, deux arcs de parabole se coupent seulement dans le cas où les deux paraboles partagent un même foyer ou une directrice commune. Ainsi le lemme 2.6 montre que la détermination de l'intersection de deux arêtes de Voronoï de type parabolique consiste d'abord à résoudre le problème de savoir si ces deux paraboles partagent un élément commun qui est peut-être un point (le foyer) ou un segment de droite (la directrice). Dans le cas contraire, on est sûr qu'il n'existe pas d'intersection entre les deux arêtes de Voronoï de type arc de parabole.

Après avoir jugé si deux arêtes de Voronoï de type arc de parabole ont en commun soit un foyer soit une directrice, nous pouvons alors résoudre le problème d'intersection entre deux arêtes de Voronoï de type arc de parabole selon: (a) soit le lemme 2.4 si elles se partagent un foyer commun; (b) soit le lemme 2.5 si elles se partagent une directrice commune.

2.4 Quelques propriétés du diagramme de Voronoï généralisé

Dans ce qui suit, nous allons montrer quelques propriétés du diagramme de Voronoï généralisé. Nous verrons que certaines propriétés seront une extension naturelle de celles du diagramme de Voronoï classique.

Définition 2.12 Une région polygonale R dans le plan est de type étoile généralisée avec noyau $N, N \subseteq R$, si pour tout point r de R il existe un point n de N tel que le segment fermé $[r, n]$ est situé complètement dans R .

Lemme 2.7 La région de Voronoï généralisé $V(e_i, S)$ est de type étoile généralisée avec e_i comme noyau.

Preuve. Lemme 1 dans [LEE 81].

Lemme 2.8 Le nombre de régions de Voronoï, le nombre d'arêtes de Voronoï et le nombre de sommets de Voronoï dans $VOD(S)$ sont tous en $O(n)$ où n est nombre d'objets de S .

Preuve. Corollaire 1 dans [LEE 81].

Définition 2.13 On dit qu'un objet o_i de l'ensemble d'objets S est situé sur la frontière de l'enveloppe convexe de S si il existe au moins un point $q \in \partial o_i$ tel que q appartienne à la frontière de l'enveloppe convexe.

Lemme 2.9 Un objet o de l'ensemble d'objets S est situé sur la frontière de l'enveloppe convexe de S si et seulement si $V(o, S-o)$ est non bornée.

Preuve. Voir lemme 5.2 et corollaire 5.1

Lemme 2.10 Les deux régions de Voronoï, $V(o_i, S-o_i)$ et $V(o_j, S-o_j)$, sont voisines si et seulement si il existe un point q tel que le cercle centré en q et de rayon $r=d(q, o_i)=d(q, o_j)$ ne contient en son intérieur ou sur sa frontière aucun point d'un autre objet.

Preuve. Similaire au lemme 2 dans [LEE 81].

Pour un ensemble d'objets S , on dit que deux objets o_i et o_j de S sont visibles mutuellement si et seulement si il existe au moins un point p de o_i , et un point q de o_j , tels que le segment (p, q) ne contient aucun point appartenant à S . (voir définitions 3.3 et 3.4 au chapitre 3).

Lemme 2.11 Deux objets invisibles mutuellement ne partagent jamais d'arête de Voronoï commune.

Preuve. Similaire à la preuve du lemme 3.2.

Lemme 2.12 Chaque sommet de Voronoï est une intersection commune de trois arêtes de Voronoï au moins.

Preuve. Supposons qu'un sommet de Voronoï, v , soit formé seulement par l'intersection de deux arêtes de Voronoï. Notons ces deux arêtes par $b(e_i, e_j)$ et $b(e_i, e_k)$ (cf. figure 2.8). Comme $d(e_i, v)=d(e_k, v)$ et $d(e_i, v)=d(e_j, v)$, on a $d(e_j, v) = d(e_k, v)$. v appartient alors à une arête de Voronoï partagée par e_j et e_k , notée $b(e_k, e_j)$. Maintenant, nous voulons montrer qu'une telle arête existe certainement. En supposant que $b(e_k, e_j)$ n'existe pas, il y a alors un élément, noté par e_x , qui empêche la relation adjacente entre $V(e_j)$ et $V(e_k)$. Par conséquent, $b(e_j, e_x)$ va intersecter $b(e_i, e_j)$ en v_x . Puisque l'arête de Voronoï entre v_x et v est partagée par e_i et e_x , donc il est impossible que v soit l'intersection de $b(e_i, e_j)$ et de $b(e_i, e_k)$. Ceci montre qu'une troisième arête de Voronoï connectée à v existe. Dans quel cas, v et v_x sont-ils confondus ? Il y

a deux possibilités : (a) $e_x = e_k$; (b) e_x est situé sur la frontière d'un cercle qui est tangent à la fois à e_i , e_j et e_k . Ce dernier cas montre que le nombre d'arêtes connectées au sommet v peut être supérieur à trois.

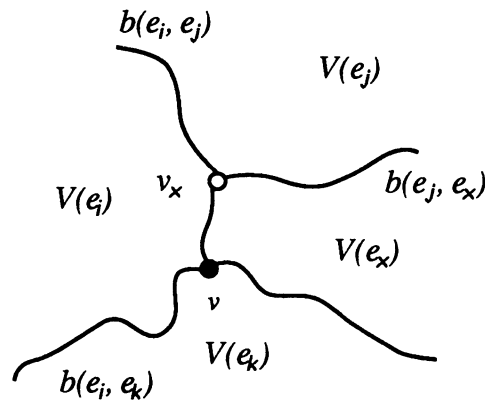


Figure 2. 8. Illustration de lemme 2.12

Corollaire 2.2 A chaque sommet v de Voronoï d'un ensemble d'objets S , il existe un cercle $C(v)$ centré en v tel que $C(v)$ soit tangent à au moins trois objets de S .

Preuve. Immédiate à partir des lemmes 2.10 et 2.12.

Corollaire 2.3 $C(v)$ ne contient, ni intersecte aucun objet de S .

Preuve. Supposons que $C(v)$ soit déterminé par O_1, O_2 et O_3 . Nous avons:

$$d(v, O_1) = d(v, O_2) = d(v, O_3) \tag{2.22}$$

Si un autre objet, dit O_4 , intersecte $C(v)$, il existe alors un point q de O_4 tel que

$$d(v, q) = d(v, O_4) \tag{2.23}$$

ainsi que

$$d(v, q) < d(v, O_1)$$

$$d(v, q) < d(v, O_2)$$

et

$$d(v, q) < d(v, O_3) \tag{2.24}$$

D'où et d'après la définition du diagramme de Voronoï généralisé,

$$v \in V(O_4, S - O_4)$$

Ceci est en contradiction avec le fait que v est un sommet de Voronoï.

Lemme 2.13 Soit S_1 et S_2 deux ensembles d'objets. $V(O_1, S_1 - O_1)$ et $V(O_2, S_2 - O_2)$ partagent une arête de Voronoï commune, notée par $b(O_1, O_2)$, si et seulement si $Sep(S_1, S_2) \cap V(O_1, S_1 - O_1) \cap V(O_2, S_2 - O_2) \neq \emptyset$. De plus, on a : $b(O_1, O_2) = Sep(S_1, S_2) \cap V(O_1, S_1 - O_1) \cap V(O_2, S_2 - O_2)$.

Preuve. Notons $b_{1,2} = Sep(S_1, S_2) \cap V(O_1, S_1-O_1) \cap V(O_2, S_2-O_2)$.

(a). Supposons que $b_{1,2}$ est vide. Dans ce cas-ci, nous avons deux possibilités:

- i) $Sep(S_1, S_2)$ est dans une seule région de $V(O_1, S_1-O_1)$ et $V(O_2, S_2-O_2)$ (figure 2.9.a);
- ii) $Sep(S_1, S_2)$ n'est ni dans $V(O_1, S_1-O_1)$, ni dans $V(O_2, S_2-O_2)$ (figure 2.9.b).

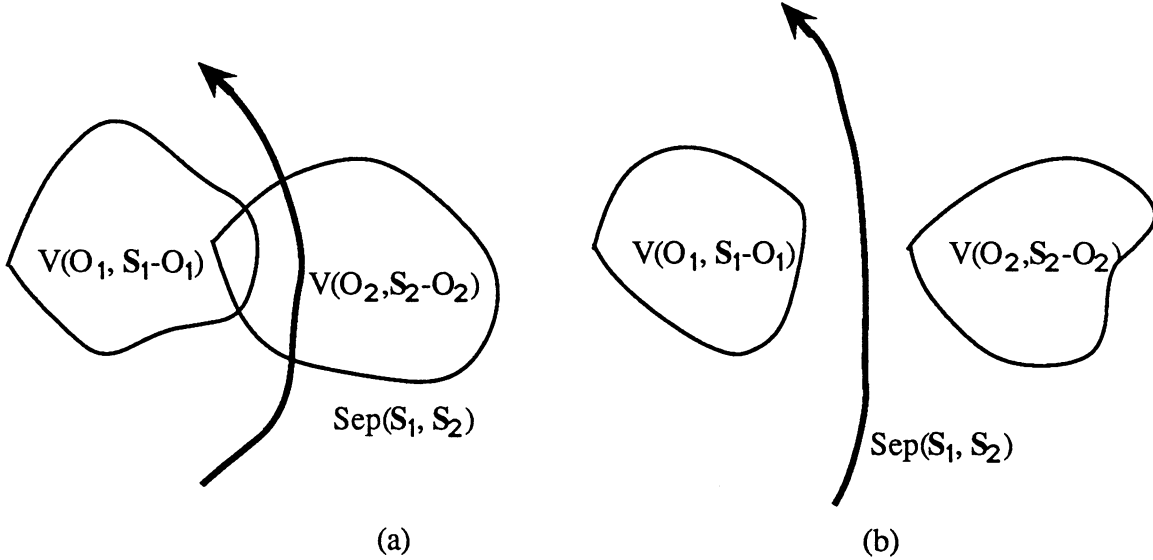


Figure 2.9 $Sep(S_1, S_2) \cap V(O_1, S_1-O_1) \cap V(O_2, S_2-O_2) \neq \emptyset$,
alors $V(O_1, S_1-O_1)$ et $V(O_2, S_2-O_2)$ partagent une arête de Voronoï

Nous savons que tout point q appartenant à $b(O_1, O_2)$ doit vérifier les relations suivantes:

$$d(q, O_1) = d(q, O_2) \tag{2.25}$$

et

$$d(q, O_1) = \min_{O_i \in S_1} d(q, O_i) \tag{2.26}$$

et

$$d(q, O_2) = \min_{O_j \in S_2} d(q, O_j) \tag{2.27}$$

Cependant, il y a certainement dans le premier cas (la figure 2.9.a) une région $V(O_x, S_1-O_x)$ avec $x \neq 1$ telle que $d(q, O_x) < d(q, O_1)$. De même, dans le second cas (figure 2.9.b), il y a deux régions $V(O_L, S_1-O_L)$ et $V(O_R, S_2-O_R)$ avec $L \neq 1$ et $R \neq 2$ telles que $d(q, O_L) < d(q, O_1)$ ainsi que $d(q, O_R) < d(q, O_2)$. Donc, dans tous les cas, $b(O_1, O_2)$ n'existe pas.

(b) Maintenant, nous devons montrer que $b_{1,2} \neq \emptyset$ et que $b(O_1, O_2) = b_{1,2}$. Soit $q \in b(O_1, O_2)$. On a immédiatement:

$$q \in Sep(S_1, S_2)$$

$$d(q, O_1) = d(q, O_2) \quad (2.28)$$

et

$$d(q, O_1) = \min_{O_i \in S_1} d(q, O_i) \quad (2.29)$$

et

$$d(q, O_2) = \min_{O_j \in S_2} d(q, O_j) \quad (2.30)$$

Ceci implique:

$$q \in V(O_1, S_1 - O_1)$$

et

$$q \in V(O_2, S_2 - O_2)$$

Donc:

$$q \in b_{1,2}$$

Comme $Sep(S_1, S_2)$ est non vide, alors $b_{1,2} \neq \emptyset$. Maintenant, supposons que $q \in b_{1,2}$. Il est très facile de montrer que $q \in b(O_1, O_2)$ aussi. Par conséquent, $b_{1,2} = b(O_1, O_2)$.

2.5 Structure de graphe du diagramme de Voronoï généralisé

Afin de représenter le *graphe* du diagramme de Voronoï généralisé précédemment défini, nous devons choisir une *structure de données* pour mémoriser les informations géométriques et topologiques. En fait, le choix de structure de données est une étape cruciale dans la conception d'un algorithme géométrique. Par exemple, la complexité d'un algorithme dépend de la structure de données utilisé dans l'algorithme. Il existe trois *unités élémentaires géométriques* qui peuvent représenter un diagramme de Voronoï: *régions*, *arêtes* et *sommets* de Voronoï. Chaque région est bornée par une chaîne d'arêtes, et chaque arête est partagée par exactement deux régions. Chaque arête est limitée par deux sommets (possibles à l'infini), et chaque sommet peut être partagé par plusieurs arêtes. Pour préciser toutes les informations géométriques et les relations topologiques dans le diagramme de Voronoï généralisé, nous allons utiliser la structure de données décrite ci-dessous.

(1) Chaque sommet de Voronoï est décrit par : ses *coordonnées* ;

(2) Chaque arête de Voronoï orientée est décrite par :

(a) sa *forme*: DROITE ou PARABOLE ;

(b) son *type*: FERMEE ou OUVERTE ou OUVERTE_A ou OUVERTE_DE ;

(c) son *sommet initial* et son *sommet terminal* ;

- (d) sa *région gauche* et sa *région droite* qui sont étiquetées par les éléments associés qui sont soit des points soit des segments de droite ;
- (e) un *pointeur* vers son *arête précédente* (associée à son sommet initial) et un autre pointeur vers son *arête suivante* (associée à son sommet terminal) ;

(3) Chaque région de Voronoï est décrite par :

- (a) l'*élément associé*: POINT ou SEGMENT ;
- (b) un *pointeur* vers la chaîne d'arêtes de Voronoï de la frontière de la région qui prend un sens CW (ClockWise) comme orientation positive ;

Nous étiquetons une région de Voronoï par l'élément associé et sa frontière est représentée comme une liste doublement chaînée d'arêtes. Du fait qu'une arête de Voronoï est constituée soit d'un segment de droite soit d'une droite soit d'un arc parabolique, nous utilisons deux champs particuliers pour mémoriser les données géométriques : i) le champ *forme* qui prend l'une des deux valeurs *DROITE* et *PARABOLE* ; ii) le champ *type* qui prend l'une des quatre valeurs, *FERMEE* représentant une arête avec deux sommets à distance finie, *OUVERTE* représentant une arête avec deux sommets à distance infinie, *OUVERTE_A* représentant une arête avec un sommet initial à distance finie et l'autre sommet terminal à distance infinie, *OUVERTE_DE* représentant une arête avec un sommet initial à distance infinie et l'autre sommet terminal à distance finie. Soit *EDGE* une arête courante. L'arête suivante de *EDGE* dans la liste est définie comme l'arête qui précède quand on part de *EDGE* en sens de la frontière de la région. Une telle abstraction de données est essentiellement équivalente à la structure *DCEL* (*Doubly-Connected-Edge-List*) [PREPARATA 88].

2.6 Localisation d'un point dans une région de Voronoï généralisée

Le problème de la *localisation d'un point* dans une subdivision plane est un des problèmes fondamentaux en géométrie algorithmique. Il est formellement décrit comme suit : Etant donné un graphe plan G composé de droites avec n sommets et un point quelconque q , déterminer quelle région de la subdivision introduite par G contient q . Pour ce problème, on suppose toujours que l'on s'est donné un *graphe plan de droites* (ou PSLG en anglais). Plusieurs méthodes ont été développées :

- (a) la méthode de "Slab" par Shamos [SHAMOS 75] ;
- (b) la méthode de la "Chaîne" par Lee-Preprata [LEE 77] ;
- (c) la méthode du "trapezoïde" par Preparata [PREPARATA 81] ;

- (d) la méthode de la “*triangulation refinement*” par Kirpatrick [KIRPATRICK 83] ;
- (e) la méthode du “*bucket*” par Edahiro-Kokubo-Asano [EDAHIRO 84] ;
- etc...

Remarquons que toutes les méthodes de localisation de point sont essentiellement issues d’une idée fondamentale: créer de nouveaux objets géométriques afin de permettre d’effectuer une recherche dichotomique.

Ici, nous ne voulons pas détailler toutes les méthodes ci-dessus. Par contre, nous nous intéressons au problème de *localisation d’un point dans un diagramme de Voronoï généralisé* au lieu d’un graphe plan de droites.

La différence entre ce nouveau problème et le précédent concerne le fait que le diagramme de Voronoï généralisé comprend en plus des portions vis-à-vis du PSLG. En fait, la localisation d’un point dans le diagramme de Voronoï est toujours présentée quand on calcule le diagramme de Voronoï lui-même.

Supposons que nous ayons obtenu un diagramme de Voronoï $VOD(S)$ à un instant donné et qu’un point z soit donné. Remarquons que le diagramme de Voronoï est mémorisé selon la structure de données similaire à DCEL. C’est-à-dire, la frontière de chaque région de Voronoï est représentée par un cycle d’arêtes doublement chaînées en orientation CW. Dans le cas où les germes seraient des points, les régions de Voronoï sont convexes. Alors la localisation du point z dans le diagramme de Voronoï ponctuel peut être résolue selon la solution d’inclusion d’un point dans une partition formée de polygones convexes. Pour un polygone convexe avec N sommets, l’inclusion d’un point peut être déterminée en un temps $O(\log N)$ avec un temps de prétraitement en $O(N)$ [PREPARATA 88]. L’algorithme correspondant [PREPARATA 88] est donné ci-après.

Algorithme CONVEX_INCLUSION ;

entrée: la liste doublement chaînée d’un polygone convexe $P = \langle p_1, p_2, \dots, p_N \rangle$;
le point z ;

sortie: *INC-CON* {boolean};

Début

1. Soit q un point quelconque intérieur à P . Déterminer par recherche dichotomique un secteur dans lequel z est situé. Le point z est situé dans le secteur défini par deux rayons passant respectivement par $[q, p_i]$ et $[q, p_{i+1}]$ si et seulement si l’angle (zqp_{i+1}) et l’angle (zqp_i) ont des orientations inverses ;

2. Si p_i et p_{i+1} sont trouvés, $INC-CON=vrai$ si et seulement si $(p_i p_{i+1} z)$ forme un cycle dans le sens des aiguilles d'une montre ;

Fin;

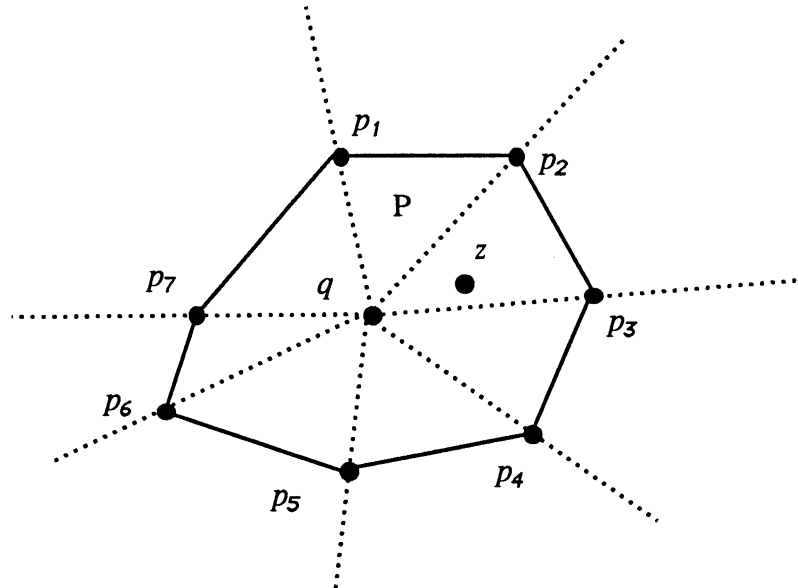


Figure 2.10. Division d'un polygone convexe en secteurs

On peut remarquer que du fait que toutes les arêtes d'une région de Voronoï sont orientées, une autre méthode pour l'inclusion d'un point dans une région de Voronoï ponctuel se ramène à vérifier si le point en question est situé d'un même côté vis-à-vis de toutes les arêtes de la région. Ceci peut être fait en N fois dans le cas le plus défavorable mais sans prétraitement.

A présent, nous allons exposer notre méthode pour localiser un point dans le diagramme de Voronoï généralisé.

Définition 2.14 Une région de Voronoï généralisée s'appelle *s-région* si l'élément associé est un segment de droite ouvert. De même, une région de Voronoï s'appelle *p-région* si l'élément associé est un point.

Rappelons qu'une région de Voronoï $V(e, S)$ généralisée est de type étoile généralisée avec e comme noyau (lemme 2.7). Par conséquent, le segment $[z, P(z, e)]$ où $z \in \partial V(e, S)$ et $P(z, e)$ est la projection de z sur e , est situé entièrement dans $V(e, S)$ (Voir définition 2.12). Quand on projette les points des arêtes de la frontière d'une région de Voronoï sur l'élément associé, on peut trouver quelques propriétés intéressantes.

Définition 2.15 Une chaîne $C = \langle v_1, v_2, \dots, v_k \rangle$ est une suite de portions de courbe formée de l'ensemble de sommets $\{v_1, v_2, \dots, v_k\}$ et de l'ensemble d'arêtes $\{(v_i, v_{i+1}) : i = 1, \dots, k-1\}$. Une chaîne C est appelée *monotone* par rapport à une droite l si toute demi-droite orthogonale à l issue d'un point de l intersecte C en un seul point.

Remarquons que l'arête (v_i, v_{i+1}) dans cette définition est une portion de courbe générale. Quand (v_i, v_{i+1}) est un segment de droite, on a alors une *chaîne polygonale*. La figure 2.11 montre une chaîne polygonale monotone. Les projections orthogonales $\{l(v_1), l(v_2), \dots, l(v_k)\}$ des sommets de C sur l sont ordonnées le long de l .

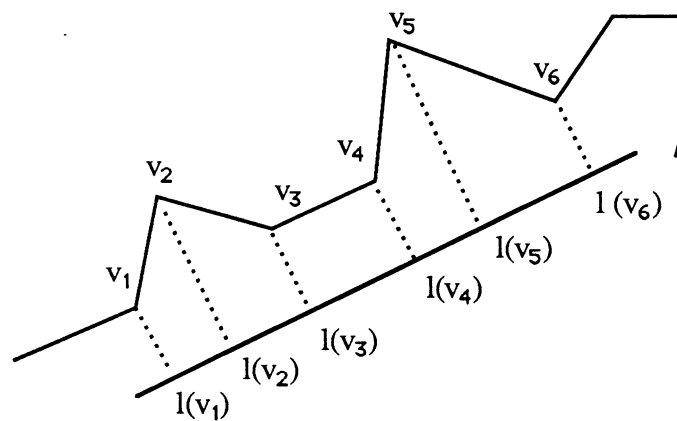


Figure 2.11. Chaîne polygonale monotone par rapport à la droite l

Définition 2.16 Une région de Voronoï est appelée *monotone* si sa frontière peut être décomposée en chaînes monotones par rapport à une même ligne de droite.

La figure 2.12 montre un exemple de la chaîne de la frontière d'une s -région de Voronoï.

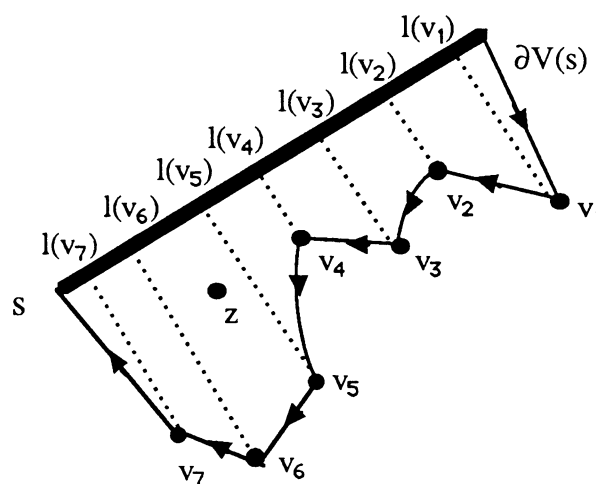


Figure 2.12 La chaîne monotone d'une s -région de Voronoï

Corollaire 2.4 Une s -région de Voronoï est monotone par rapport au segment associé.

Preuve. Soit e un segment. Nous décomposons la frontière $\partial V(e)$ de $V(e)$ en deux chaînes C_l et C_r telles que C_l et C_r sont situées respectivement à gauche de e et à droite de e (suivant l'orientation de e). Nous devons démontrer que C_l et C_r sont monotones par rapport à e . Nous le démontrons par contradiction. Supposons que C_l ne soit pas monotone. D'après la définition 2.15, une droite l orthogonale à e va intersecter C_l en plus d'un point. Soit t_1 et t_2 deux intersections entre C_l et l (voir figure 2.13). Il est impossible de trouver un point $q \in e$ tel que le segment $[t_2, q]$ est situé dans $V(e)$. Ceci contredit la propriété que $V(e)$ est de type étoile généralisée avec e comme noyau (cf. lemme 2.7). La preuve pour C_r est identique à celle pour C_l .

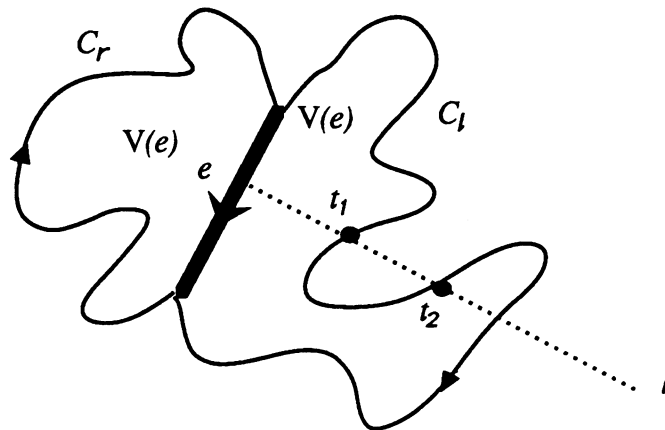


Figure 2.13 la frontière d'une *s-région* de Voronoï est monotone

La méthode pour localiser un point dans une *s-région* est similaire à la méthode de chaîne pour localiser un point dans un graphe PSLG par Lee-Preprata [LEE 77]. Pour une *s-région* $V(l, S)$ donnée et un point quelconque z , nous décomposons $\partial V(l, S)$ en deux chaînes monotones C_l et C_r ainsi que cela a été mentionné dans la démonstration du corollaire 2.4. Ensuite pour chacune des chaînes C_l et C_r nous projetons tous les sommets de Voronoï dans l . Il est clair que la projection de z sur l peut être localisée dans un intervalle unique $(l(v_i), l(v_{i+1}))$ par recherche dichotomique. Enfin, un seul examen peut déterminer de quel côté de l'arête de Voronoï se situe le point z . Donc, la localisation d'un point dans une *s-région* est effectuée en temps $O(\log k)$ où k est le nombre d'arêtes de Voronoï de la région. L'algorithme est décrit ci-dessous.

Algorithme S_REGION_INCLUSION;

entrée: la liste doublement chaînée de la frontière d'une *s-région* de Voronoï $V(l, S)$;
un point quelconque z ;

sortie: *INC-S-REGION*; {boolean}

Début

1. Décomposer $\partial V(l, S)$ en deux chaînes monotones $^1 C_l = \langle \alpha_{l1}, \alpha_{l2}, \dots, \alpha_{lu} \rangle$ et $C_r = \langle \alpha_{r1}, \alpha_{r2}, \dots, \alpha_{rv} \rangle$ où $\alpha_{\beta i}$ est une arête de Voronoï et $\alpha_{\beta i+1}$ est l'arête suivante de $\alpha_{\beta i}$; faire la projection de z sur l , notée par $l(z)$ {initialisation};
 2. Si z est à gauche de l alors faire (voir la figure 2.12)
 - début
 - a. pour $i=1$ à u faire la projection du sommet initial de $\alpha_{\beta i}$ sur l , notée par $l(i)$;
 - b. localiser $l(z)$ dans un intervalle $(l(i), l(i+1))$ par recherche dichotomique;
 - c. $INC-S-REGION =$ vrai si z est situé à droite de l'arête de Voronoï orienté $\alpha_{\beta i}$;
 - fin;
 - sinon
 - si z est à droite de l alors faire
 - début
 - a. pour $i=1$ à v faire la projection du sommet initial de $\alpha_{\beta i}$ sur l , notée par $l(i)$;
 - b. localiser $l(z)$ dans un intervalle $(l(i), l(i+1))$ par recherche dichotomique;
 - c. $INC-S-REGION =$ vrai si z est situé à droite de l'arête de Voronoï orienté $\alpha_{\beta i}$;
 - fin;
- Fin;

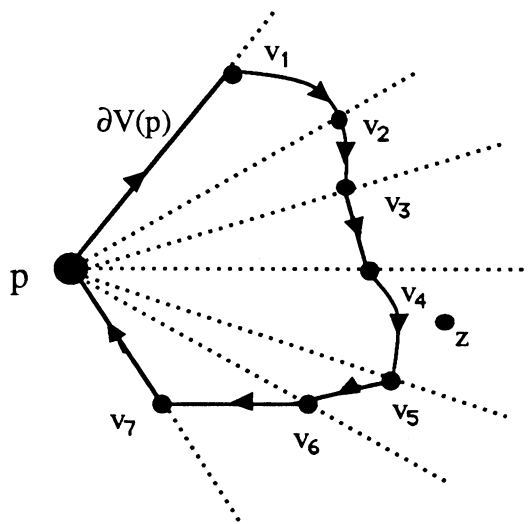


Figure 2.14. Localisation d'un point dans une p -région de Voronoï

¹ Quand nous construisons le diagramme de Voronoï généralisé pour l'intérieur ou à l'extérieur de polygones (voir le chapitre 3 et le chapitre 6), une s -région de Voronoï sera composée d'une seule chaîne monotone. Donc, la décomposition de la frontière d'une s -région en deux chaînes monotones n'est pas nécessaire.

Pour une p -région, une propriété similaire à celle d'une s -région existe.

Corollaire 2.5. Une p -région de Voronoï est monotone par rapport au point associé.

Preuve. Similaire au corollaire 2.4.

Donc, quand nous projetons les sommets de Voronoï d'une p -région sur l'élément p , la région sera divisée en m secteurs $(v_i p v_{i+1})$ par les lignes de projections. On peut trouver le secteur (s'il existe) dans lequel se trouve le point z par recherche dichotomique car les droites de projections seront produites selon un ordre angulaire. Une fois que nous avons déterminé le secteur contenant le point en question, le problème de la localisation d'un point dans une p -région peut être résolu en déterminant si z est situé à droite de l'arête de Voronoï comprise dans le secteur (voir figure 2.14). Evidemment, un tel algorithme pour localiser un point dans une p -région est en principe similaire à l'algorithme CONVEX_INCLUSION. Ceci peut se faire en $O(\log m)$ où m est le nombre d'arêtes de Voronoï d'une p -région.

En résumé, nous avons donc:

Lemme 2.14. Etant donné une région de Voronoï généralisée et un point quelconque z , la détermination du point z dans la région peut se faire en $O(\log n)$ avec un temps $O(n)$ de prétraitement où n est le nombre d'arêtes de Voronoï de cette région.

Chapitre 3

Diagramme de Voronoï Généralisé d'un Polygone Basé sur les Eléments

3.1 Introduction

Dans ce chapitre, nous allons nous intéresser au problème du diagramme de Voronoï d'un polygone au sens où les sommets et les arêtes du polygone désigneront les *germes*. C'est-à-dire, plus précisément, nous allons considérer tous les points de la frontière d'un polygone simple. Soit $\zeta\Omega$ le domaine intérieur du polygone Ω , $\varepsilon\Omega$ le domaine extérieur du polygone Ω et $\partial\Omega$ la frontière du polygone Ω . Nous avons alors une partition suivante du plan \mathbb{R}^2 :

$$\mathbb{R}^2 = (\zeta\Omega) \cup (\partial\Omega) \cup (\varepsilon\Omega) \quad (3. 1)$$

avec $\zeta\Omega \cap \varepsilon\Omega = \emptyset$.

Rappelons que le diagramme de Voronoï, noté $VOD_e(S)$, d'un ensemble S d'éléments, est une partition du plan en régions de Voronoï telle que tout point de chaque région est plus proche de l'élément associé que de n'importe quel autre élément de S . De plus, si nous nous restreignons à l'intérieur d'un polygone Ω , nous avons alors un diagramme de Voronoï borné dans l'intérieur du polygone, pouvant être défini par $VOD_e(\Omega) \cap \zeta\Omega$. Par contre, si nous ne nous intéressons qu'à l'extérieur du polygone, nous allons avoir un diagramme de Voronoï pour l'extérieur du polygone, pouvant être défini par $VOD_e(\Omega) \cap \varepsilon\Omega$. Ainsi le problème de la construction du diagramme de Voronoï généralisé d'éléments constituant la frontière d'un polygone peut se diviser en deux sous-problèmes suivants :

Problème INVOD : Trouver une partition de Voronoï pour $\zeta\Omega:VOD_e(\Omega) \cap \zeta\Omega$;

Problème EXVOD : Trouver une partition de Voronoï pour $\varepsilon\Omega:VOD_e(\Omega) \cap \varepsilon\Omega$.

Nous divisons le problème de la construction de $VOD_e(\Omega)$ en deux sous-problèmes, non seulement du fait que la réunion des solutions de ces deux sous-problèmes fournit directement $VOD_e(\Omega)$, mais aussi du fait que chacun de ces deux sous-problèmes possède un intérêt en lui-même, Détaillons les problèmes dans les paragraphes suivants.

3.2 Représentation et décomposition de la frontière d'un polygone

3.2.1 Représentation basée sur les sommets

Définition 3.1 Un *polygone simple* G de N sommets, noté par $G = (q_0, q_1, \dots, q_{N-1})$, est une suite de N points tels que $[q_i, q_{i+1}]$ soit un segment de droite fermé pour $i=0, 1, \dots, N-1$ (on identifie q_N à q_0), appelé *arête* et tel qu'il n'y ait pas d'intersection entre deux arêtes non-consécutives.

Un polygone simple G est une figure plane fermée composée d'arêtes. Il divise le plan \mathbf{R}^2 en deux parties, l'intérieur et l'extérieur. Dans ce qui suit, nous parlerons en terme de polygone pour simplification d'écriture.

De plus, nous rappelons deux notations identifiant le sens de parcours de la frontière du polygone:

[CW] : le sens des aiguilles d'une montre (ClockWise).

[CCW] : le sens inverse des aiguilles d'une montre (CounterClockWise).

L'introduction de la notion de sens permet de distinguer l'intérieur et l'extérieur d'un polygone G de telle façon que l'intérieur de G se trouve toujours à gauche de la frontière quand le polygone est parcouru en sens inverse des aiguilles d'une montre (CCW).

Etant données les coordonnées de N sommets d'un polygone, une représentation simple et naturelle du polygone est la liste doublement chaînée des sommets illustrée sur la figure 3.1.

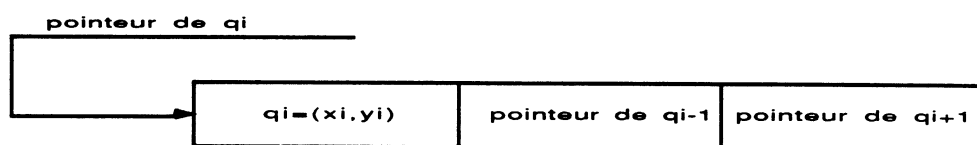


Figure 3.1. représentation d'un polygone par une liste doublement chaînée

3.2.2 Décomposition basée sur les chaînes de sommets

Pour trois sommets consécutifs q_{i-1} , q_i et q_{i+1} , le sommet q_i est appelé *convexe* si l'angle $\angle q_{i-1} q_i q_{i+1}$ est de valeur inférieure à π quand il est mesuré selon le sens CW (cf. figure 3.2) et *concave* autrement.

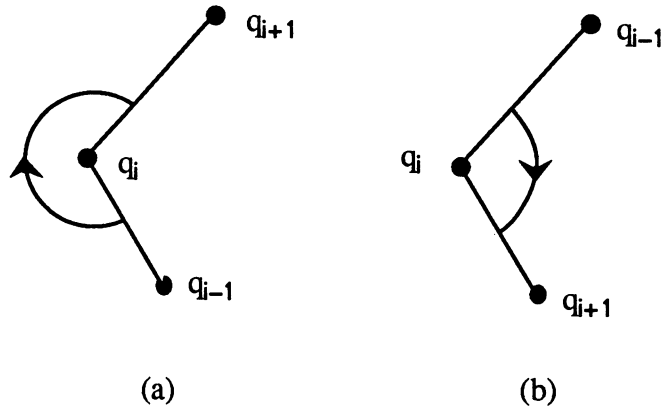


Figure 3.2 sommet concave (a) et sommet convexe (b)

Définition 3.2 Etant donné un polygone simple $G = (q_0, q_1, \dots, q_{N-1})$, une *chaîne polygonale* de G basée sur des sommets consécutifs est définie comme suit:

$$CV(i, j) = \langle q_i, \dots, q_h, \dots, q_j \rangle \tag{3.2}$$

où q_i et q_j sont sommets convexes et q_h (pour $h=i+1$ jusqu'à $j-1$) est sommet concave. La longueur de la chaîne est égale au nombre d'arêtes dans la chaîne.

N°. de Chaîne

1		
	.	.
	.	.
	.	.
k	pointeur (q_i)	pointeur (q_j)
	.	.
	.	.
	.	.
Cmax		

Figure 3.3 Représentation d'un polygone sous forme de chaînes basées sur les sommets

$$CV(k) = CV(i, j) = \langle q_i, \dots, q_h, \dots, q_j \rangle$$

Tout polygone peut se décomposer en chaînes polygonales successives basées sur les sommets. En conséquence, une représentation du polygone dans une machine est un tableau unidimensionnel dont chaque unité contient un index du sommet par lequel la chaîne commence et un autre par lequel la chaîne se termine. C'est un tel schéma de représentation qui est illustré en figure 3.3.

Des exemples de chaînes sont donnés en figure 3.4. Selon cette définition, $CV(i, i+1)$ est réduite à une chaîne de longueur un si les deux sommets consécutifs q_i et q_{i+1} sont convexes.

3.2.3 Décomposition basée sur les chaînes d'éléments

Dans le chapitre 2, nous avons introduit le concept d'élément qui est soit un point, soit un segment de droite ouvert. Au cas où nous voudrions construire le diagramme de Voronoï pour les objets polygonaux, les éléments sont soit les arêtes du polygone à l'exclusion des deux extrémités, soit les sommets.

Une chaîne basée sur les éléments, CE , est induite à partir de la chaîne basée sur les sommets :

$$CE(i, j) = \langle e_i, \dots, e_s, e_t, \dots, e_j \rangle \quad (3.3)$$

où

$$\begin{aligned} e_i &= q_i, \\ e_j &= q_j, \\ e_s &= \begin{cases} (q_{h-1}, q_h) & \text{si } q_h \text{ existe} \\ \emptyset & \text{sinon} \end{cases} \end{aligned} \quad (3.4)$$

et

$$e_t = q_h.$$

Par exemple, les chaînes correspondantes illustrées en figure 3.4 sont formulées ci-après:

(a) $CV(1) = \langle a, b \rangle = \langle q_1, q_2 \rangle$

$CE(1) = \langle a, (a,b), b \rangle = \langle e_1, e_2, e_3 \rangle$

(b) $CV(1) = \langle a, b, c \rangle = \langle q_1, q_2, q_3 \rangle$

$CE(1) = \langle a, (a,b), b, (b,c), c \rangle = \langle e_1, e_2, e_3, e_4, e_5 \rangle$

(c) $CV(1) = \langle a, b, c, d \rangle = \langle q_1, q_2, q_3, q_4 \rangle$

$CE(1) = \langle a, (a,b), b, (b,c), c, (c,d), d \rangle = \langle e_1, \dots, e_7 \rangle$

(d) $CV(1) = \langle a, b \rangle = \langle q_1, q_2 \rangle$

$CE(1) = \langle a, (a,b), b \rangle = \langle e_1, e_2, e_3 \rangle$

$CV(2) = \langle b, c \rangle = \langle q_2, q_3 \rangle$

$CE(2) = \langle b, (b,c), c \rangle = \langle e_1, e_2, e_3 \rangle.$

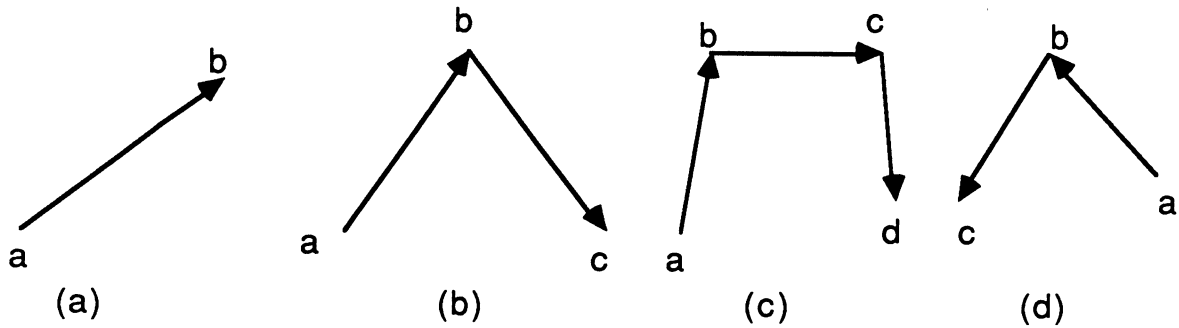


Figure 3.4 exemples des chaînes

Du fait que tous les objets polygonaux que nous manipulons sont des figures planes sans pondération d'arêtes, nous mesurerons la *longueur* (LEN) d'une chaîne par le nombre des arêtes dans la chaîne. De plus, le nombre des éléments dans une chaîne est déterminé par la longueur de la chaîne. Précisément, celui-ci peut s'exprimer selon la formule suivante:

$$\text{NBR}(\text{éléments}) = 2 \text{LEN}(\text{chaîne}) + 1 \quad (3.5)$$

Afin de générer le diagramme de Voronoï pour tous les éléments, nous construisons d'abord une liste chaînée d'éléments à partir de la liste chaînée de sommets. Ensuite, nous définissons à partir de la liste chaînée d'éléments un tableau de pointeurs identifiant l'élément par lequel la chaîne commence et l'élément par lequel la chaîne se termine. Une telle structure de données est similaire à celle présentée en figure 3.3.

Nous avons représenté l'objet polygonal par les éléments. Ainsi nous avons décomposé un polygone en chaînes d'éléments. Un cas particulier se présente lorsque le nombre de chaînes décomposées est égal à zéro. C'est-à-dire, qu'on ne peut pas obtenir une chaîne complète selon la définition donnée ci-dessus. Cela ne peut apparaître que lorsqu'un polygone convexe (tous les angles internes sont inférieurs à π) est parcouru selon le sens CW et on s'intéresse seulement à l'extérieur du polygone convexe. Nous désignerons par *boucle* une telle situation. La figure 3.5 montre deux exemples de boucles.

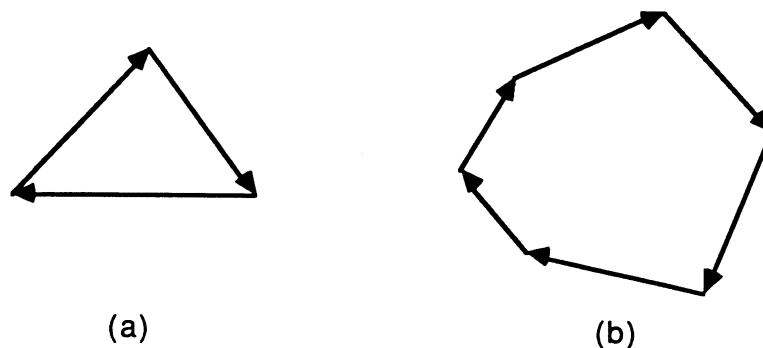


Figure 3.5 Quand on ne s'intéresse qu'à l'extérieur d'un polygone convexe, la chaîne décomposée sera une boucle

3.3 Diagramme de Voronoï pour l'intérieur d'un polygone

3.3.1 Diagramme de Voronoï d'une chaîne d'éléments

Le diagramme de Voronoï pour une chaîne d'éléments est composé de demi-droites initialisées ou se terminant aux sommets de la chaîne. La construction d'un tel diagramme sera très facile. L'algorithme pour construire le diagramme de Voronoï d'une chaîne d'éléments est donné ci-dessous.

Algorithme VORONOI_OF_CHAIN ;

{ les éléments sont stockés en une liste doublement chaînée; p_c : pointeur vers l'élément courant ; p_p : pointeur vers l'élément précédent; p_s : pointeur vers l'élément suivant ; $e(x)$: le contenu du pointeur x }

Entrée: $pre-el$, $der-el$ { pointeurs de premier élément et de dernier élément de chaîne $CE(i)$ }

Sortie: $VOD(CE(i))$; { le diagramme de Voronoï pour chaîne $CE(i)$ }

Début

1. initialisation. $p_c \leftarrow pre-el$;

2. tant que $p_c \neq der-el$ faire

début

{ créer une région de Voronoï pour l'élément courant }

a). si $p_c = pre-el$ alors déterminer une demi-droite passant par le point correspondant à cet élément et perpendiculaire au segment associé à l'élément de p_s ;

b). sinon si $p_c = der-el$, alors déterminer une demi-droite passant par le point correspondant à cet élément et perpendiculaire au segment associé à l'élément de p_p ;

c). sinon

si $e(p_c)$ est type SEGMENT alors

déterminer deux demi-droites orientées à partir des deux extrémités du segment et perpendiculaires à l'élément $e(p_c)$;

sinon

si $e(p_c)$ est de type POINT, alors déterminer deux demi-droites orientées dont l'une commence à partir de ce point et est perpendiculaire à l'élément précédent $e(p_c)$ et l'autre se termine en ce point et est perpendiculaire à celui suivant $e(p_s)$;

d). lier toutes les demi-droites en une liste doublement chaînée ; donc la frontière de la région de Voronoï $V(e(p_c))$ est bien représentée par une liste doublement chaînée de demi-droites ;

e). $p_c \leftarrow p_s$;

fin ;

Fin ;

L'orientation d'une arête de Voronoï dans l'algorithme VORONOI_OF_CHAIN est définie de telle manière que l'élément e est toujours situé à droite ou sur une arête de Voronoï quand on traverse la frontière de la région de Voronoï correspondante dans le sens CW. Les figures 3.6 et 3.7 donnent des exemples de diagrammes de Voronoï pour des chaînes d'éléments. La chaîne la plus courte est un segment de droite fermé composé de trois éléments: deux points d'extrémité et le segment ouvert correspondant. Soit $[a, b]$ une telle chaîne illustrée en figure 3.6. L'orientation $a \rightarrow b$ signale que nous nous intéressons aux points qui sont situés à droite de $[a, b]$. $\partial V(a)$, la frontière de la région du premier élément a de cette chaîne est composée d'une seule demi-droite orientée $[a, c]$. De même, $\partial V(b)$, est composée aussi d'une seule demi-droite orientée $[d, b]$. Ainsi, $\partial V((a, b))$ est composée de deux demi-droites $[c, a]$ et $[b, d]$, et de $[a, b]$. D'une façon similaire, nous pouvons donner le diagramme de Voronoï pour une chaîne de longueur supérieure à un. Ceci est montré dans la figure 3.7.

Il faut mentionner que les régions de Voronoï des deux extrémités d'une chaîne sont singulières, par exemple $V(a)$ et $V(b)$ en figure 3.6, $V(l)$ et $V(9)$ dans la figure 3.7. En réalité, ces régions singulières sont provisoires, puisqu'il n'existe pas de telles régions dans le diagramme de Voronoï final pour l'intérieur d'un polygone ni dans le diagramme de Voronoï pour l'extérieur de polygones.

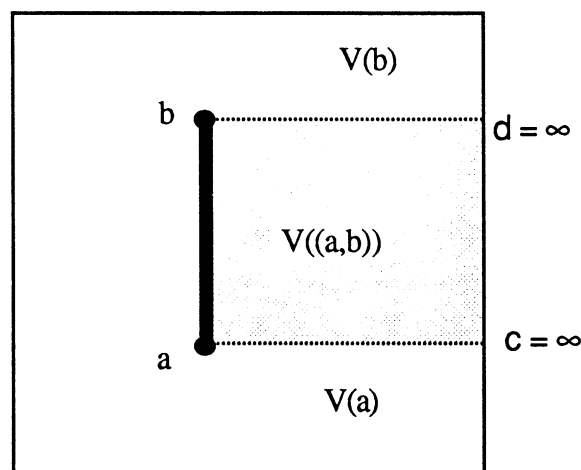


Figure 3.6 Diagramme de Voronoï d'un côté d'un segment fermé

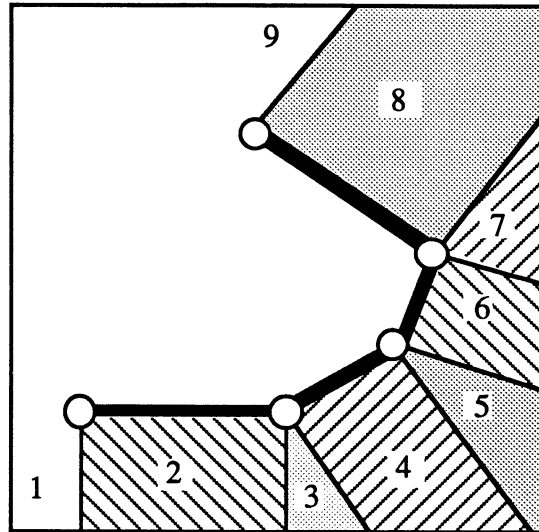


Figure 3.7 Diagramme de Voronoï d'un côté pour une chaîne de longueur supérieure à un

3.3.2 Approche Divide-and-Conquer pour l'intérieur d'un polygone

L'algorithme que nous avons adopté pour construire le diagramme de Voronoï d'un domaine intérieur à un polygone est structuré comme suit.

Algorithme INVOD ;

entrée : $\partial\Omega_i$

sortie : $VOD(\partial\Omega_i) \cap \zeta\Omega_i$

Début

1. Décomposer $\partial\Omega_i$ en une suite de chaînes basées sur les éléments,

$$\partial\Omega_i = \langle CE(1), \dots, CE(M) \rangle,$$

de telle façon que $\zeta\Omega_i$ soit toujours situé à gauche de $\partial\Omega_i$ quand on parcourt $\partial\Omega_i$ (selon un sens CCW).

2. Diviser $\partial\Omega_i$ en deux sous-ensembles (notés G_1 et G_2) de chaînes basées sur les éléments ayant à peu près la même taille, c'est-à-dire:

$$G_1 = \langle CE(1), \dots, CE(k) \rangle$$

et

$$G_2 = \langle CE(k+1), \dots, CE(M) \rangle$$

où

$$k = \text{round}(M/2);$$

3. Construire $VOD(\partial G_1) \cap \zeta G_1$ et $VOD(G_2) \cap \zeta G_2$ récursivement;

4. Fusionner $VOD(\partial G_1) \cap \zeta G_1$ et $VOD(G_2) \cap \zeta G_2$ pour obtenir $VOD(\partial \Omega_i) \cap \zeta \Omega_i$;

Fin;

On retrouve le schéma d'une méthode du type divide-and-conquer. Au cours de la récursivité, la fusion des deux sous-diagrammes est la clé de l'algorithme. La suite de ce paragraphe va être consacrée à la présentation de cette étape cruciale.

Supposons que nous ayons deux sous-diagrammes de Voronoï, $VOD(G_1)$ et $VOD(G_2)$ où G_1 et G_2 sont deux sous-ensembles de G disjoints, sauf aux extrémités de G_1 et G_2 . Pour obtenir une nouvelle partition de Voronoï de ζG , il nous faut d'abord trouver un séparateur de G_1 et G_2 , noté par $Sep(G_1, G_2)$ tel que

$$Sep(G_1, G_2) = \{p \mid \min_{e_k \in G_1} d(p, e_k) = \min_{e_l \in G_2} d(p, e_l)\}$$

Soient G_1 et G_2 deux sous-ensembles de chaînes basées sur les éléments décomposées à partir de quelques polygones (non nécessairement fermés) durant la récursivité de l'algorithme INVOD. Alors il est évident que G_1 et G_2 sont liés en "tête-à-queue", c'est-à-dire, G_1 et G_2 ont en commun un élément de type POINT. Précisément, cet élément est celui du début de G_1 et celui de la fin de G_2 ou l'inverse.

On déclare que les deux ensembles de chaînes, G_1 et G_2 , sont voisins s'il existe au moins une chaîne $CE(i) \in G_1$ et une chaîne $CE(j) \in G_2$ telle que $CE(i)$ et $CE(j)$ soient successives (figure 3.8).

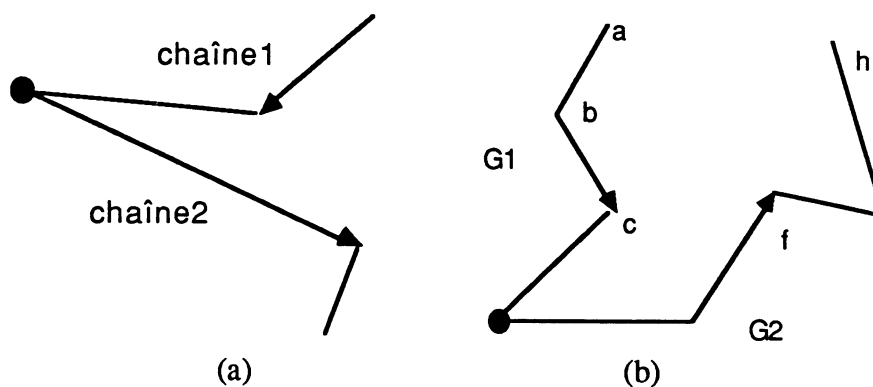


Figure 3.8 chaînes successives et ensembles de chaînes voisins

Il est évident que les deux sous-ensembles de chaînes formés pendant la récursivité de l'algorithme INVOD sont toujours voisins.

Point de départ pour la construction du séparateur entre deux chaînes polygonales

Lemme 3.1 L'intersection, $G_1 \cap G_2$, de deux sous-ensembles de chaînes voisines formés pendant la récursivité de l'algorithme INVOD est située dans le séparateur de G_1 et G_2 . C'est-à-dire:

$$G_1 \cap G_2 \in Sep(G_1, G_2) \tag{3.6}$$

Preuve. Soit $e = G_1 \cap G_2$. Il est évident que e est un élément de type POINT. On a:

$$e \in \{ p \mid \min_{e_k \in G_1} d(p, e_k) = \min_{e_l \in G_2} d(p, e_l) \}$$

On rappelle la définition du séparateur:

$$Sep(G_1, G_2) = \{ p \mid \min_{e_k \in G_1} d(p, e_k) = \min_{e_l \in G_2} d(p, e_l) \}$$

Donc

$$e \in Sep(G_1, G_2)$$

Comme à chaque étape de récursivité le séparateur $Sep(G_1, G_2)$ est semi-borné ou borné et l'intersection $G_1 \cap G_2$ est un point appartenant à $Sep(G_1, G_2)$ (lemme 3.1), nous pouvons alors prendre $G_1 \cap G_2$ comme point de départ pour tracer $Sep(G_1, G_2)$. Par conséquent, nous ajoutons une direction au séparateur, et remplaçons G_1 par G_L et G_2 par G_R , ayant précisé que G_L et G_R sont situées respectivement à la gauche et à la droite du séparateur.

Condition d'arrêt d'un processus de fusion

Etant donné un polygone G , les deux ensembles de chaînes polygonales, G_L et G_R , sont appelés les *divisions premières* de G si $G_L \cup G_R = G$. Autrement G_L et G_R sont appelés les *divisions secondaires* de G .

Afin de donner la condition d'arrêt de la fusion, considérons d'abord les deux cas suivants.

(1) G_L et G_R sont les divisions premières du polygone donné. Un tel exemple est montré dans la figure 3.9.

Etant donné un ensemble de chaînes orientées G , notons par $HEAD(G)$ et par $TAIL(G)$ respectivement la tête et la queue de G . Alors, on a les relations:

$$HEAD(G_L) = TAIL(G_R)$$

et

$$HEAD(G_R) = TAIL(G_L).$$

Dans ce cas, l'intersection de G_L et G_R est:

$$\begin{aligned} G_L \cap G_R &= \{ HEAD(G_L), TAIL(G_L) \} \\ &= \{ TAIL(G_R), HEAD(G_R) \} \end{aligned}$$

D'après le lemme 3.1, ces deux éléments sont situés sur $Sep(G_L, G_R)$. Ainsi, il est évident dans ce cas que $Sep(G_L, G_R)$ est borné. Donc, $HEAD(G_L)$ ou $TAIL(G_R)$ est le point d'arrivée du séparateur de G_L et G_R . Alors, la condition d'arrêt de la construction du séparateur dans ce cas peut s'exprimer facilement.

(2) G_L et G_R sont les divisions secondaires du polygone donné (voir figure 3.10). Dans ce cas, nous avons:

$$G_L \cap G_R = \{ HEAD(G_R) \} = \{ TAIL(G_L) \}$$

Ceci correspond au point de départ du séparateur.

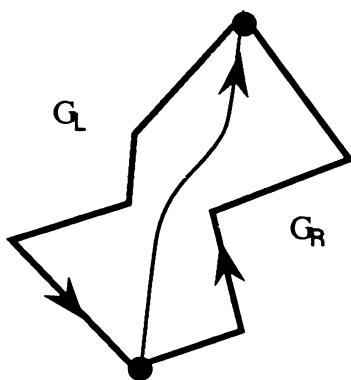


Figure 3.9 G_L et G_R sont les divisions premières du polygone G

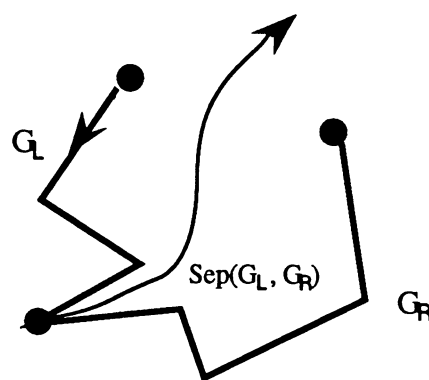


Figure 3.10 G_L et G_R sont les divisions secondaires du polygone G

En effet, le séparateur de G_L et G_R consiste en une suite de composants de bissectrices $B(e_l, e_r)$ où $e_l \in G_L$ et $e_r \in G_R$. Donc, si nous avons tracé le dernier composant de la bissectrice, nous construisons alors le séparateur de G_L et G_R .

Soit e_{ls} et e_{rs} deux éléments par lesquels le processus de construction du séparateur se termine. D.T. Lee [LEE 82a] a donné deux conditions d'arrêt sans preuve pour ces deux cas:

- (a) $e_{ls} = \text{HEAD}(G_L)$ et $e_{rs} = \text{TAIL}(G_R)$ ou
- (b) tous les sommets de $G_L \cup G_R$ sont situés à la droite du segment orienté, déterminé par e_{ls} et e_{rs} où e_{ls} et e_{rs} sont deux sommets.

Nous allons démontrer que ces deux conditions ne sont pas suffisantes pour vérifier si le séparateur de G_L et G_R est entièrement déterminé. L'exemple de la figure 3.11 soutient une telle remarque.

Supposons que nous ayons un polygone ouvert G à traiter (cf. la figure 3.11). Les flèches sur le polygone indiquent que nous sommes intéressés par la partie gauche du polygone si la frontière du polygone est parcourue selon le sens des flèches. Ce polygone est composé de trois chaînes C_1, C_2 et C_3 où $C_1 = \langle v_1, v_2 \rangle$, $C_2 = \langle v_2, \dots, v_6 \rangle$ et $C_3 = \langle v_6, v_7 \rangle$. Selon le schéma 'divide-and-conquer', nous pouvons diviser G en deux sous-polygones, $G_L = C_1$ et $G_R = C_2 \cup C_3$.

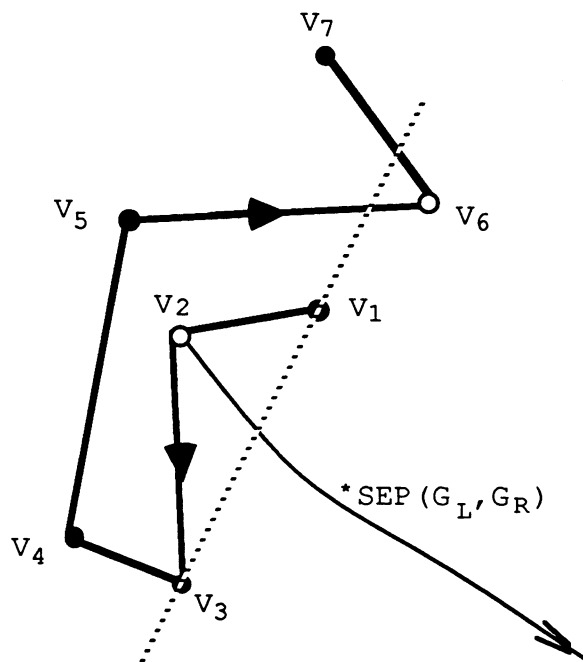


Figure 3.11 Un exemple ne convenant pas aux conditions d'arrêt de Lee [LEE 82a].

Quand nous construisons le séparateur entre G_L et G_R , nous pouvons vérifier qu'il est composé de trois composants. Ceux-ci sont les parties des bissectrices $B((v_1, v_2), (v_2, v_3))$, de $B(v_1, (v_2, v_3))$ et de $B(v_1, v_3)$. Ici, $B(v_1, v_3)$ est la dernière partie de $Sep(G_L, G_R)$ orientée. C'est-à-dire que dans ce cas $Sep(G_L, G_R)$ se termine par $B(e_l, e_r)$ où $e_l = v_1$ et $e_r = v_3$. Examinons tous les

sommets de G . Il est évident qu'aucune des deux conditions données par Lee n'est vérifiée ici. Donc ce contre-exemple montre que les conditions d'arrêt de Lee sont incomplètes.

Afin de trouver la condition générale d'arrêt pour la construction du séparateur de G_L et G_R , nous devons introduire le concept de visibilité défini comme suit.

Définition 3.3 Soit Ω un polygone simple, un point quelconque $s \in \partial\Omega$ est *visible* à partir d'un point q hors du polygone Ω si et seulement si le segment de droite (s, q) ne contient aucun point intérieur du polygone. (cf. la figure 3.12)

Définition 3.4 Deux éléments sont réciproquement visibles si tous les points des éléments sont *visibles mutuellement* ou *visibles entièrement*.

Quelquefois, on parle de *visibilité partielle*. Deux éléments e_i et e_j sont visibles partiellement s'il existe deux sous-ensembles non vides de points $s_i \subset e_i$ et $s_j \subset e_j$ telle que s_i et s_j sont visibles mutuellement.

Définition 3.5 Soit $q \in \partial\Omega$ (Ω est un polygone fermé). Soit p un sommet de $\partial\Omega$. p est *visible intérieurement* à partir de q si et seulement si le segment (p, q) est situé à l'intérieur de Ω . p est *visible extérieurement* à partir de q si et seulement si le segment (p, q) est situé à l'extérieur de Ω .

Dans la figure 3.14, s est visible intérieurement à partir de q ; t est visible extérieurement à partir de q ; et p n'est ni visible intérieurement ni extérieurement à partir de q . Il est facile de montrer que les deux segments joints sont visibles extérieurement à condition que ces deux segments forment un angle convexe (vu à partir de l'intérieur du polygone).

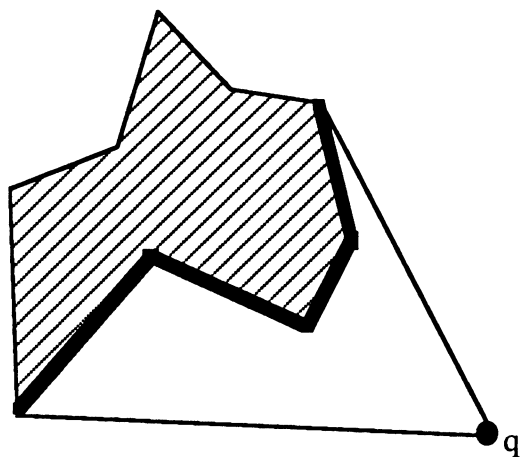


Figure 3.12 Visibilité à partir d'un point q

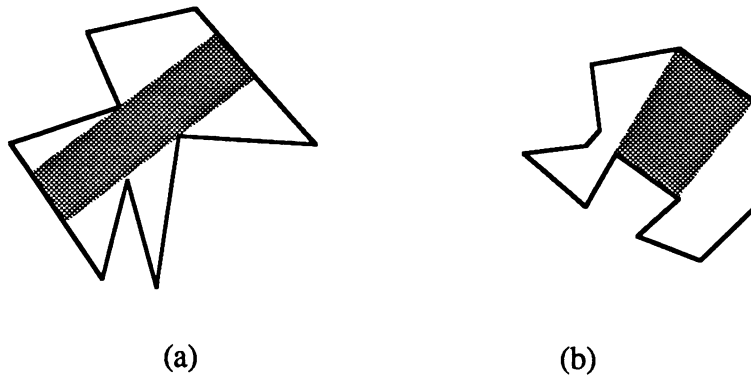


Figure 3.13 Visibilité partielle (a) et entière (b)

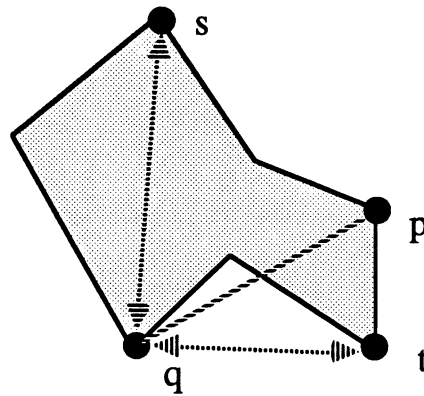


Figure 3.14. Visibilité intérieure et extérieure

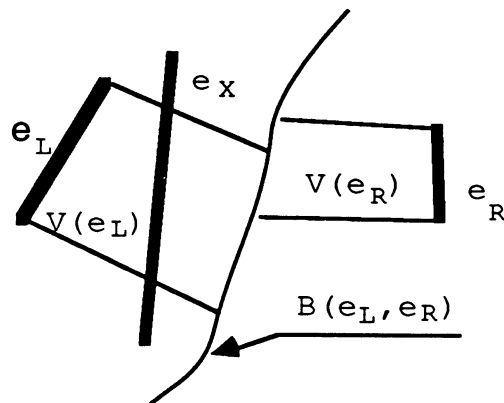


Figure 3.15 Illustration pour le lemme 3.2

Lemme 3.2 Deux éléments mutuellement invisibles ne partagent jamais en commun la même arête commune de Voronoï.

Preuve. Supposons d'abord que deux éléments e_L et e_R mutuellement invisibles soient communs à quelque arête de Voronoï, notée $B(e_L, e_R)$. Alors, tous les points situés à gauche de $B(e_L, e_R)$ appartiennent à $V(e_L)$ et ceux situés à droite de $B(e_L, e_R)$ appartiennent à $V(e_R)$.

Comme e_L et e_R sont invisibles réciproquement, alors il existe certainement quelque élément e_x situé entre e_L et e_R qui empêche la visibilité entre e_L et e_R . Par conséquent, il existe des points appartenant à $V(e_L)$ qui sont plus proches de e_x que de e_L et il existe des points appartenant à $V(e_R)$ qui sont plus proches de e_x que de e_R . Ceci est en contradiction avec la définition du diagramme de Voronoï. (voir figure 3.15).

Maintenant, nous pouvons, d'après le lemme 3.2, formuler une condition générale indiquant qu'un processus de fusion de deux sous-diagrammes de Voronoï se termine correctement.

Corollaire 3.1 Etant donné un polygone G orienté, composé de deux sous-polygones G_L et G_R , alors la condition d'arrêt de construction du séparateur $Sep(G_L, G_R)$ s'exprime comme suit: trouver deux éléments de type "POINT" (sommets) e_{SL} et e_{SR} qui divisent respectivement G_L et G_R en deux parties notées:

$$G_L = \langle e_1, \dots, e_{SL} \rangle \cup \langle e_{SL}, e_{SL+1}, \dots, e_K \rangle \quad (3.7)$$

et

$$G_R = \langle e_{K+1}, \dots, e_{SR} \rangle \cup \langle e_{SR}, e_{SR+1}, \dots, e_N \rangle \quad (3.8)$$

telles que

$$\langle e_{SL}, e_{SL+1}, \dots, e_K \rangle \text{ et } \langle e_{SR}, e_{SR+1}, \dots, e_N \rangle$$

sont mutuellement invisibles.

Preuve. D'abord nous devons remarquer que tous les séparateurs concernés dans l'algorithme INVOD sont continus. Une fois que nous avons rencontré un couple d'éléments, $e_l \in G_L$ et $e_r \in G_R$, qui sont mutuellement invisibles, alors le processus de calcul du séparateur se termine. En effet si e_l et e_r sont invisibles, d'après le lemme 3.2 il n'y a pas d'arête commune entre e_l et e_r . Par conséquent, le séparateur sera interrompu à ce moment-là. Mais ceci est impossible. La seule possibilité est que le séparateur s'étende jusqu'à l'infini.

En reconsidérant le contre-exemple de la figure 3.11 avec les conditions d'arrêt ci-dessus, il est facile de comprendre pourquoi les conditions données par Lee [LEE 82a] ne sont pas suffisantes. Pour cet exemple-là, on a $e_{SL} = v_1$ et $e_{SR} = v_3$. Tous les éléments situés après v_3 dans G_R sont intérieurement invisibles vis-à-vis de v_1 .

Maintenant, l'étape 4 de fusion de l'algorithme INVOD peut être détaillée dans l'algorithme donné ci-dessous.

Algorithme MERGEI ;

entrée: G_L, G_R où $HEAD(G_R) = TAIL(G_L)$;

sortie: $VOD[\partial(G_L \cup G_R)] \cap \zeta(G_L \cup G_R)$;

Début

1. Initialisation: Trouver un point de départ $S_0 = \text{HEAD}(G_R) = \text{TAIL}(G_L)$ et $e_L \in G_L$, $e_R \in G_R$ tel que S_0 soit situé dans $B(e_L, e_R)$;

2. {construire $Sep(G_L, G_R)$ }

Tantque $stop = \text{faux}$ faire

début

a. $stop \leftarrow \text{vrai}$ si les conditions d'arrêt sont vérifiées
sinon $stop \leftarrow \text{faux}$;

b. construire $B(e_L, e_R)$ à partir de S_0 ;

c. 1). $S_L \leftarrow B(e_L, e_R) \cap \partial V(e_L)$;

2). $S_R \leftarrow B(e_L, e_R) \cap \partial V(e_R)$;

3). Si $\text{distance}(S_0, S_L) > \text{distance}(S_0, S_R)$ alors $S_0 \leftarrow S_R$
sinon $S_0 \leftarrow S_L$;

d. Si S_0 est situé sur $\partial V(e_L, \partial G_L - e_L)$ alors

début

1). trouver $\lambda \in \partial G_L$ tel que $B(e_L, e_R)$ va entrer dans $V(\lambda, \partial G_L - \lambda)$ à partir de $V(e_L, \partial G_L - e_L)$;

2). $e_L \leftarrow \lambda$;

fin ;

e. Si S_0 est situé sur $\partial V(e_R, \partial G_R - e_R)$ alors

début

1). trouver $\rho \in \partial G_R$ tel que $B(e_L, e_R)$ va entrer dans $V(\rho, \partial G_R - \rho)$ à partir de $V(e_R, \partial G_R - e_R)$;

2). $e_R \leftarrow \rho$;

fin ;

fin ;

3. éliminer les portions de $VOD(\partial G_L)$ qui sont situées à la droite de $Sep(G_L, G_R)$ et toutes les portions de $VOD(\partial G_R)$ qui sont situées à la gauche de $Sep(G_L, G_R)$;

Fin ;

Le temps de calcul de l'algorithme Divide-and-Conquer peut s'exprimer selon une équation réursive:

$$T(N) = 2T(N/2) + \tau(N) \quad (3.9)$$

où N est la taille de l'ensemble original et $\tau(N)$ est le temps de fusion des deux solutions aux sous-problèmes. La solution de (3.9) dépend de $\tau(N)$. Si $\tau(N)$ est linéaire, alors $T(N)$ sera sous la forme $N \log N$. Dans l'algorithme INVOD, $\tau(N)$ est le temps de calcul de l'algorithme MERGEI. Nous reviendrons à l'analyse du temps de calcul de l'algorithme MERGEI au chapitre 5. Nous déclarons d'avance que le processus de la fusion de deux sous-diagrammes de Voronoï peut se faire en temps linéaire. Donc, nous avons le lemme suivant.

Lemme 3.3 L'algorithme INVOD est de complexité $O(N \log N)$ où N est le nombre d'éléments composant la frontière du polygone.

3.4 Diagramme de Voronoï pour l'extérieur d'un polygone

Soit Ω un polygone simple. Pour définir l'extérieur de Ω , nous devons parcourir la frontière de Ω , $\partial\Omega$, de telle manière que $\varepsilon\Omega$ soit toujours situé à la gauche de $\partial\Omega$. La figure 3.16 illustre des parcours inversés pour distinguer l'intérieur et l'extérieur du polygone Ω .

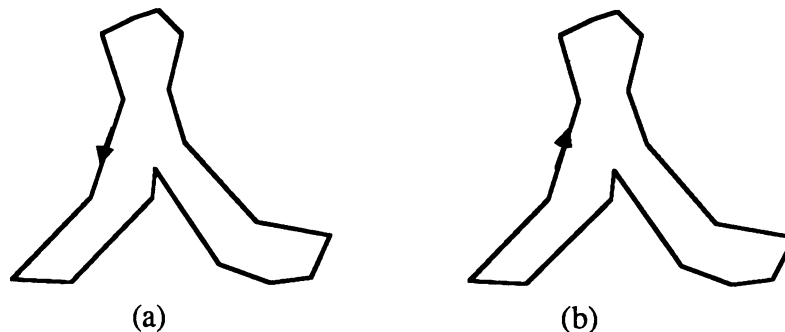


Figure 3.16 Un sens spécifié de balayage de la frontière d'un polygone définit l'intérieur (a) ou l'extérieur du polygone (b)

Le principe de l'algorithme pour calculer le diagramme de Voronoï de l'extérieur est identique à l'algorithme pour l'intérieur. Cependant, la construction du diagramme de Voronoï pour l'extérieur est en pratique un peu plus compliqué que celle décrite pour l'intérieur. Par exemple, soit P un polygone composé de 10 sommets et illustré en figure 3.17. Selon la définition de la chaîne (cf. §3.2.2), ce polygone peut se décomposer en 4 chaînes. Celles-ci sont : $C_1 = \langle 1, 2 \rangle$, $C_2 = \langle 2, 3, 4, 5, 6 \rangle$, $C_3 = \langle 6, 7 \rangle$, $C_4 = \langle 7, 8, 9, 10, 1 \rangle$. Quand nous appliquons l'algorithme INVOD, présenté dans le paragraphe précédent, à ces quatre chaînes, nous trouvons que le résultat issu de l'algorithme n'est pas correct. En effet le séparateur entre $C_1 \cup C_2$ et $C_3 \cup C_4$ est composé de deux parties dont l'une part du sommet 6 et s'étend jusqu'à l'infini, et l'autre part de l'infini jusqu'au sommet 1. On n'est plus dans le cas où un séparateur est une courbe sans

discontinuité comme pour l'intérieur d'un polygone. En outre il y a toujours un tel séparateur coupé en deux parties pour tous les polygones ayant plus de deux chaînes. Donc, l'algorithme INVOD doit être modifié afin de s'appliquer à l'extérieur d'un polygone. L'algorithme EXVOD décrit ci-après est la modification de l'algorithme INVOD. Avec cet algorithme, on peut construire correctement le diagramme de Voronoï pour l'extérieur d'un polygone simple.

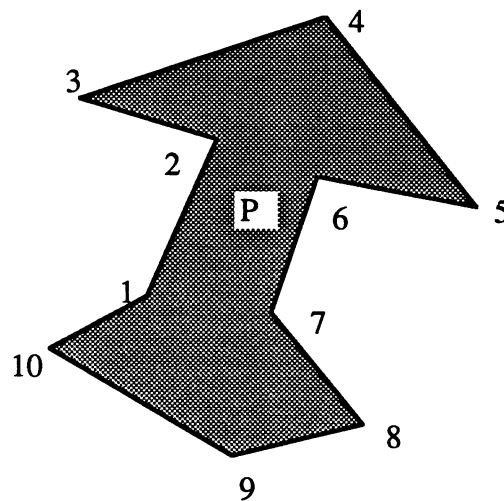


Figure 3.17 Un polygone ayant quatre chaînes

Algorithme EXVOD ;

entrée: $\partial\Omega$

sortie: $VOD(\partial\Omega) \cap \varepsilon\Omega$

Début

1. Décomposer $\partial\Omega$ en plusieurs chaînes d'éléments: $\partial\Omega = \langle CE(1), \dots, CE(N) \rangle$

avec

$$CE(1) \cap CE(N) = \text{HEAD}(CE(1)) = \text{TAIL}(CE(N))$$

et

$$CE(i) \cap CE(i+1) = \text{HEAD}(CE(i+1)) = \text{TAIL}(CE(i)) \text{ où } 1 < i < N;$$

de telle façon que $\varepsilon\Omega_i$ soit toujours situé à la gauche de $\partial\Omega$ quand on suit $\partial\Omega$ (selon le sens CW) ;

2. Diviser $\partial\Omega$ en deux sous-ensembles de chaînes basées sur les éléments G_L et G_R ayant à peu près la même taille et définis par :

$$G_L = \langle CE(1), \dots, CE(k) \rangle$$

et

$$G_R = \langle CE(k+1), \dots, CE(N) \rangle$$

où $k = (N/2)$;

3. début {récurrence}

a. Construire $VOD(\partial G_L) \cap \varepsilon G_L$ et $VOD(\partial G_R) \cap \varepsilon G_R$ récursivement ;

b. Fusionner $VOD(\partial G_L) \cap \varepsilon G_L$ et $VOD(\partial G_R) \cap \varepsilon G_R$ pour obtenir

$VOD(\partial \Omega) \cap \varepsilon \Omega$;

fin ; {récurrence}

4. {fusion supplémentaire}

début

a. En prenant $q = CE(1) \cap CE(N)$ comme point de départ, construire le séparateur

$Sep(G_R, G_L)$;

b. Renouveler toutes les régions de Voronoï que $Sep(G_R, G_L)$ traverse ;

fin ;

Fin ;

Dans l'algorithme *EXVOD*, on peut vérifier que la modification est effectuée à l'étape 4 vis-à-vis de l'algorithme *INVOD* présenté au §3.3.2. Après une récurrence normale, nous avons introduit une fusion supplémentaire où nous calculons d'abord le séparateur entre G_R et G_L , et ensuite renouvelons les régions de Voronoï concernées. Remarquons que G_R est mis à gauche du séparateur et G_L est mis à droite du séparateur. C'est le contraire du cas de fusion normale.

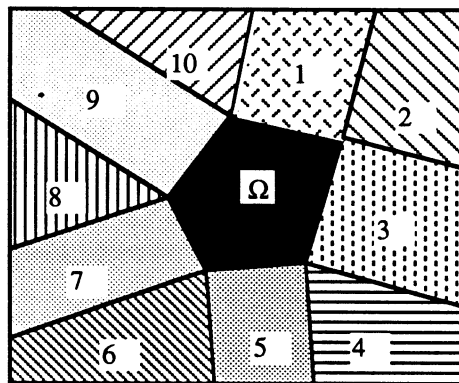


Figure 3.18 Diagramme de Voronoï pour l'extérieur d'un polygone convexe

Le polygone Ω de N sommets définit, à son extérieur,

$2N$ régions de Voronoï dont chacune est numérotée

Pour construire $VOD(\partial \Omega_i) \cap \varepsilon \Omega_i$, un cas particulier se présente: celui où $\partial \Omega_i$ est formé d'une boucle indiquée comme en figure 3.5. Il est évident que ce cas n'apparaît que lorsque Ω_i est un polygone convexe simple. Dans ce cas, il faut imposer que le nombre de chaînes soit égal à un. Le diagramme de Voronoï pour l'extérieur du polygone, $VOD(\partial \Omega_i) \cap \varepsilon \Omega_i$, s'identifie alors entièrement avec le diagramme de Voronoï d'une chaîne des éléments. Par conséquent, pour ce

cas particulier, on n'a besoin d'aucun processus de fusion. Le diagramme de Voronoï pour l'extérieur d'un polygone ne contient que des demi-droites initialisées ou se terminant aux sommets du polygone Ω_i . La figure 3.18 montre le diagramme de Voronoï en ce cas.

Comme conclusion, nous avons les lemmes suivants.

Lemme 3.5 L'algorithme EXVOD prend un temps de calcul en $O(N \log N)$ où N est le nombre d'éléments décomposés de la frontière du polygone.

Lemme 3.6 Le diagramme de Voronoï de l'extérieur d'un polygone convexe peut être construit en complexité $O(N)$ où N est le nombre de sommets du polygone.

3.5 Exemples de résultats

Quelques sorties du programme pour le diagramme de Voronoï généralisé basé sur les éléments de l'intérieur d'un polygone sont fournies dans les figures 3.19 (a) - (j). Les figures 3.20 (a) - (d) montrent quelques sorties du diagramme de Voronoï généralisé pour l'extérieur d'un polygone.

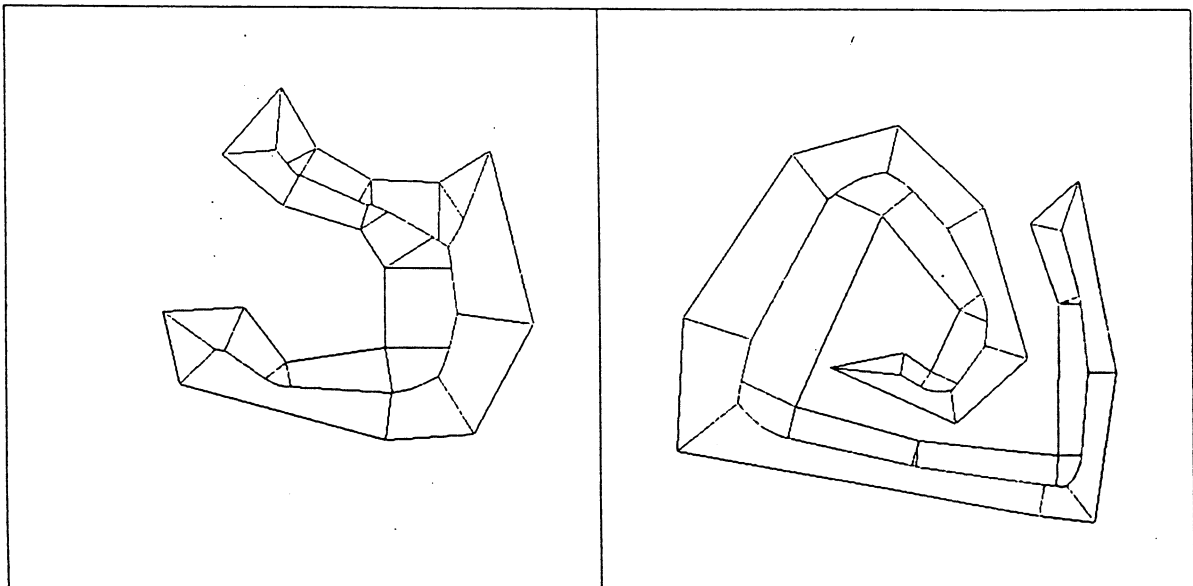


Figure 3.19 (a) et (b)

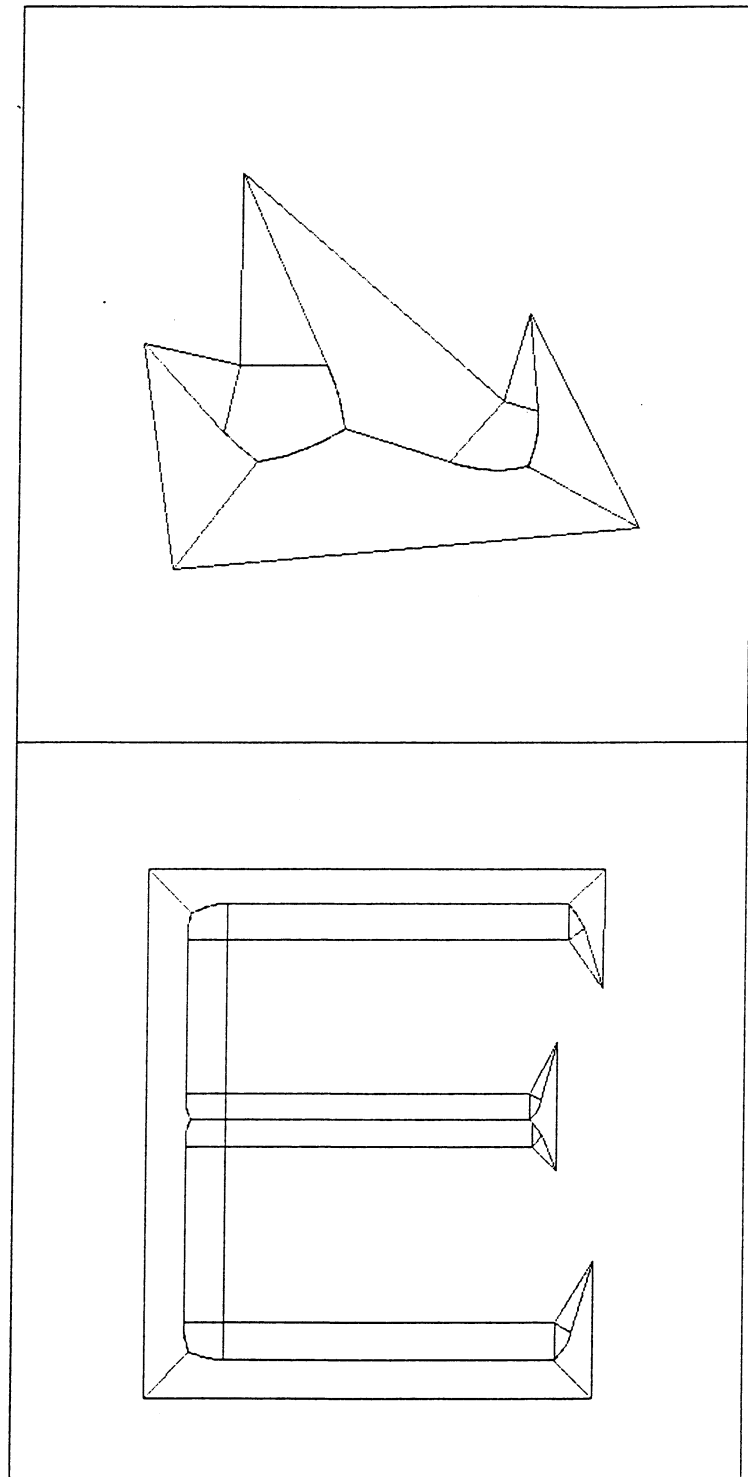


Figure 3.19 (c) et (d)

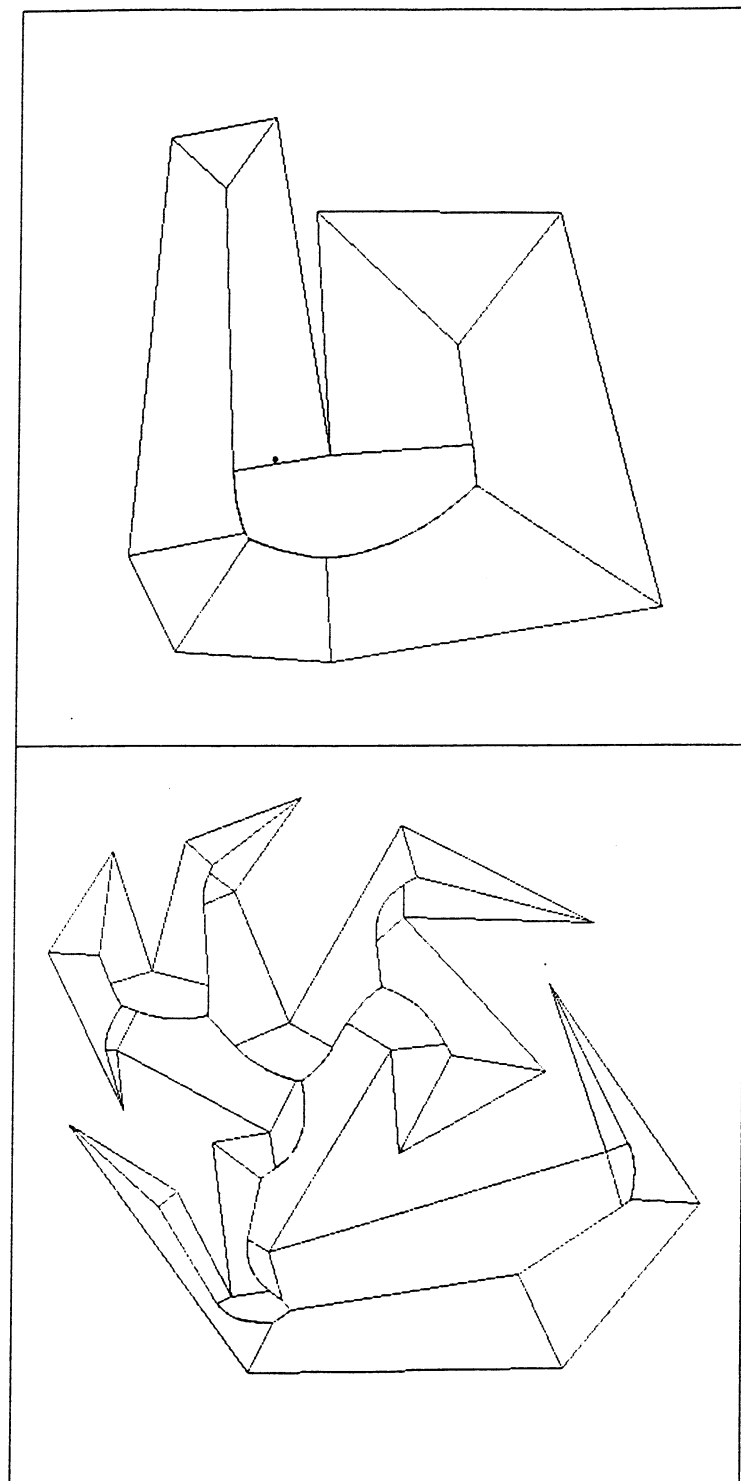


Figure 3.19 (e) et (f)

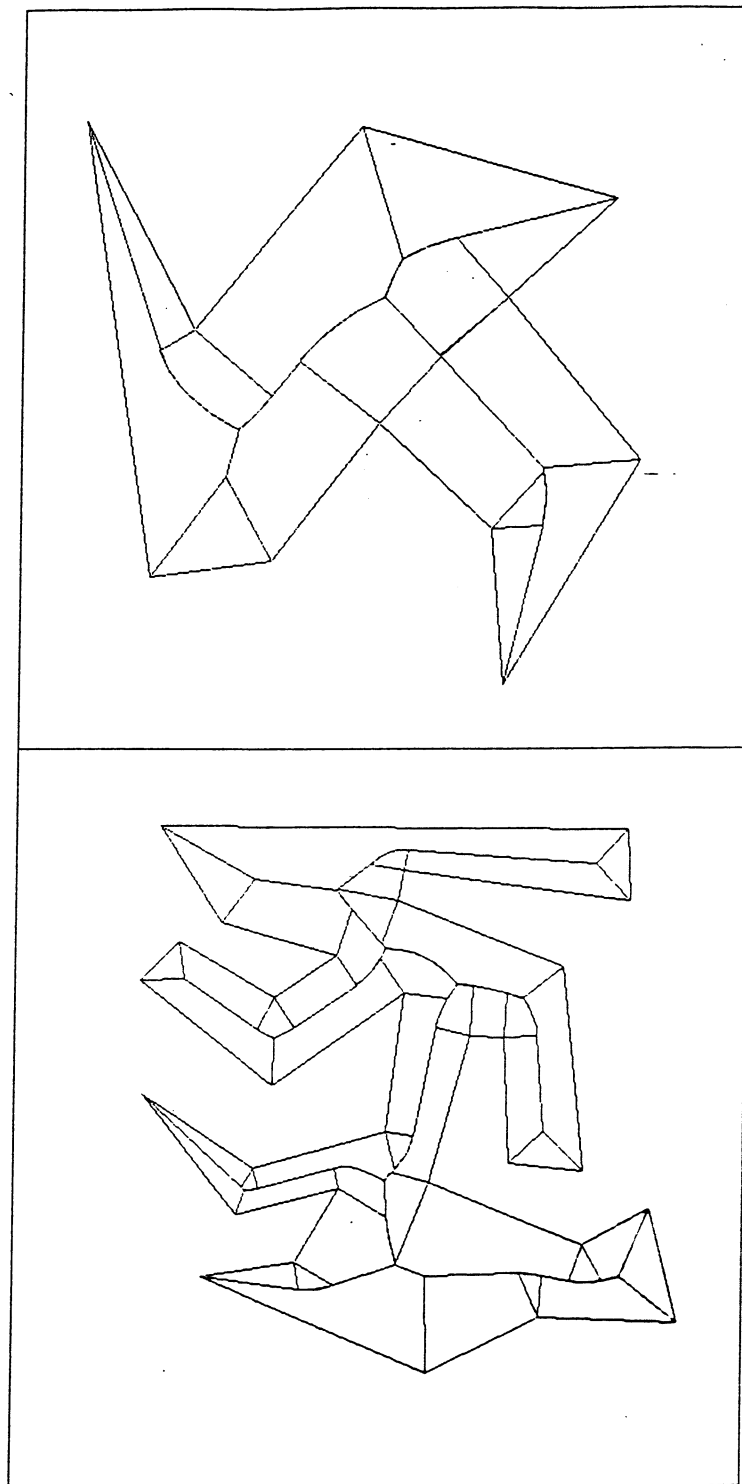


Figure 3.19 (g) et (h)

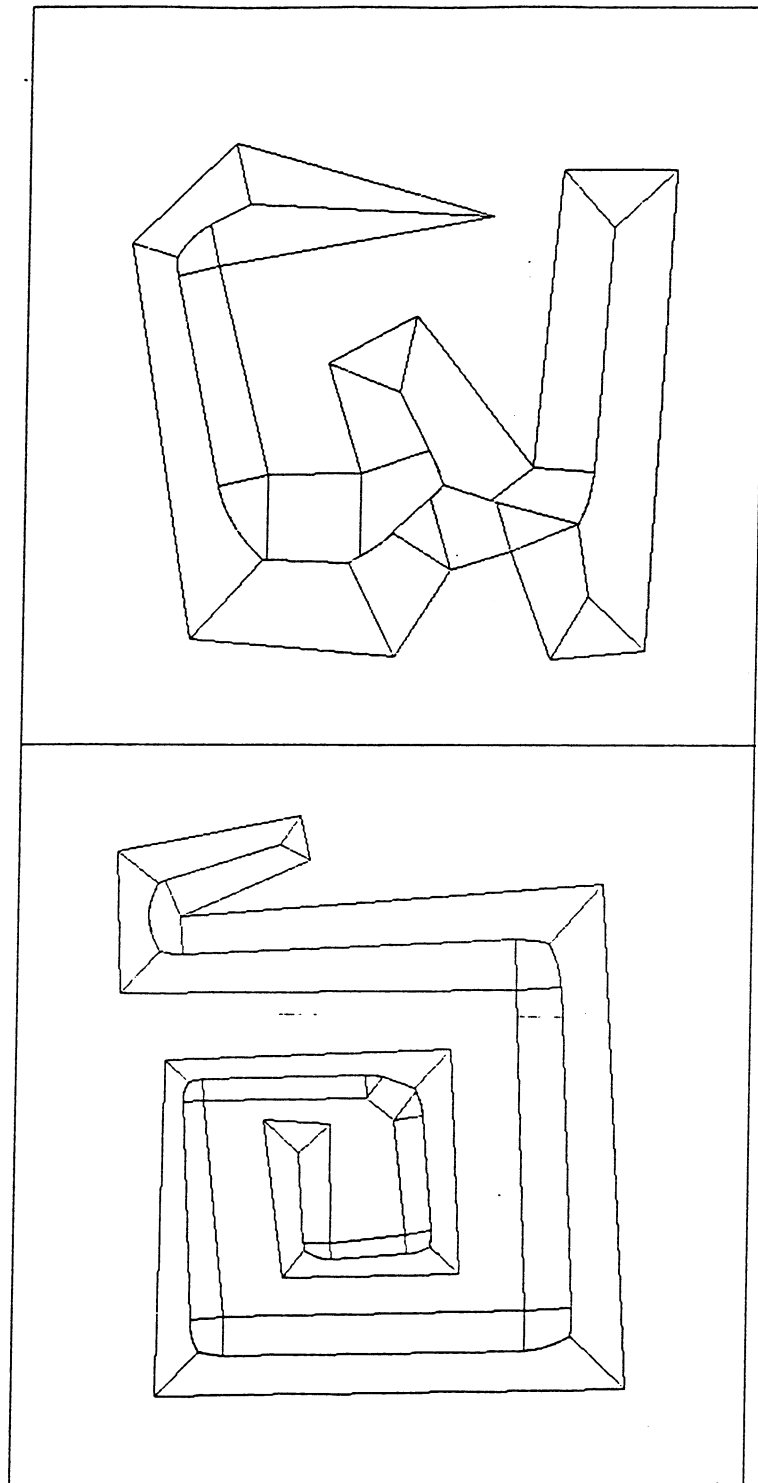


Figure 3.19 (i) et (j)

Figure 3.19 Diagrammes de Voronoï généralisés pour l'intérieur d'un polygone

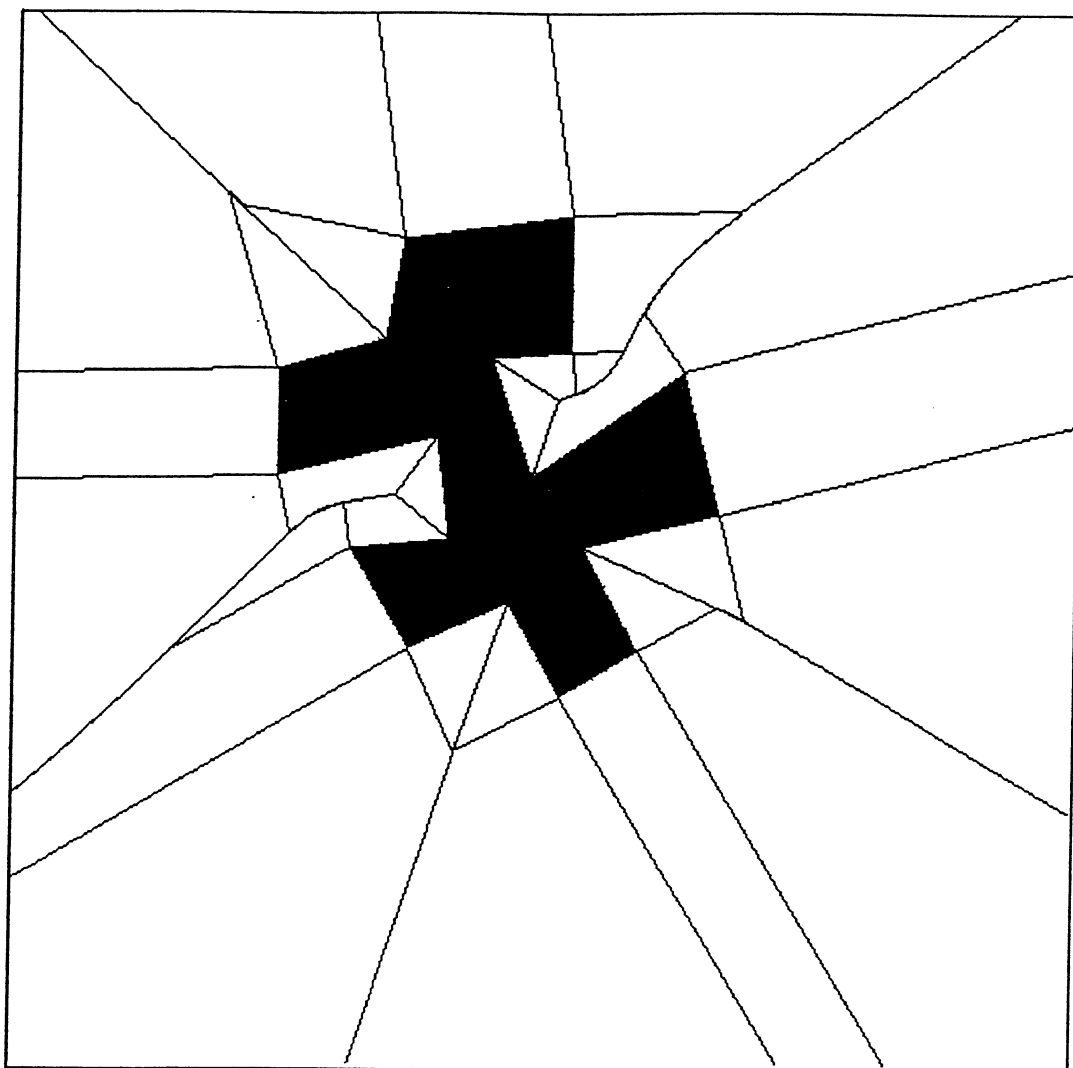


Figure 3.20 (a)

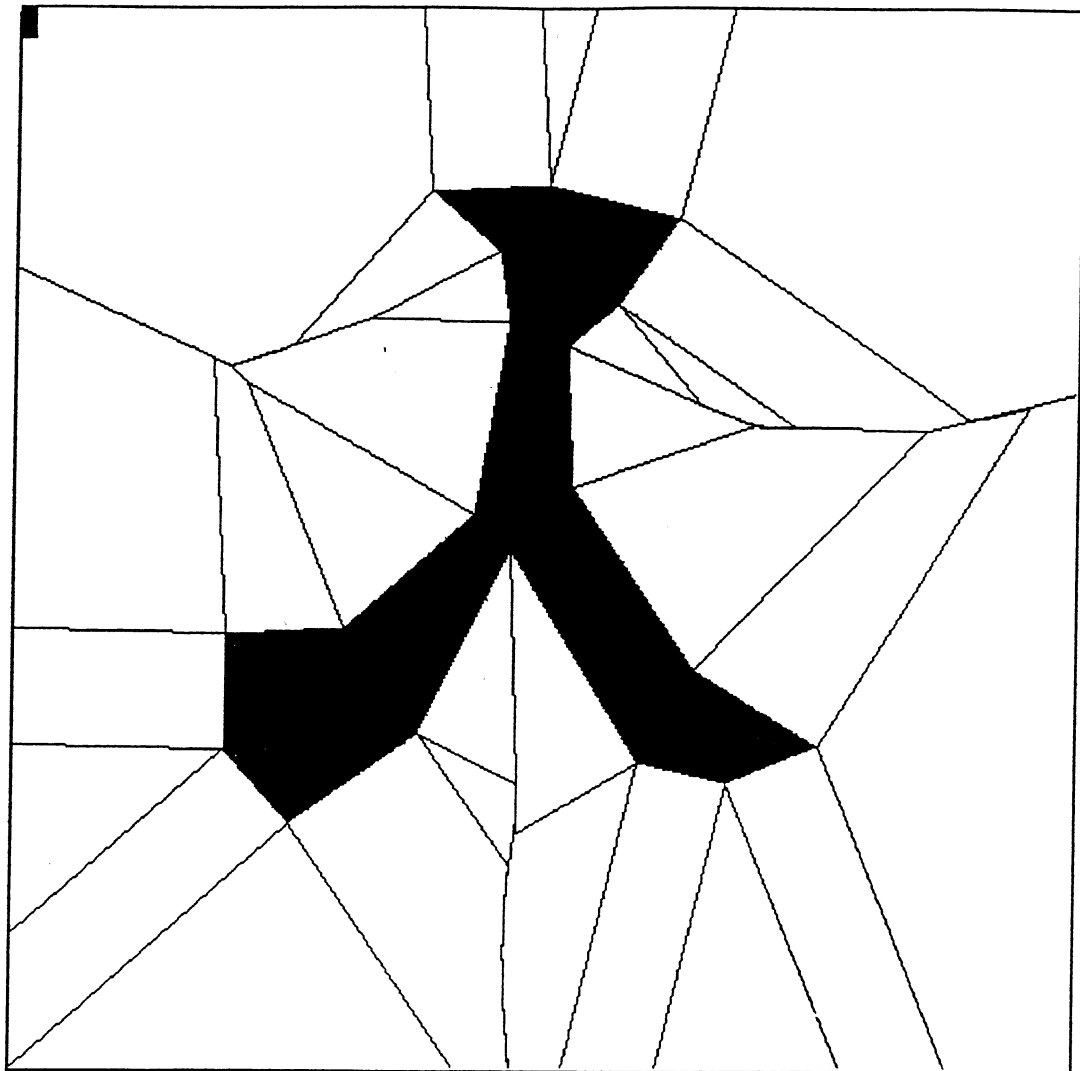


Figure 3.20 (b)

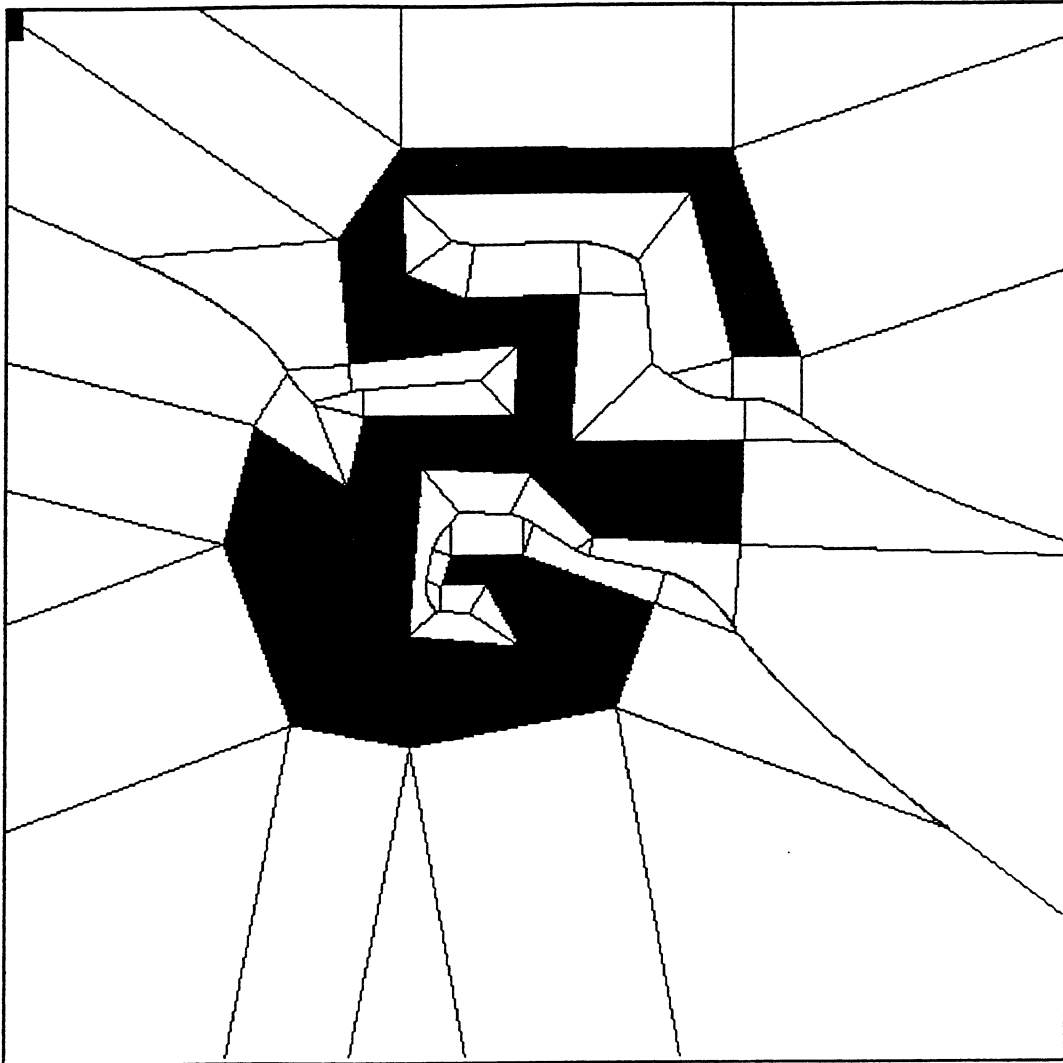


Figure 3.20 (c)

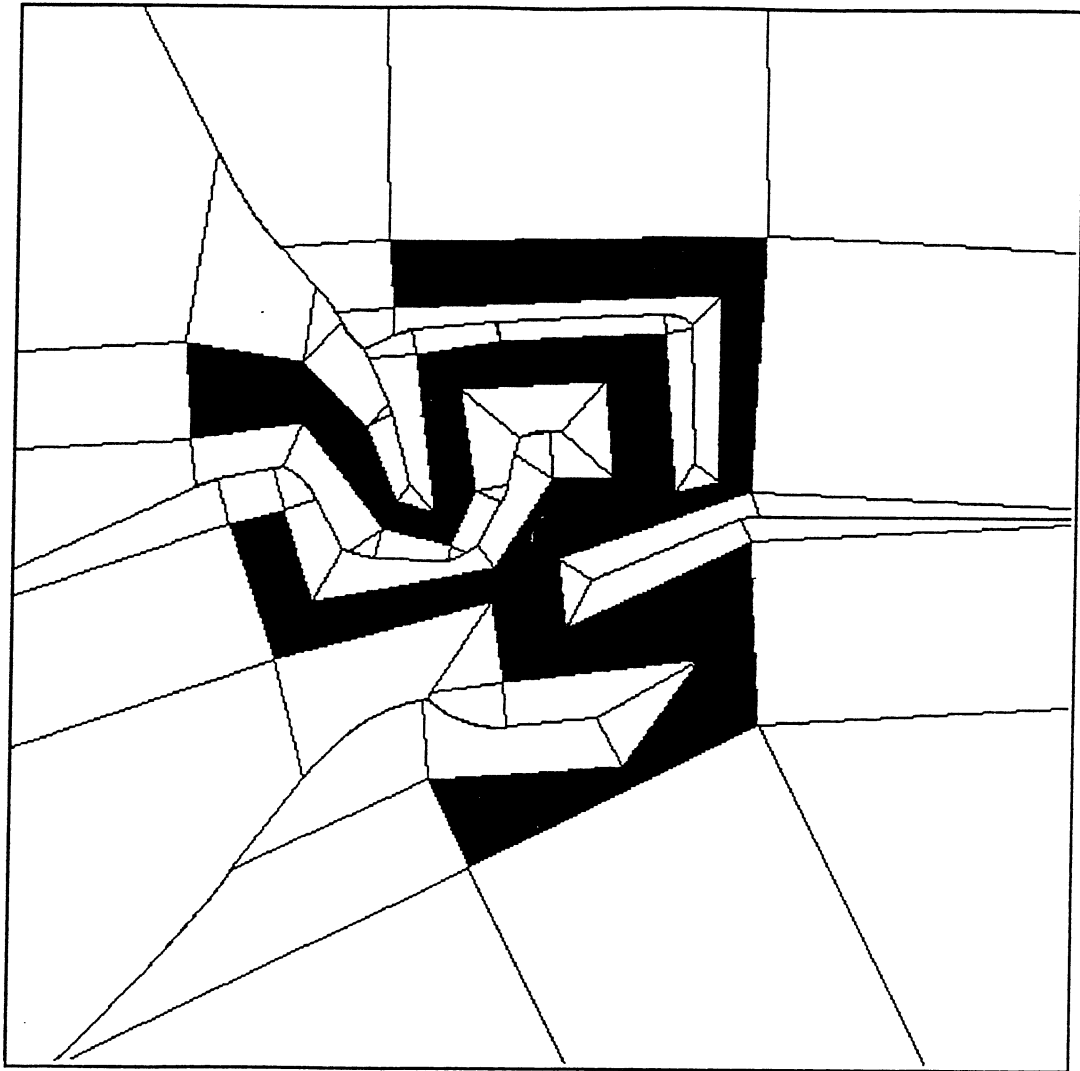


Figure 3.20 (d)

Figure 3.20 Diagrammes de Voronoï généralisés pour l'extérieur d'un polygone

Chapitre 4

Enveloppe Convexe de Polygones

4.1 Introduction

Le problème du calcul de l'enveloppe convexe d'un ensemble de n points du plan a largement été étudié. La solution optimale pour ce problème est un algorithme d'une complexité en $O(n \log n)$ [PREPARATA 88]. Notamment, il existe une littérature importante concernant la bibliographie sur le problème de convexité [RONSE 89]. Pourtant, un problème nouveau apparaît naturellement si cet ensemble de points est donné comme une liste de sommets ordonnés d'un polygone simple. Dans ce cas particulier, il a été indiqué qu'un algorithme en complexité $O(n)$ peut réaliser l'enveloppe convexe [CHEN 89] [SHIN 86].

Pour quelques applications, le calcul de l'enveloppe convexe n'est pas toujours un but final. Par exemple, nous pouvons vouloir résoudre le problème de visibilité entre deux polygones. Bien sûr, tout algorithme direct peut être appliqué. Pourtant, ce problème peut aussi être résolu par un algorithme de calcul de l'enveloppe convexe des deux polygones. C'est parce que la visibilité est en relation étroite avec la structure de données d'un polygone contenant quelque information topologique que l'on est capable de savoir que les deux chaînes internes entre deux lignes d'appui communes des deux polygones sont visibles.

L'enveloppe convexe est aussi en relation étroite avec le diagramme de Voronoï. En fait, un objet parmi un ensemble d'objets est situé à la frontière de l'enveloppe convexe de l'ensemble si et seulement si la région de Voronoï de l'objet est non bornée (le lemme 2.9). Dans le chapitre 6, nous allons étudier la construction du *diagramme de Voronoï généralisé* (DVG) d'un ensemble de polygones simples $P_i (i=1, \dots, N)$. Une étape importante dans une telle construction du DVG est de calculer un séparateur entre $\bigcup_{i=1}^{k-1} P_i$ et P_k . De plus, le séparateur est déterminé par deux chaînes internes visibles qui appartiennent respectivement à chacun des polygones. Il semble donc très important de calculer dans un premier temps l'enveloppe convexe de deux polygones simples, arbitraires et disjoints.

Nous présentons un nouvel algorithme pour calculer l'enveloppe convexe d'un ensemble de polygones. Cet algorithme est basé sur une représentation de *segments d'appui orientés*. L'algorithme est de type incrémentation. L'avantage de notre algorithme est qu'il peut donner à chaque étape d'incrément, en même temps l'enveloppe convexe et l'information concernant

l'inclusion entre les polygones associés. Nous verrons au chapitre 5 que ces deux dernières informations sont cruciales pour réaliser notre algorithme du diagramme de Voronoï généralisé d'un ensemble de polygones.

Ce chapitre est organisé comme suit. Un rappel méthodologique est fait sur les algorithmes de l'enveloppe convexe d'un polygone simple au § 4.2. Dans § 4.3, nous présentons des algorithmes pour calculer l'enveloppe convexe de deux polygones convexes. Après une présentation détaillée de deux méthodes existantes dans 4.3.1, une nouvelle méthode sera proposée. Cette nouvelle méthode consiste d'abord à construire une séquence contenant l'information de convexité des deux polygones et ensuite à trouver deux segments d'appui orientés à partir de la séquence formée. Dans § 4.4, un algorithme de calcul de l'enveloppe convexe de polygones convexes ayant une *intersection simple* est introduit. Enfin, nous discutons le problème de l'enveloppe convexe d'un ensemble de polygones simples arbitraires dans § 4.5. Après un prétraitement qui ordonne les polygones, notre algorithme introduit incrémentalement un nouveau polygone P_j à un ensemble de $j-1$ polygones $P(j-1)$. A chaque étape de l'incrémentement, l'algorithme réalise une enveloppe convexe locale jusqu'à ce que tous les polygones soient introduits.

4.2 Enveloppe convexe d'un polygone simple

4.2.1 Introduction

Par définition, l'enveloppe convexe d'un polygone donné P , est le polygone convexe minimal contenant P . Il existe beaucoup d'articles sur des algorithmes linéaires pour déterminer l'enveloppe convexe d'un polygone simple. En 1972, Sklansky [SKLANSKY 72] a proposé pour la première fois un algorithme linéaire pour calculer l'enveloppe convexe d'un polygone simple. Malheureusement, comme ce qui est indiqué par Bykat [BYKAT 78], l'algorithme ne marche pas toujours. Un autre algorithme en temps linéaire plus compliqué basé sur l'utilisation de deux piles a été proposé par McCallan et Avis avec une preuve de la correction [McCALLUM 79]. En 1982, Sklansky a donné une autre version d'algorithme linéaire [SKLANSKY 82]. Mais, des contre-exemples sont montrés successivement par Toussaint et Gindy [TOUSSAINT 83]. Graham et Yao [GRAHAM 83] a proposé un algorithme compact. Dans leurs algorithmes, une notation importante appelée *poche* ("pocket") est utilisée. Dans l'article de Shin et Woo [SHIN 86], l'idée originale de Sklansky est adoptée et un seul test de poche est utilisé. En 1989, Chen [CHEN 89] a proposé encore un nouvel algorithme compact.

Remarquons que la plupart des modèles d'algorithmes pour l'enveloppe convexe d'un polygone sont basés sur une condition implicite qu'un polygone donné est toujours simple, c'est-à-dire,

sans intersection d'arêtes du polygone sauf à leurs extrémités. Pourtant, une telle condition implicite peut ne pas être garantie durant le fonctionnement de l'algorithme. Ceci est une cause principale de non-réussite de quelques algorithmes donnés.

Par exemple, la figure 4.1.a montre un polygone avec 7 sommets. Nous allons montrer pourquoi le premier algorithme de Sklansky [SKLANSKY 72] ne réussit pas pour ce polygone.

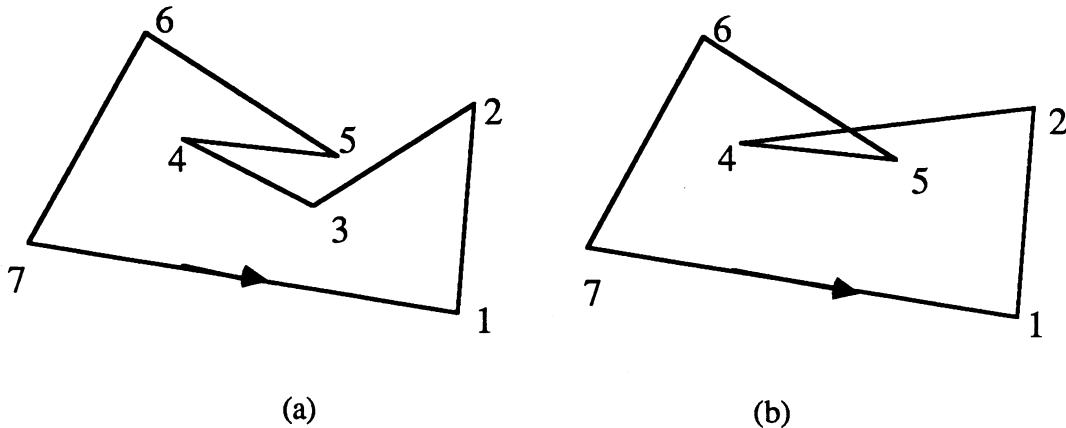


Figure 4.1 un test simple de convexité ne réussit pas pour le polygone donné
(a) polygone original; (b) l'enveloppe convexe incorrecte

L'idée principale de l'algorithme est d'explorer les sommets successifs du polygone et d'abandonner les sommets concaves afin d'obtenir finalement une liste de sommets convexes. En commençant par v_1 et en parcourant dans le sens trigonométrique, l'algorithme abandonne d'abord le sommet v_3 à cause de sa concavité. Ensuite, l'algorithme reconsidère le triplet (v_1, v_2, v_4) qui est convexe. L'algorithme donne finalement le résultat $CH(P) = (v_1, v_2, v_4, v_5, v_6, v_7)$ qui est illustré dans la figure 4.2.b. $CH(P)$ n'est ni convexe ni simple! Donc, pour résoudre le problème de l'enveloppe convexe d'un polygone simple, on doit chercher un mécanisme plus avancé que le simple test de convexité.

4.2.2 Préliminaires

Soit P un polygone simple ayant N sommets dans le plan. Sans perte de généralité, supposons que $P = (v_1, v_2, \dots, v_N)$ soit donné par une liste chaînée de sommets. Pour chaque sommet v_i nous associons un enregistrement qui contient deux champs différents: un champ de coordonnées et un champ topologique. Le champ de coordonnées contient les coordonnées x et y d'un sommet. Le champ topologique contient la relation d'adjacence entre une paire de sommets voisins. Un polygone simple est orienté dans le sens trigonométrique noté CCW si son intérieur est situé à gauche de la frontière ∂P .

Définition 4.1 L'enveloppe convexe d'un polygone simple P , notée par $CH(P)$, est le plus petit polygone convexe contenant P . Tous les sommets de $CH(P)$ sont appelés *extrêmes*.

Définition 4.2 $L(v_i, v_j)$ désigne un *segment orienté* joignant deux points v_i et v_j dans la direction de v_i à v_j .

Définition 4.3 Une *chaîne CHN* $CHN(v_i, v_j)$ est une séquence de $L(v_i, v_{i+1}), L(v_{i+1}, v_{i+2}), \dots, L(v_{j-1}, v_j)$ sur ∂P dans l'ordre CCW.

Définition 4.4 Une région fermée délimitée par $L(v_i, v_j)$ et $CHN(v_i, v_j)$ est appelée *poche* et notée par $PKT(v_i, v_j)$ si tous les sommets de $CHN(v_i, v_j)$ sont sur ou à gauche de $L(v_i, v_j)$ avec la contrainte qu'il existe au plus deux sommets appartenant à $L(v_i, v_j)$.

Deux propriétés intéressantes associées à la notion de poche dues à Graham et Yao [GRAHAM 83] sont rappelées.

Lemme 4.1 Aucun sommet de P situé dans $PKT(v_i, v_j)$ ne peut être extrême à l'exception de v_i et v_j [GRAHAM 83].

Lemme 4.2 Il existe un *successeur* de v_j dans P qui est situé en dehors de $PKT(v_i, v_j)$ [GRAHAM 83].

Définition 4.5 Le premier successeur de v_j situé en dehors de $PKT(v_i, v_j)$ est appelé *sommet d'émergence* de $PKT(v_i, v_j)$ et noté par v_{j^*} .

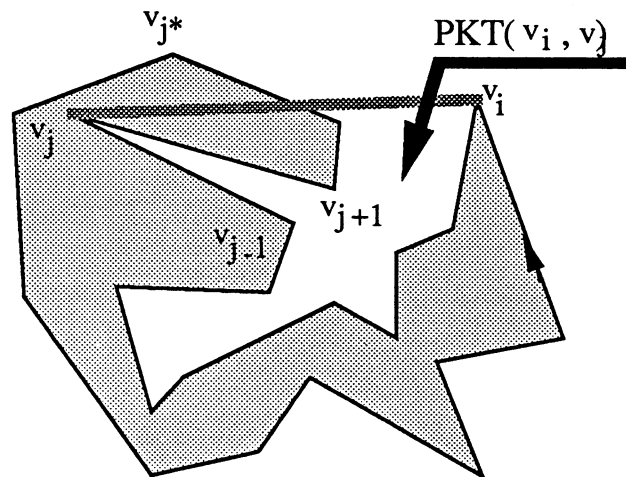


Figure 4.2 Illustration d'une poche

Lemme 4.3 Le sommet v_{j+1} est situé à l'intérieur de $PKT(v_i, v_j)$ si v_{j+1} est à gauche de $L(v_i, v_j)$ et à droite de $L(v_{j-1}, v_j)$.

Preuve. D'après la définition d'une poche, nous savons que tous les points situés dans la poche $PKT(v_i, v_j)$ doivent être situés à gauche du segment $L(v_i, v_j)$. De plus, la propriété d'un polygone simple indique que v_{j+1} situé dans de $PKT(v_i, v_j)$ doit être à droite de $L(v_{j-1}, v_j)$ (voir figure 4.2).

Lemme 4.4 Si le sommet v_k est dans $PKT(v_i, v_j)$ alors v_{k+1} est situé aussi dans $PKT(v_i, v_j)$ à moins que v_{k+1} soit situé à droite de $L(v_i, v_j)$.

Preuve. Comme P est simple, une seule possibilité pour que v_{k+1} devienne un sommet extrême de $PKT(v_i, v_j)$ est qu'il soit à droite de $L(v_i, v_j)$.

Définition 4.6 Une poche est *maximale* si elle n'est pas contenue dans une autre.

Lemme 4.5 Les sommets v_q et v_r ($r \neq q$) sont deux sommets consécutifs de l'enveloppe convexe $CH(P)$ si et seulement si $PKT(v_q, v_r)$ est une poche maximale.

Preuve. Soit h le demi-plan situé à gauche du segment orienté $L(v_q, v_r)$ et contenant $CH(P)$, h^* le complément de h (voir figure 4.3). Si $PKT(v_q, v_r)$ est maximale alors tous les successeurs de v_r sont situés dans le demi-plan h (facile à prouver). Par conséquent, P est situé dans h . De plus, comme v_r et v_q appartiennent à P , alors $L(v_q, v_r) \in CH(P)$. Maintenant, supposons que v_q et v_r sont deux sommets consécutifs de $CH(P)$. Nous allons prouver que $PKT(v_q, v_r)$ est maximale. Si $PKT(v_q, v_r)$ n'est pas maximale alors il existe un successeur v_x de v_r tel que $PKT(v_q, v_x) \supset PKT(v_q, v_r)$. La seule possibilité de la position de v_x est qu'il soit situé dans le demi-plan h^* (sinon une telle poche $PKT(v_q, v_x)$ n'existe pas). Comme v_x est situé à droite de $L(v_q, v_r)$, donc $L(v_q, v_r)$ sépare P en deux parties différentes non vides. Par conséquent, v_q et v_r ne sont pas sommets consécutifs de $CH(P)$. Ceci est en contradiction avec la supposition que v_q et v_r sont deux sommets consécutifs de $CH(P)$. Donc, $PKT(v_q, v_r)$ est maximale.

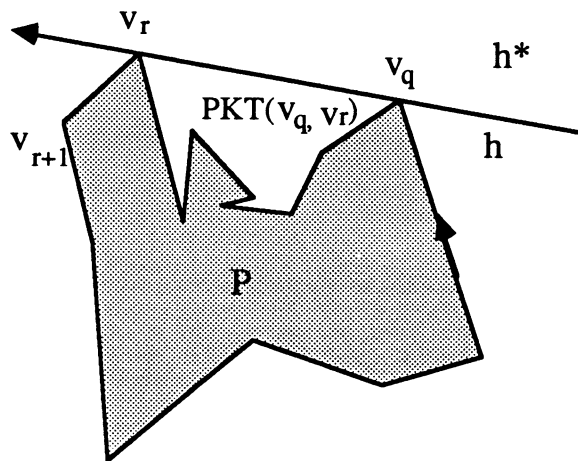


Figure 4.3 Une poche maximale

Corollaire 4.1 Soit $PKT(v_q, v_r)$ ($1 \leq q < r \leq N$, ou $r=1$ si $q=N$) les poches maximales d'un polygone simple donné. L'enveloppe convexe du polygone est déterminée par la séquence des $L(v_q, v_r)$ ($1 \leq r < q \leq N$, où $q=1$ si $r=N$).

Preuve. Soit $PKT(v_r, v_q)$ une poche initiale maximale d'un polygone simple P . Comme P est fermé, on peut trouver un successeur, noté v_x , de v_r tel que $PKT(v_r, v_x)$ est aussi maximale. De même manière, un successeur de v_x doit exister si bien qu'une nouvelle poche maximale soit trouvée. La poche maximale suivante est similaire. Nous construisons toutes les poches maximales $PKT(v_s, v_t)$ jusqu'à $v_t = v_q$. D'après le lemme 4.5, la séquence ordonnée $(v_q, v_r, \dots, v_x, \dots, v_s)$, déterminée par la séquence des poches maximales, correspond à $CH(P)$.

4.2.3 Algorithme

Un algorithme pour calculer l'enveloppe convexe d'un polygone simple est le suivant. Un polygone initial est représenté par une liste doublement chaînée des sommets et il est orienté selon le sens CCW. La structure de données utilisée est une pile doublement chaînée dénommée $HULL = \langle h_1, h_2, \dots, h_t \rangle$ où h_1 représente le bas de la pile et h_t est le haut de celle-ci. Deux opérations sont effectuées sur cette pile. D'abord, l'opération $PUSH(p)$ met un point p au haut de la pile et déplace le pointeur de la pile vers ce nouvel élément. Deuxièmement, l'opération POP élimine l'élément situé au haut de la pile et déplace le pointeur vers l'élément précédent. Lorsque l'algorithme est terminé, la liste doublement chaînée restant dans la pile $HULL$ donne l'enveloppe convexe du polygone initial.

Introduisons quelques notations pour décrire l'algorithme:

p :	sommet courant dans le polygone initial;
q :	sommet inséré dernièrement dans la pile;
$LAST(x)$:	sommet précédent de x ;
$NEXT(x)$:	sommet successif de x ;
LID :	segment $L(h_{t-1}, h_t)$ correspondant à $PKT(h_{t-1}, h_t)$ où h_t est le haut de la pile ;
EG :	segment $L(p_1, p_2)$ où $p_1 = LAST(LAST(p))$ et $p_2 = LAST(p)$.

Algorithme HULL_POLYGON

Début

1. {initialisation}
 - a. trouver un sommet p_s appartenant à $CH(P)$. Par exemple, p_s est l'un des quatre sommets extrêmes du polygone initial;
 - b. {construire la première poche }
 - $PUSH(p_s); p \leftarrow NEXT(p_s); \{t=1\}$
 - $PUSH(p); p \leftarrow NEXT(p); \{t=2\}$
2. tant que $p \neq p_s$ faire
 - début

```

a. si  $p$  est à gauche de  $LID$  alors {  $p$  est dans le demi-plan  $h$  }
    si  $p$  est à gauche de  $EG$ 
        {  $p$  est un sommet d'émergence de la poche courante; lemme 4.3 }
    alors
    début
         $PUSH(p); \quad \{t = t+1\}$ 
         $p \leftarrow NEXT(p);$ 
    fin
    sinon
        { trouver un sommet d'émergence de la poche courante; lemme 4.4 }
    tant que  $p$  n'est pas situé à droite de  $LID$  faire
         $p \leftarrow NEXT(p);$ 
b. sinon {  $p$  est dans le demi-plan  $h^*$ , modifier la poche courante }
    début
        tant que  $LAST(q)$  est situé à gauche de  $L(p, q)$  faire
             $POP(q); \quad \{t = t-1\}$ 
             $PUSH(p); \quad \{t = t+1\}$ 
             $p \leftarrow NEXT(p);$ 
        fin;
    fin;
Fin;

```

4.3 Enveloppe convexe de deux polygones convexes

Dans ce qui suit, nous allons discuter du problème suivant: étant donné deux polygones simples disjoints, trouver l'enveloppe convexe de leur union. La figure 4.4 illustre un tel exemple pour deux polygones P et Q . Nous remarquons que l'enveloppe convexe de P et celle de Q peuvent s'intersecter. En principe, chacune des enveloppes convexes des deux polygones est d'abord calculée et ensuite nous calculons l'enveloppe convexe de ces deux polygones convexes.

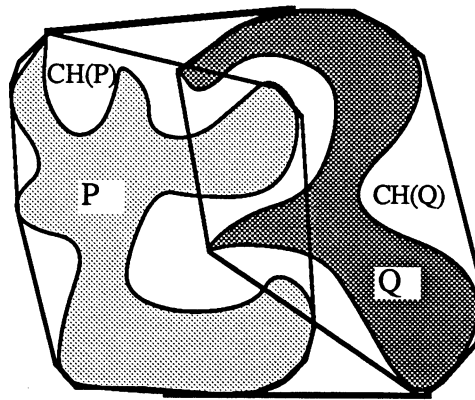


Figure 4.4 L'enveloppe convexe $CH(P \cup Q)$ de deux polygones simples disjoints P et Q . Intersection peut avoir lieu entre $CH(P)$ et $CH(Q)$

4.3.1 Deux méthodes pour deux polygones convexes

4.3.1.1 Méthode basée sur le balayage de Graham

La première méthode pour calculer l'enveloppe convexe de la réunion de deux polygones convexes consiste à calculer l'enveloppe convexe d'un ensemble de points correspondant à la réunion des sommets issus des deux polygones. Cette approche a été proposée par Shamos [SHAMOS 78] dans le cas où les polygones sont convexes. La méthode est composée de deux étapes:

Etape 1: Fusionner les deux listes des sommets de P et Q afin de former une liste nouvelle en ordonnant tous les sommets dans l'ordre angulaire par rapport à quelque point interne;

Etape 2: Appliquer l'algorithme du balayage de Graham [GRAHAM 72] à la nouvelle liste de sommets obtenue pour trouver $CH(P \cup Q)$.

Il faut remarquer qu'une telle approche n'intègre pas d'information topologique entre les deux polygones. Remarquons également aussi que les deux polygones convexes peuvent s'intersecter dans cette méthode.

4.3.1.2 Méthode basée sur les lignes d'appui

Le calcul de l'enveloppe convexe n'est pas seulement un problème isolé mais constitue souvent un sous-problème important dans d'autres problèmes géométriques. Par exemple, le problème de la visibilité entre polygones peut être résolu à l'aide du calcul de l'enveloppe convexe. Nous présentons une autre méthode pour calculer l'enveloppe convexe de deux polygones convexes. Cette méthode est basée sur le calcul de lignes d'appui communes de polygones P et Q .

Définition 4.7 Une *ligne d'appui* d'un polygone P est une droite l ayant au moins un point commun avec P appelé *point d'appui* et telle que tous les sommets de P sont situés entièrement d'un même côté de l .

Une définition similaire pour une ligne d'appui de deux polygones P et Q peut être donnée. De plus, la *ligne d'appui commune* est une ligne d'appui de P et Q ayant respectivement au moins un point d'intersection avec P et un autre point d'intersection avec Q . Trouver les lignes d'appui de P et Q peut être résolu en balayant la liste de sommets de $CH(P \cup Q)$ et en cherchant une paire de sommets consécutifs v_i et v_{i+1} de $CH(P \cup Q)$ tels que v_i appartient à P et v_{i+1} appartient à Q . v_i et v_{i+1} identifient une ligne d'appui commune entre P et Q .

La technique basée sur les lignes d'appui pour calculer l'enveloppe convexe de deux polygones convexes disjoints est d'abord développée par Preparata-Hong [PREPARATA 77]. Ils déterminent en temps linéaire les deux lignes d'appui communes des deux polygones. Dans leur terminologie, ces deux lignes d'appui communes sont appelées respectivement "*left tangent*" et "*right tangent*" (cf. la figure 4.5). Une fois que les deux lignes d'appui communes sont trouvées, l'enveloppe convexe des deux polygones peut s'obtenir en liant les deux chaînes polygonales extérieures avec les lignes d'appui et en éliminant les sommets de P et Q qui deviennent intérieurs au polygone résultant.

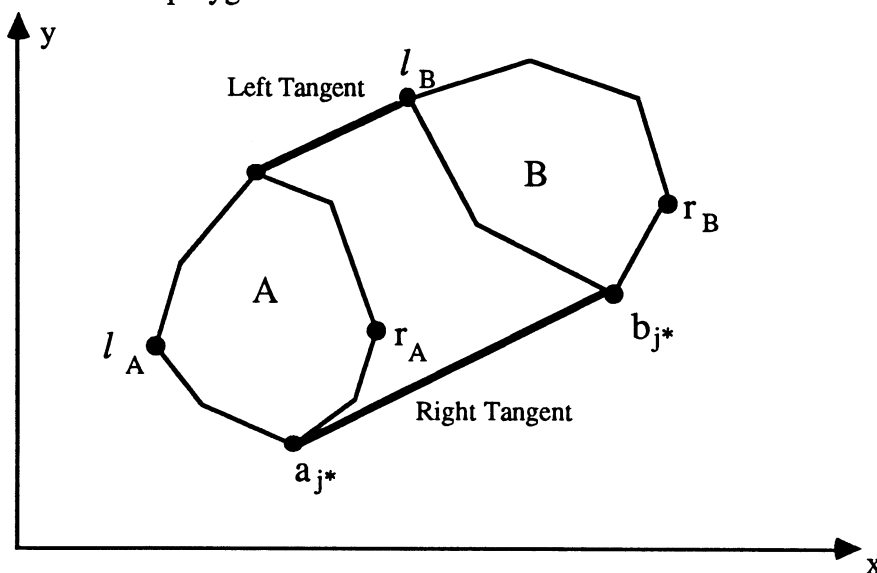


Figure 4.5 "*Left tangent*" et "*right tangent*" de deux polygones convexes
(d'après [PREPARATA 77])

$$x(l_A) = \min\{x(a_i), a_i \in A\}, \quad x(l_B) = \min\{x(b_j), b_j \in B\}$$

$$x(r_A) = \max\{x(a_i), a_i \in A\}, \quad x(r_B) = \max\{x(b_j), b_j \in B\}$$

Détaillons leur algorithme. Soit A et B deux polygones convexes disjoints. Preparata et Hong ont utilisé trois critères de pente pour caractériser les deux points d'appui a et b appartenant

respectivement à A et B . A la fin du traitement, le segment (a, b) correspond à une ligne d'appui commune.

Etant donné deux points u et v , une valeur de pente, notée par $sl(u, v)$, est définie par:

$$sl(u, v) = (y(u)-y(v))/(x(u) - x(v))$$

Trois types de pente sont introduits:

$$\alpha_{i,i+1} = sl(a_i, a_{i+1}), \quad \beta_{j,j+1} = sl(b_j, b_{j+1}) \quad \text{et} \quad \gamma_{ij} = sl(a_i, b_j)$$

où $a_i \in A$ et $b_j \in B$.

Afin d'assurer la condition de descente monotone stricte des séquences $(\alpha_{12}, \alpha_{23}, \dots)$ et $(\beta_{12}, \beta_{23}, \dots)$, un prétraitement est nécessaire pour obtenir deux chaînes $CHN1$ et $CHN2$ satisfaisant les contraintes suivantes:

$$CHN1 = (a_1, a_2, \dots, l_A) \text{ avec } x(a_i) \geq x(a_{i+1}) \text{ où } a_i \in A$$

et

$$CHN2 = (b_1, b_2, \dots, l_B) \text{ avec } x(b_j) \geq x(b_{j+1}) \text{ où } b_j \in B.$$

Ce sont ces deux chaînes qui sont prises comme entrée de l'algorithme.

Preparata et Hong [PREPARATA 77] ont déclaré que l'algorithme suivant détermine uniquement a_{i^*} et b_{j^*} .

Algorithme RT (right tangent)

entrée: Coordonnées de (a_1, a_2, \dots, l_A) et (b_1, b_2, \dots, l_B) , et pentes $(\alpha_{12}, \alpha_{23}, \dots)$ et $(\beta_{12}, \beta_{23}, \dots)$.

sortie: i^*, j^* , les indices des extrémités du segment "right tangent".

RT1. mettre $i \leftarrow 1, j \leftarrow 1$;

RT2. calculer $\gamma_{ij} \leftarrow (y(a_i)-y(b_j)) / (x(a_i)-x(b_j))$;

RT3. si $\alpha_{i,i+1} \geq \gamma_{ij}$, mettre à jour $i \leftarrow i+1$ et aller à RT2;

RT4. si $\beta_{j,j+1} \geq \gamma_{ij}$, mettre à jour $j \leftarrow j+1$ et aller à RT2;

RT5. mettre $i^* \leftarrow i, j^* \leftarrow j$.

Il est évident que nous devons redéfinir l_A et r_A (respectivement l_B et r_B) quand deux polygones ne s'étendent pas dans une direction horizontale relativement (voir la figure 4.6). Dans ce cas-

ci, nous devons déterminer l_A et r_A (respectivement l_B et r_B) par y-coordonnées au lieu de x-coordonnées si bien que deux séquences monotones de sommets puissent être trouvées. Cependant, la détermination de la position relative entre deux polygones n'est pas facile!

On dit qu'un algorithme est *entrée-contrainte* si ses variables d'entrée sont conditionnées pas des contraintes sous quelque condition. Evidemment, l'algorithme ci-dessus est *entrée-contrainte*.

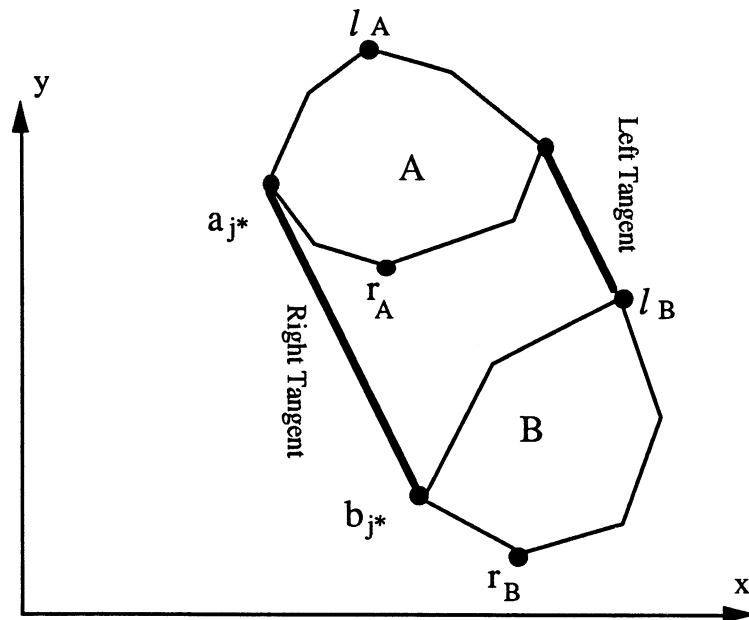


Figure 4.6 Rotation de 90° de la figure 4.5, deux chaînes *CHN1* et *CHN2* doivent être retrouvées par y-coordonnée au lieu de x-coordonnée

4.3.2 Un nouvel algorithme — libre de contraintes

Nous allons proposer ci-après un nouvel algorithme pour trouver l'enveloppe convexe de deux polygones convexe. Il sera montré dans 4.4 que les polygones convexe ne sont pas nécessairement disjoints. Il sera montré que l'algorithme présenté est entrée-libre et invariant à la position relative entre les deux polygones.

4.3.2.1 Préliminaires

Soit P et Q deux polygones convexe disjoints. Supposons que P a n sommets et Q a m sommets. Par ailleurs, les polygones sont donnés sous forme standard. C'est-à-dire, les sommets de chaque polygone sont spécifiés par les coordonnées cartésiennes, sont liés en une liste dans le sens CCW, et aucun triplet de trois sommets consécutifs n'est colinéaire. Pour étiqueter les deux polygones, nous identifions l'un des deux polygones comme le premier et l'autre comme le second.

Théorème 4.1 Le segment de droite l défini par deux points est une arête de l'enveloppe convexe si et seulement si tous les autres points de l'ensemble sont situés dans l ou d'un même côté de celui-ci (Théorème 3.8 dans [PREPARATA 88]).

Ce théorème sera à la base de notre nouvel algorithme. Avant la présentation de l'algorithme, nous avons d'abord besoin de quelques définitions et lemmes.

Définition 4.8 Soit $P = (p_0, p_1, \dots, p_{n-1})$ et $Q = (q_0, q_1, \dots, q_{m-1})$ deux polygones convexes. Le polygone P (respectivement Q) peut se représenter par $E = (e_0, e_1, \dots, e_{n-1})$ où e_i est un vecteur défini par les sommets successifs p_{i-1} et p_i ($i=i$ module (n)). Pour chaque vecteur v appartenant à la représentation du polygone P , nous associons une valeur donnée par la fonction:

$$L_Q(v) = \begin{cases} +1 & \text{si tous les points de } Q \text{ sont sur ou à gauche de } v \\ -1 & \text{autrement} \end{cases}$$

La figure 4.7 donne l'illustration de cette fonction. Le graphe associé à la fonction $L_Q(v)$ avec v appartenant à P correspondant à la figure 4.7 est représenté en figure 4.8.

Evidemment, la fonction $L_Q(v)$ possède les propriétés suivantes:

Propriété 1: $v \in \partial CH(P \cup Q)$ si et seulement si $L_Q(v) = 1$ avec $v \in P$.

Propriété 2: $L_Q(v)$ est une fonction périodique avec une période égale au nombre de sommets de P .

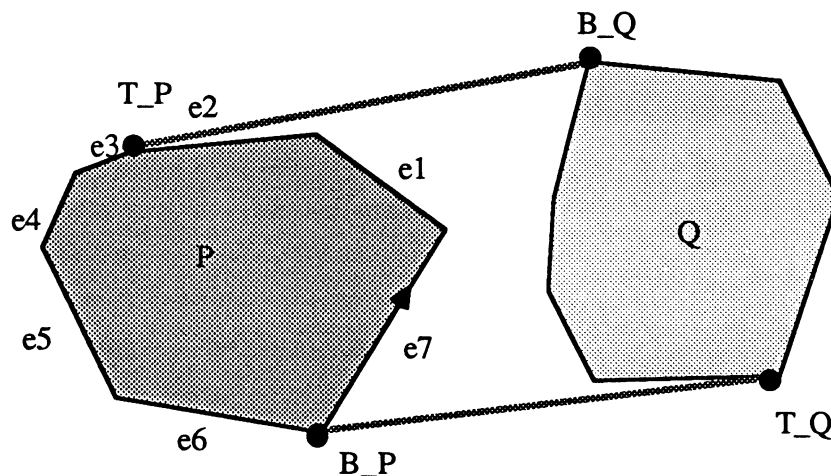


Figure 4.7 Illustration de la définition 4.8 : deux polygones convexes disjoints, P possède 7 arêtes

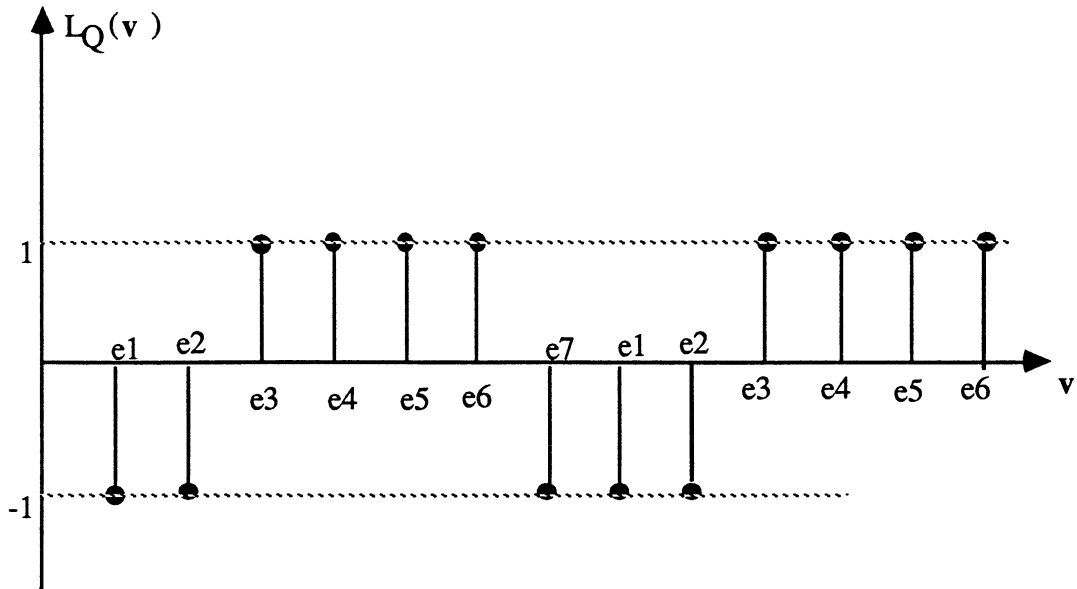


Figure 4.8 Le graphe de $L_Q(v)$ correspondant aux polygones dans figure 4.7

Définition 4.9 La séquence différentielle de la séquence $S(n)$, notée par $\Delta S(n)$, est définie par:

$$\Delta S(n) = S(n) - S(n-1).$$

Selon cette définition, $\Delta L_Q(v)$ ($v \in P$) ne peut prendre que les valeurs suivantes:

$$\Delta L_Q(v) (v \in P) = \begin{cases} 2 \\ 0 \\ -2 \end{cases}$$

Grâce à la convexité des polygones donnés, il existe au plus un point e_T où $\Delta L_Q(e_T) = 2$ et au plus un autre point e_B où $\Delta L_Q(e_B) = -2$. Nous appellerons $T_P = e_T \cap e_{T-1}$ un point d'appui positif et $B_P = e_B \cap e_{B-1}$ un point d'appui négatif. Il est évident qu'une chaîne extérieure de P , $C1 = (e_T, e_{T+1}, \dots, e_{B-1})$ avec $e_T \in P$ est une séquence d'arêtes de $CH(P \cup Q)$. De même, nous utilisons les notations T_Q et B_Q pour représenter les deux points d'appui de Q . En fait, $S1 = (B_P, T_Q)$ et $S2 = (B_Q, T_P)$ sont justement les deux lignes d'appui communes à déterminer.

4.3.2.2 Raffinement de l'algorithme

Nous avons supposé qu'il existe deux points d'appui différents dans chacun des deux polygones originaux. Cependant, cette supposition n'est pas toujours vrai en réalité. Considérons l'exemple de la figure 4.9. En ce cas-ci, il existe seulement un point d'appui dans P . En outre, ce point d'appui ne peut pas être trouvé par la séquence décrite dans la définition 4.9. Précisons comment trouver un point d'appui dans ce cas-ci.

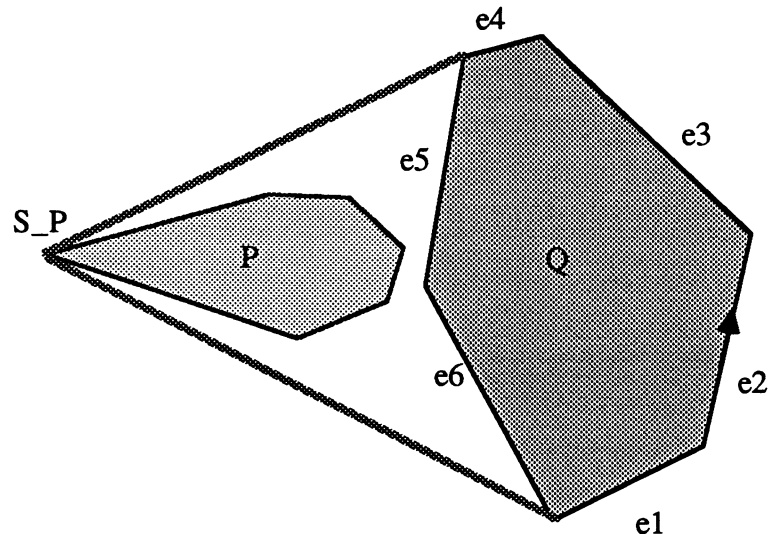


Figure 4.9 Seul un point d'appui dans le polygone P qui ne peut pas être trouvé par la séquence différentielle

Lemme 4.6 Soit P et Q deux polygones convexes disjoints. Il existe deux points d'appui dans Q (respectivement dans P) si P possède un seul point d'appui (respectivement Q).

Preuve. Soit P le polygone ayant seul point d'appui. Nous supposons que Q ne possède pas deux points d'appui définis par la séquence différentielle. D'après cette supposition nous obtenons immédiatement une conclusion comme quoi aucune arête de Q n'appartient à $CH(P \cup Q)$. Du fait que P possède un seul point d'appui, aucune arête de P n'appartient à $CH(P \cup Q)$, non plus. Cependant, il est impossible que toutes les arêtes de P et de Q soient situées à l'intérieur de $CH(P \cup Q)$. Donc, la supposition est fausse.

Définition 4.10 Soit P et Q deux polygones convexes. Un sommet de P , noté par v_i , est appelé *point d'appui positif* si $\Delta L_Q(v_i) = 2$ et est appelé *point d'appui négatif* si $\Delta L_Q(v_i) = -2$ où $v_i = (v_{i-1}, v_i)$.

Lemme 4.7 Etant donné deux polygones convexes P et Q . Si P (respectivement. Q) contient un point d'appui positif alors il possède aussi un point d'appui négatif.

Preuve. Comme P est fermé, $L_Q(v)$ avec $v \in P$ est une séquence périodique quand v_i varie suivant ∂P dans le sens CCW. La périodicité de $L_Q(v)$ implique que les points d'appui de P définis par la séquence différentielle apparaissent en paires, c'est-à-dire, s'il y a un point d'appui positif alors il y a aussi un autre point d'appui négatif.

Comme P n'est pas inclus à l'intérieur de Q alors il existe au moins un sommet de P appelé *le point d'ancrage* qui est situé dans $\partial CH(P \cup Q)$ même si P ne possède pas de point d'appui défini

selon la définition 4.10. Le lemme 4.6 déclare l'existence de deux points d'appui différents de Q dans le cas où P ne possède qu'un point d'appui. En outre, le lemme 4.7 affirme que les deux points d'appui d'un polygone ne sont pas identiques s'ils existent. Dans le cas où P ne possède pas un tel point d'appui défini par la définition 4.10, nous pouvons trouver facilement le point d'ancrage à l'aide de deux points d'appui de Q . Ceci peut être fait par le test de l'aire d'un triangle déterminé par T_Q, B_Q et q où T_Q et B_Q sont deux points d'appui de Q (voir figure 4.7) et q un sommet appartenant à P . Le sommet q correspondant à un triangle ayant l'aire maximale donne le point d'ancrage désiré.

En résumé, nous pouvons donner ci-dessous notre algorithme pour calculer l'enveloppe convexe de deux polygones convexes disjoints.

Algorithme HULL_TWO_DISJOINT_CONVEX_POLYGONS ;

Début

1. trouver deux points d'appui de P , notés par T_P (le point d'appui positif) et B_P (le point d'appui négatif), par analyse de la séquence $\Delta L_Q(v)$ ($v \in P$);
2. trouver deux points d'appui de Q , notés par T_Q (le point d'appui positif) et B_Q (le point d'appui négatif), en analysant la séquence $\Delta L_P(v)$ ($v \in Q$);
3. si $T_P = \phi$ (on a aussi $B_P = \phi$) et $T_Q \neq \phi$ alors
 - a. trouver le point d'ancrage de P ;
 - b. $T_P \leftarrow$ point d'ancrage; $B_P \leftarrow$ point d'ancrage;
 - c. aller à 5;
4. si $T_Q = \phi$ (on a aussi $B_Q = \phi$) et $T_P \neq \phi$ alors
 - a. trouver le point d'ancrage de Q ;
 - b. $T_Q \leftarrow$ point d'ancrage; $B_Q \leftarrow$ point d'ancrage;
5. lier les deux chaînes $C_P = \langle T_P, \dots, B_P \rangle$ dans P et $C_Q = \langle T_Q, \dots, B_Q \rangle$ de Q en utilisant deux lignes d'appui (B_P, T_Q) et (B_Q, T_P) pour former l'enveloppe convexe de $P \cup Q$;

Fin;

Maintenant, nous analysons la complexité de l'algorithme proposé. Supposons que P a n sommets et Q a m sommets. L'étape 1 peut se faire en $O(n)$ avec un prétraitement pour la séquence binaire, par simple test de deux points d'appui. De même, un test en $O(m)$ avec un prétraitement est nécessaire pour l'étape 2. L'étape 3 et l'étape 4 peuvent être faites en temps

$O(n)$ et $O(m)$ respectivement. Donc, le point d'ancrage peut être trouvé en $O(\max(n, m))$ dans le cas le plus défavorable. L'étape 5 est une opération simple de pointeur. Par conséquent, le temps total d'exécution est proportionnel au nombre total des sommets de deux polygones du cas le plus défavorable avec un prétraitement.

L'algorithme proposé ici possède les avantages suivants :

(a) L'algorithme est très simple et son code est très court en utilisant la fonction différentielle définie;

(b) Pour trouver les points d'appui, il suffit de parcourir la frontière d'un polygone. Ceci est une opération simple de pointeur. Les points d'appui obtenus fournissent une information importante pour construire un séparateur (voir chapitre 5);

(c) L'algorithme est *entrée-libre* car l'algorithme ne se préoccupe pas de la position relative des deux polygones.

(d) On vérifiera dans le paragraphe suivant, que l'algorithme peut s'appliquer aussi à deux polygones appelés *polygones d'intersection simple* (voir ci-dessous définition 4.11).

4.4 Enveloppe convexe de deux polygones disjoints

Dans ce paragraphe, nous allons étendre l'algorithme de l'enveloppe convexe de deux polygones convexes disjoints proposé précédemment au cas où les deux polygones ne sont pas nécessairement convexes. Pour cela, nous introduisons tout d'abord un type de polygones convexes appelés *polygones d'intersection simple*. Puis, un algorithme correspondant pour l'enveloppe convexe de deux polygones de ce type est proposé. Enfin, en utilisant la propriété: "si P et Q sont deux polygones disjoints alors $CH(P)$ et $CH(Q)$ sont d'intersection simple" (lemme 4.8), l'enveloppe convexe de deux polygones simples se ramène à celle de deux polygones d'intersection simple.

4.4.1 Deux polygones convexes d'intersection simple

Définition 4.11 Deux polygones convexes P et Q sont d'*intersection simple* si et seulement l'une des deux conditions suivantes est satisfait:

(a) $P \cap Q = \emptyset$, c'est-à-dire, P et Q sont disjoints;

(b) si $P \cap Q \neq \emptyset$, le nombre d'intersection entre les arêtes "ouvertes" des polygones est égal à 2 .

Avec la définition ci-dessus il n'existe pas d'intersection si un sommet du polygone P appartient à une arête du polygone Q .

Des exemples de polygones convexes d'intersection simple sont montrés en figure 4.10.

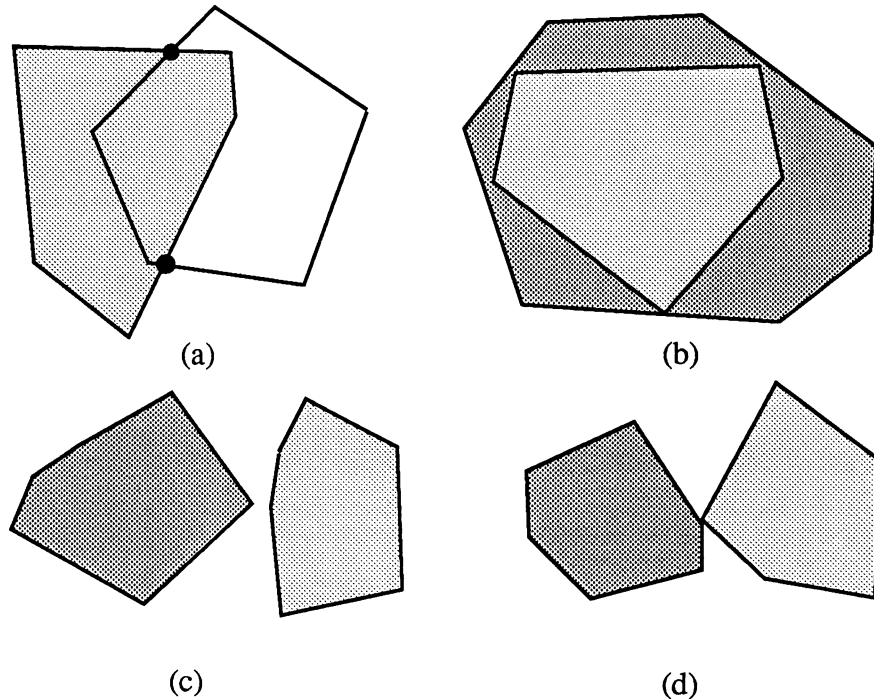


Figure 4.10 Polygones convexes d'intersection simple
(a) avec 2 points d'intersection; (b) (c) (d) avec 0 point d'intersection

4.4.2 Enveloppe convexe de polygones convexes d'intersection simple

Selon la définition de deux polygones convexes d'intersection simple, on a trois classes suivantes:

- (1) P et Q sont disjoints ;
- (2) P est inclus entièrement dans Q ou le contraire ;
- (3) P et Q s'intersectent seulement en deux points.

Examinons comment modifier l'algorithme proposé en §4.3.2.2 pour qu'il puisse s'appliquer aux polygones d'intersection simple de la seconde classe et de la troisième classe.

Supposons que P est inclus entièrement dans Q (figure 4.11.a). Revenons aux séquences $L_Q(v)$ ($v \in P$) et $L_P(v)$ ($v \in Q$). Les graphes de ces deux séquences sont fournis en figure 4.11.b.

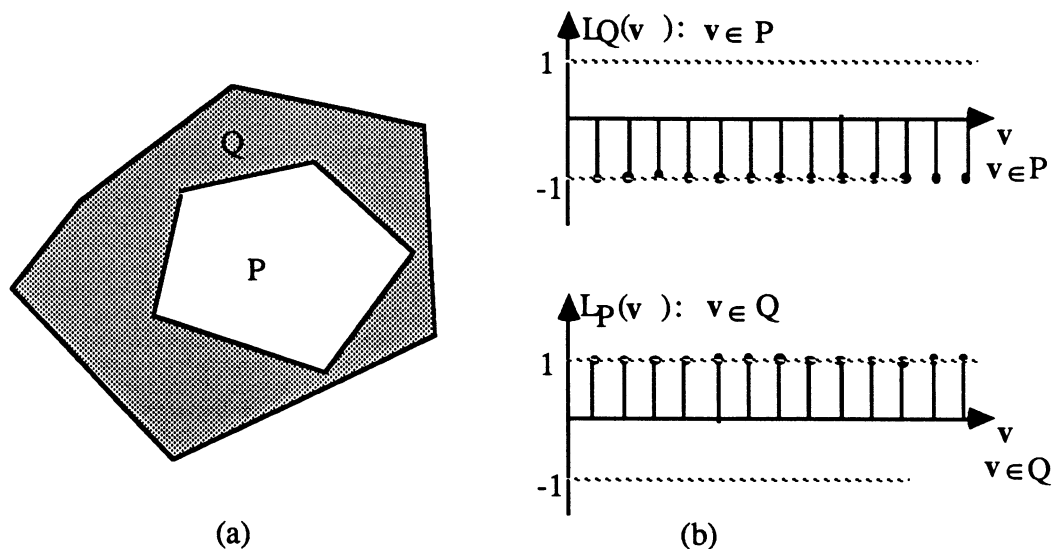


Figure 4.11 (a) Polygone P est inclus dans polygone Q ;
 (b) les graphes des séquences $L_Q(v) | v \in P$ et $L_P(v) | v \in Q$

D'après l'exemple ci-dessus, il n'existe pas de point d'appui dans les deux séquences différentielles correspondantes si un polygone est inclus dans l'autre (la seconde classe). Cette observation nous permet de résoudre le problème d'inclusion en détectant les points d'appui. Précisément, si les points d'appui n'existent pas et si $L_Q(v) | v \in P = 1$ alors P est inclus dans Q . Sinon Q est inclus dans P . Pour la troisième classe des polygones convexes d'intersection simple, deux lignes d'appui communes existent à cause de la propriété de fermeture des polygones. De plus, les deux lignes d'appui communes peuvent être déterminées soit par quatre points d'appui soit par deux points d'appui et un autre point d'ancrage. Donc, l'algorithme HULL_TWO_DISJOINT_CONVEX_POLYGONS proposé est aussi applicable à la troisième classe des polygones convexes d'intersection simple.

D'après la discussion ci-dessus, nous savons que l'algorithme de l'enveloppe convexe de deux polygones convexes disjoints peut s'appliquer avec un peu de modification à deux arbitraires polygones convexes d'intersection simple. Nous réécrivons ci-dessous l'algorithme avec la modification dénommé HULL_TWO_SIMPLE_INTERSECTION_CONVEX. Et nous avons:

Théorème 4.1 L'enveloppe convexe de deux polygones convexes d'intersection simple peut être trouvée en temps proportionnel au nombre total de sommets .

Algorithme HULL_TWO_SIMPLE_INTERSECTION_CONVEX;

entrée: deux polygones d'intersection simple P et Q ;

sortie: l'enveloppe convexe de P et Q , $CH(P \cup Q)$;

P_dans_Q, Q_dans_P : booléen;

Début

1. trouver deux points d'appui de P , notés par T_P (le point d'appui positif) et B_P (le point d'appui négatif), en analysant la séquence $\Delta L_Q(v)$ ($v \in P$);
2. trouver deux points d'appui de Q , notés par T_Q (le point d'appui positif) et B_Q (le point d'appui négatif), en analysant la séquence $\Delta L_P(v)$ ($v \in Q$);
3. si $T_P = \phi$ (on a aussi $B_P = \phi$) et $T_Q \neq \phi$ alors
début
 - a. trouver le point d'ancrage de P ;
 - b. $T_P \leftarrow$ point d'ancrage; $B_P \leftarrow$ point d'ancrage;
 fin;
4. sinon si $T_Q = \phi$ (on a aussi $B_Q = \phi$) et $T_P \neq \phi$ alors
début
 - a. trouver le point d'ancrage de Q ;
 - b. $T_Q \leftarrow$ point d'ancrage; $B_Q \leftarrow$ point d'ancrage;
 fin;
5. sinon si $T_P = \emptyset$ et $B_Q = \emptyset$ alors
 - a. si $L_Q(v) = 1$ pour un vecteur v de P alors
début
 - 1) $P_{\text{dans}_Q} \leftarrow$ vrai;
 - 2) $CH(P \cup Q) \leftarrow Q$;
 fin;
 - b. sinon
début
 - 1) $Q_{\text{dans}_P} \leftarrow$ vrai;
 - 2) $CH(P \cup Q) \leftarrow P$;
 fin;
6. si $P_{\text{dans}_Q} = \text{faux}$ et $Q_{\text{dans}_P} = \text{faux}$ alors lier deux chaînes $C_p = \langle T_P, \dots, B_P \rangle$ dans P et $C_q = \langle T_Q, \dots, B_Q \rangle$ dans Q en utilisant deux lignes d'appui (B_P, T_Q) and (B_Q, T_P) pour former l'enveloppe convexe de P et Q , $CH(P \cup Q)$;

Fin;

4.4.3 Enveloppe convexe de deux polygones simples disjoints

Dans ce paragraphe, nous supposons que P et Q sont deux polygones simples disjoints.

Lemme 4.8 Si P et Q sont deux polygones simples disjoints alors $CH(P)$ et $CH(Q)$ sont d'intersection simple.

Preuve. Afin de prouver le lemme, nous montrons d'abord que le nombre de points d'intersection entre deux polygones convexes est certainement pair. Ceci est dû au fait que les polygones sont fermés, donc il existe un point "sortie" du polygone Q correspondant à un autre point entrant dans Q quand la frontière du polygone P est traversée dans un sens (voir figure 4.12). Par conséquent, l'intersection fait intervenir un nombre pair de points. Considérons maintenant les cas généraux de deux polygones disjoints. Nous prouvons le lemme par contradiction. Si le nombre d'intersections entre $CH(P)$ et $CH(Q)$ est supérieur à 2 alors les deux polygones initiaux s'intersectent certainement! Comme ce qui est montré dans la figure 4.13.a, $CH(P)$ va diviser $CH(Q)$ en trois parties, mais il est impossible de trouver un tel polygone, dit P_x , qui peut supporter $CH(Q)$ et qui n'intersecte pas P . Donc, $CH(P)$ et $CH(Q)$ peuvent être seulement d'intersection simple.

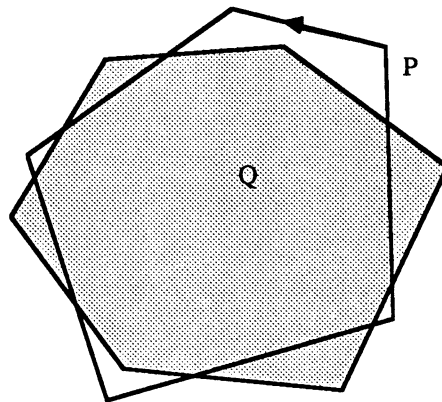
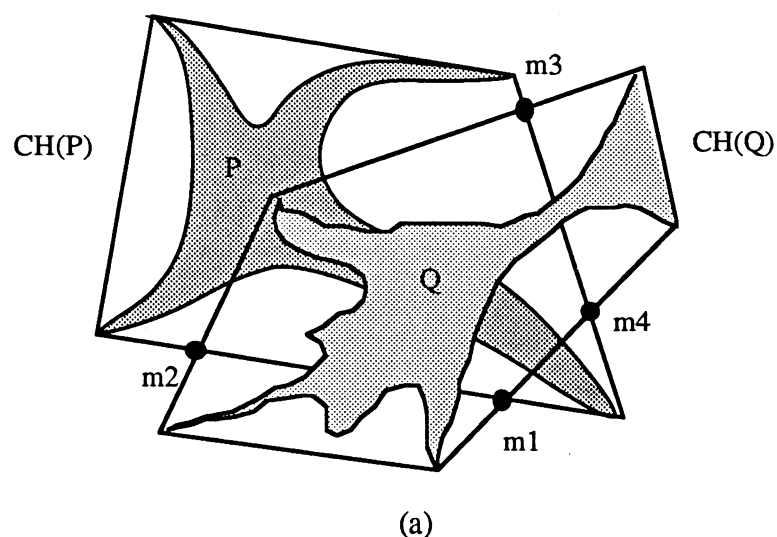


Figure 4.12 L'intersection de deux polygones convexes a lieu par couples



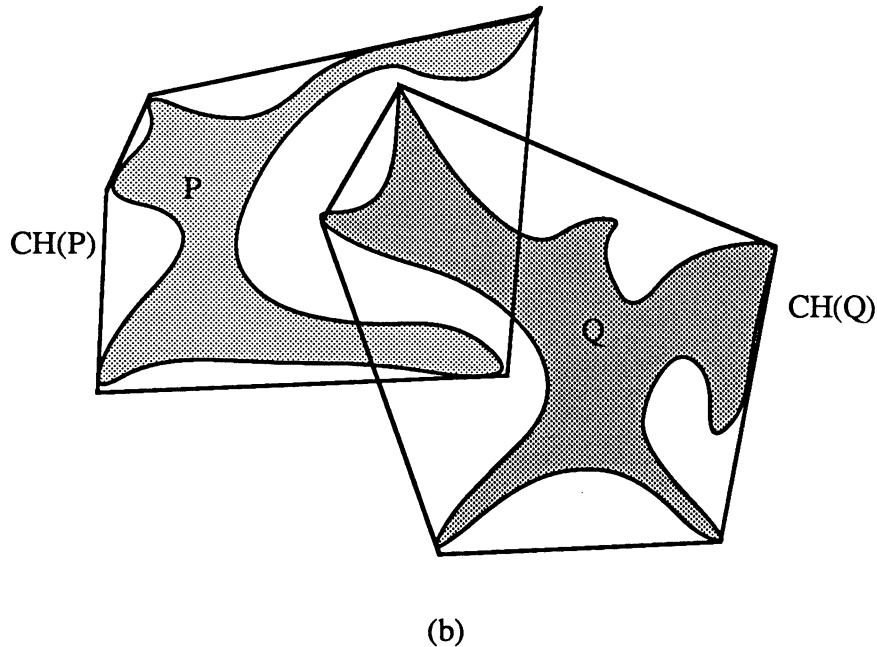


Figure 4.13 Intersection de $CH(P)$ et $CH(Q)$ où P et Q sont deux polygones disjoints:
 (a) intersection impossible; (b) intersection possible

Le lemme 4.8 précise que l'intersection entre $CH(P)$ et $CH(Q)$ où P et Q sont deux polygones disjoints est de type simple. Donc, une fois que $CH(P)$ et $CH(Q)$ ont été trouvés alors $CH(P \cup Q)$ peut être calculé par l'algorithme décrit dans 4.4.2. Par conséquent, on peut obtenir immédiatement à partir de l'algorithme ci-dessus un algorithme pour calculer l'enveloppe convexe de deux polygones disjoints.

Algorithme HULL_TWO_SIMPLE_POLYGONS;

Début

1. trouver $CH(P)$ et $CH(Q)$ par l'algorithme HULL_POLYGON;
2. calculer $CH(P \cup Q)$ par l'algorithme présenté au §4.4.2;

Fin;

Comme les deux étapes de l'algorithme sont de complexité linéaire, on a alors:

Théorème 4.2 L'enveloppe convexe de deux polygones disjoints peut être trouvée en temps linéaire.

4.5 Enveloppe convexe d'un ensemble de polygones disjoints

4.5.1 Prétraitement

Soit $\Omega_i, i=1, 2, \dots, K$ ($K > 2$) un ensemble de polygones disjoints $\Omega_i \cap \Omega_j = \emptyset$. Afin de simplifier les notations, nous allons utiliser $CONV(m)$ pour représenter l'enveloppe convexe du sous-ensemble des m premiers polygones, $CH(i)$ pour représenter l'enveloppe convexe du i -ème polygone.

Lemme 4.9 Si $CONV(i)$ et $CH(i+1)$ ont été obtenus et si $CONV(i)$ et $CH(i+1)$ sont d'intersection simple, alors $CONV(i+1)$ peut être trouvé en temps linéaire.

Prouver. Immédiate d'après le théorème 4.1.

Remarque: Afin de calculer l'enveloppe convexe d'un ensemble de polygones à l'aide du concept des polygones convexes d'intersection simple, les K polygones originaux doivent être ordonnés selon un critère donné ci-après.

Le critère de prétraitement:

- (a). $Y_{\min}(\Omega_i) < Y_{\min}(\Omega_{i+1})$;
- (b). $X_{\min}(\Omega_i) < X_{\min}(\Omega_{i+1})$ si $Y_{\min}(\Omega_i) = Y_{\min}(\Omega_{i+1})$; $i = 1, \dots, K$.

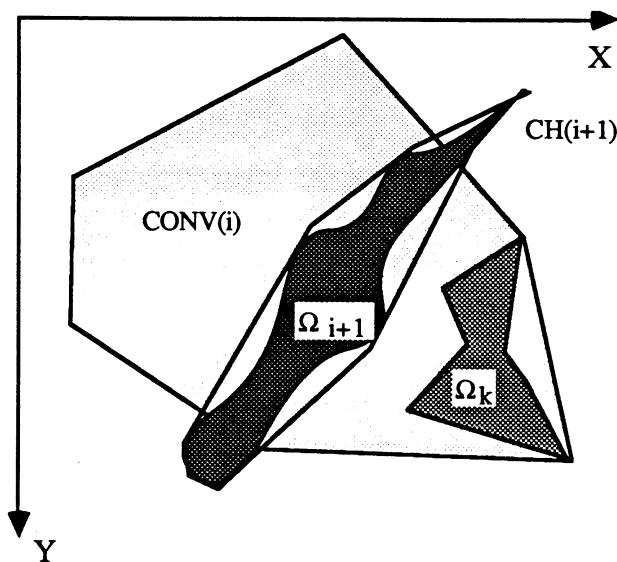


Figure 4.14 Illustration du lemme 4.10

Lemme 4.10 $CONV(i)$ et $CH(i+1)$ sont d'intersection simple si tous les polygones initiaux ont été ordonnés selon le critère de prétraitement.

Prouve. Pour $i = 1$, c'est le cas du lemme 4.8. Pour $i > 1$, nous le prouvons par contradiction. Supposons que $CH(i+1)$ intersecte $CONV(i)$ en plus de 2 points comme ce qui est illustré en figure 4.14. Dans ce cas, il existe alors certainement un polygone dans $CONV(i)$, dit Ω_k avec $k \leq i$ tel que $Y_{\min}(\Omega_k) > Y_{\min}(\Omega_{i+1})$. Autrement Ω_k intersecte Ω_{i+1} . Mais ceci est impossible car le prétraitement impose $Y_{\min}(\Omega_k) > Y_{\min}(\Omega_{i+1})$ avec $k \leq i$.

4.5.2 Algorithme

Comme $CONV(j)$ et $CH(j+1)$ sont d'intersection simple, $CONV(j+1)$ peut alors être calculé par l'algorithme proposé dans §4.4.2. L'algorithme HULL_MULTIPOLYGONES ordonne d'abord les polygones selon le critère de prétraitement (étape 1), et calcule l'enveloppe convexe du premier polygone (étape 2). Ensuite, on calcule itérativement l'enveloppe convexe de la réunion de deux polygones convexes d'intersection simple jusqu'à ce que tous les polygones soient utilisés.

Algorithme HULL_MULTIPOLYGONES;

Entrée : $\Omega_i, i=1, \dots, K$.

Sortie : $CONV(K)$;

Début

1. ordonner tous les polygones selon le critère de prétraitement;

2. calculer $CH(1)$ par HULL_POLYGON;

3. pour $i = 2$ à K faire

début

a. calculer $CH(i)$ par HULL_POLYGON;

b. trouver l'enveloppe convexe de la réunion de $CONV(i-1)$ et $CH(i)$ par l'algorithme proposé dans § 4.4.2 pour obtenir $CONV(i)$;

fin;

Fin;

Le temps d'exécution dans le cas le plus défavorable de cet algorithme est facile à déterminer. L'étape 1 peut s'effectuer d'abord en calculant les y-coordonnées minimales parmi les sommets de chaque polygone, ceci prend un temps en $O(\sum_{i=1}^K n_i)$. Ensuite, on ordonne ces y-coordonnées en ordre ascendant qui peut se faire en $O(K \log_2 K)$. Donc, le temps total pour l'étape 1 est en $O(K \log_2 K + \sum_{i=1}^K n_i)$. Dans § 4.2, nous avons montré que $CH(i)$ peut être calculé en $O(n_i)$ où n_i est le nombre de sommets du i -ième polygone. Donc, l'étape 2 nécessite $O(n_j)$. La complexité dans le pire des cas de l'étape 3b est $O(\sum_{j=1}^i n_j)$. Par conséquent, une

itération de l'étape 3 prend au plus un temps en $O(n_i + \sum_{j=1}^i n_j)$. Le temps total dans le pire des cas de cet algorithme est alors:

$$O(n_1) + O(K \log_2 K + \sum_{i=1}^K n_i) + \sum_{i=2}^K O(n_i + \sum_{j=1}^i n_j)$$

Ceci peut être réécrit comme:

$$\sum_{i=1}^K O(n_i) + \sum_{i=2}^K O(\sum_{j=1}^i n_j)$$

Soit $N = \sum_{i=1}^K n_i$, c'est-à-dire, le nombre total de sommets à traiter. Le temps total de l'algorithme peut alors s'exprimer par

$$O(N) + \sum_{i=2}^K O(\sum_{j=1}^i n_j)$$

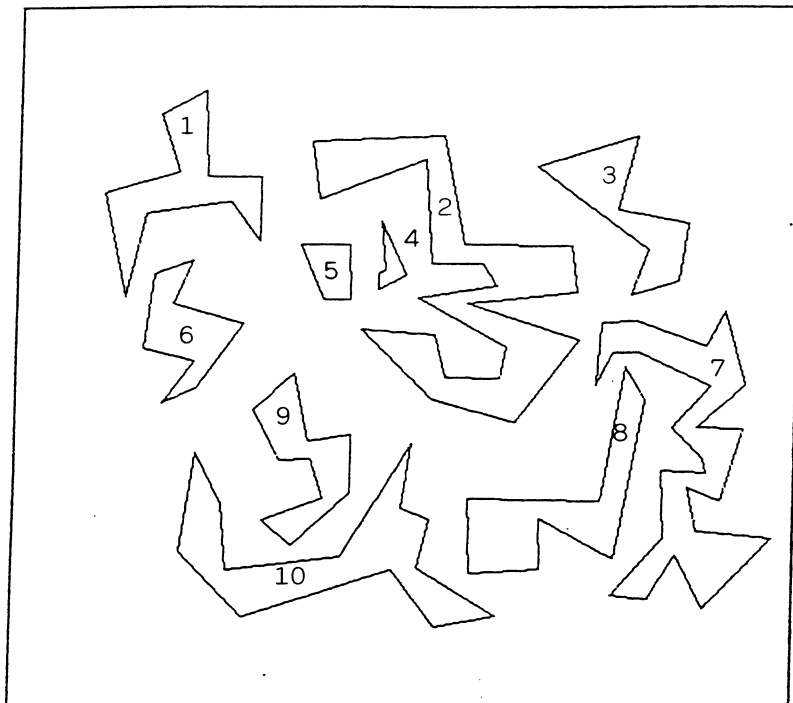
ou encore:

$$O(N(1+(K-1))) = O(NK).$$

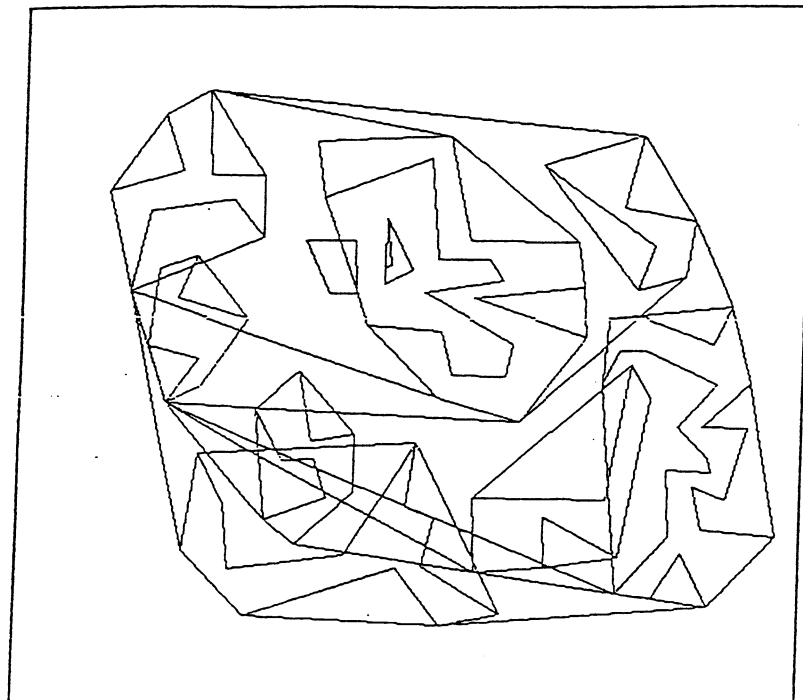
En conclusion, on a:

Lemme 4.11 L'enveloppe convexe d'un ensemble de polygones simples disjoints peut être trouvée en un temps $O(NK)$ où N est le nombre total de sommets et K est le nombre de polygones par l'algorithme HULL_MULTIPOLYGONS.

Un exemple sorti du programme pour l'enveloppe convexe d'un ensemble de polygones est illustré en figure 4.15.



(a)



(b)

Figure 4.15 Exemple de sortie (a) un ensemble de polygones ; (b) son enveloppe convexe basé sur les segments d'appui orientés

Chapitre 5

Construction d'un séparateur

5.1 Introduction

Le calcul d'un séparateur entre deux ensembles de germes est une étape importante de la construction du diagramme de Voronoï. La méthode de "Divide-and-Conquer" pour la construction du diagramme de Voronoï consiste d'abord à diviser l'ensemble d'objets original S en deux sous-ensembles de tailles égales S_1 et S_2 , et à construire récursivement les deux sous-diagrammes de Voronoï $VOD(S_1)$ et $VOD(S_2)$ et ensuite fusionner $VOD(S_1)$ et $VOD(S_2)$ pour obtenir $VOD(S)$. Cette stratégie est largement utilisée pour construire le diagramme de Voronoï d'un ensemble de points [PREPARATA 88], et le diagramme de Voronoï d'un ensemble de segments [DRYSDALE 79] [LEE 81] [KIRKPATRICK 79] [YAP 87]. Les algorithmes diffèrent, selon la façon avec laquelle ils effectuent l'étape de division. Dans le cas d'un ensemble de points, si les deux sous-ensembles de points S_1 et S_2 sont séparables par une droite alors le séparateur sera continu [PREPARATA 88]. Pour un ensemble d'objets généraux, une telle condition de séparation linéaire entre deux sous-ensembles ne peut pas généralement être assurée. Par conséquent, une division d'un ensemble d'objets généraux en deux moitiés peut souvent introduire un séparateur discontinu. Par exemple, la figure 5.1 montre un séparateur coupé en deux portions discontinues. Dans cet exemple, les segments ont été ordonnés selon leurs points d'extrémité hauts. Quand la stratégie "Divide-and-Conquer" est appliquée, la division des segments en deux sous-ensembles est : $S_1 = \{1, 2\}$ et $S_2 = \{3, 4\}$. Par conséquent, le séparateur entre S_1 et S_2 est découpé en deux morceaux discontinus.

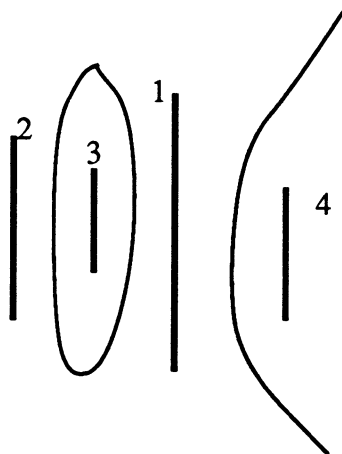


Figure 5.1. Le séparateur entre $S_1 = \{1, 2\}$ et $S_2 = \{3, 4\}$ est composé de deux parties

En général, si nous avons un ensemble d'objets polygonaux, la situation est plus compliquée que celle des segments. Si la méthode "Divide-and-Conquer" est utilisée pour ce cas, deux questions identiques à celles des segments se posent: (1) Comment ordonner les polygones? (2) Est-ce que le séparateur entre deux sous-ensembles de polygones est composé d'une seule composante non discontinue? La réponse à la deuxième question est généralement négative. Considérons un exemple de quatre polygones représentés en figure 5.2. Supposons qu'ils soient ordonnés par le critère du prétraitement proposé dans le paragraphe 4.5.1. Alors, le séparateur $Sep(S_1, S_2)$ où $S_1=\{P_1, P_2\}$ et $S_2=\{P_3, P_4\}$ sera composé de deux parties discontinues.

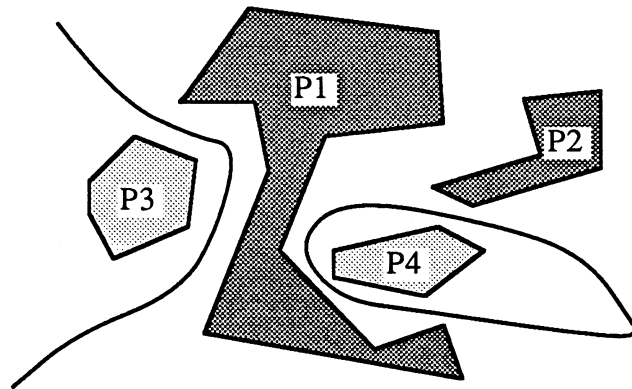


Figure 5.2 Si les quatre polygones sont divisés en deux sous-ensembles, $S_1=\{P_1, P_2\}$ et $S_2=\{P_3, P_4\}$, alors le séparateur $Sep(S_1, S_2)$ est discontinu

Dans le cas plus général, malgré la possibilité d'utiliser plusieurs approches pour ordonner les objets, il est difficile d'éviter la présence d'un séparateur discontinu quand la stratégie "Divide-and-Conquer" est appliquée. Par contre, l'utilisation de l'approche incrémentale associée au prétraitement présenté dans §4.5.1 peut fournir un séparateur continu à chaque étape d'incrémentalation. Ainsi, il va être montré qu'un tel séparateur peut se construire en temps linéaire. Ce chapitre consacre à la construction d'un séparateur, représente la part personnelle la plus importante décrite dans cette thèse.

5.2 Forme d'un séparateur

Dans ce qui suit, nous supposons disposer d'un ensemble de K objets noté $P(K)=\{P_1, P_2, \dots, P_k\}$ vérifiant les contraintes géométriques suivantes:

- (1). l'objet P_i est un polygone composé d'un domaine intérieur connexe et d'une frontière;

(2). $\partial P_i \cap \partial P_j = \emptyset$, ($P_i, P_j \in P(K)$), c'est-à-dire qu'il n'existe pas d'intersection entre les frontières de deux objets;

(3). $P(K)$ a été ordonné selon le critère de prétraitement proposé dans le paragraphe 4.5.1, c'est-à-dire, $Y_{\min}(P_i) < Y_{\min}(P_{i+1})$; et $X_{\min}(P_i) < X_{\min}(P_{i+1})$ si $Y_{\min}(P_i) = Y_{\min}(P_{i+1})$; $i = 1, \dots, K$.

Soit $L = \{P_1, P_2, \dots, P_{j-1}\}$, la réunion des $j-1$ premiers polygones, et $R = P_j$, le j -ème polygone. Alors, d'après le lemme 4.10, l'enveloppe convexe de L et celle de R sont d'intersection simple, c'est-à-dire que les frontières de $CH(L)$ et $CH(R)$ ont au plus deux intersections communes.

Par ailleurs, nous supposons que le diagramme de Voronoï généralisé basé sur éléments de L , $VOD(L)$, et celui de R , $VOD(R)$, sont déjà construits avant de construire le séparateur entre L et R . De plus, un séparateur entre L et R est toujours orienté si bien que pour tout point q de $Sep(L, R)$ il existe un élément e_l de L situé à gauche de $Sep(L, R)$ et un élément e_r de R situé à droite de $Sep(L, R)$ tels qu'ils sont équidistants du point q .

Sous les contraintes géométriques ci-dessus, nous allons discuter du problème de la construction du séparateur entre L et R . Bien que la méthode proposée soit fondée sur le modèle de calcul d'incrémentatation, l'essentiel de la méthode peut s'appliquer au calcul d'un séparateur dans le cas plus général.

Quand $CH(L)$ et $CH(R)$ se coupent d'une façon simple, ils sont dans l'une des situations suivantes:

- (a). $CH(L) \cap CH(R) = \emptyset$, c'est-à-dire que $CH(L)$ et $CH(R)$ sont séparés;
- (b). $\partial CH(L)$ coupe $\partial CH(R)$ en deux points,
- (c). $CH(R)$ est inclus dans $CH(L)$;
- (d). $CH(L)$ est inclus dans $CH(R)$;

Sans perte de généralité, nous ne considérons que les cas (a), (b) et (c) dans ce qui suit parce que le traitement du cas (d) est identique à celui du cas (c).

Lemme 5.1 Le séparateur entre L et R , $Sep(L, R)$, est formé d'une courbe continue si L et R obéissent aux contraintes géométriques (1), (2) et (3) précédemment énoncées.

Preuve. Du fait que le séparateur $Sep(L, R)$ correspond à la frontière des demi-plans généralisés $H(L, R)$ et $H(R, L)$, nous pouvons compléter la démonstration en montrant que $H(L, R)$ et $H(R, L)$ sont des chemins-connectés (un domaine est appelé chemin-connecté si et seulement si

on peut construire un chemin connexe entre deux points quelconques du domaine). Il est facile de montrer que $H(R, L)$ est chemin-connecté. Maintenant, nous montrons que $H(L, R)$ est aussi chemin-connecté. Soit V une demi-droite verticale orientée vers le bas issue du sommet le plus bas de R . Traçons une droite horizontale D telle que $y(D) < Y_{\min}(P_i)$. On choisit $y(D)$ de manière à ce que $d(c, R) = d(D, R)$ et de manière à ce que $d(c, R)$ soit supérieur à $d(c, P_i)$ (pour $P_i \in L$) où $c = D \cap V$. Comme $Y_{\min}(R) > Y_{\min}(P_i)$ ($P_i \in L$) (si $Y_{\min}(R) = Y_{\min}(P_i)$, on peut prendre une droite verticale comme D), D est complètement située dans $H(L, R)$. Ainsi, le demi-plan situé au dessous de D est inclus dans $H(L, R)$. Soit p et q deux points arbitraires dans $H(L, R)$. On projette p sur le polygone le plus proche, P_j , notant l'image correspondante par $I(p, \partial P_j)$. De même, on a $I(q, \partial P_k)$, l'image de q sur le polygone P_k . Selon le lemme 2.7, le chemin entre p et $I(p, \partial P_j)$ et celui de q à $I(q, \partial P_k)$ sont connexes. Soit u le sommet le plus bas de P_j , v celui de P_k , l_u la demi-droite verticale partant de u vers le bas intersectant D au a , l_v celle partant de v vers le bas intersectant D au b . Un chemin entre $I(p, \partial P_j)$ et $I(q, \partial P_k)$ est composé de cinq parties au plus: la partie de $I(p, \partial P_j)$ à u sur ∂P_j , les segments $[u, a]$, $[a, b]$ et $[b, v]$, et la partie de v à $I(q, \partial P_k)$ sur ∂P_k . Comme chacune de ces parties est connexe, le chemin est donc connexe. Ceci montre que le chemin construit entre p et q est connexe (cf. la figure 5.3).

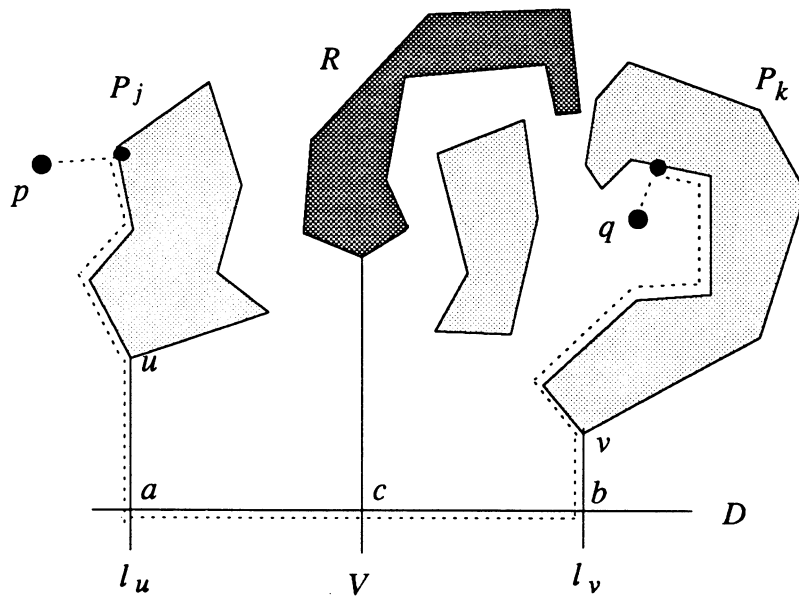


Figure 5.3 Illustration du lemme 5.1

Lemme 5.2. Le séparateur $Sep(L, R)$ est (a) soit fermé si $CH(R)$ est inclus dans $CH(L)$; (b) soit ouvert si $CH(R)$ n'est pas inclus dans $CH(L)$.

Preuve. (a) Rappelons que $Sep(L, R)$ est la frontière des régions de Voronoï $V(L, R)$ et de $V(R, L)$. Si on peut prouver que $V(L, R)$ ou $V(R, L)$ est borné, on a achevé la démonstration.

Nous allons le prouver par contradiction. Supposons que la région de Voronoï $V(R, L)$ est non bornée. Alors, il existe au moins un segment $[a, b]$ dans $\partial CH(L)$ par lequel $Sep(L, R)$ sort de $CH(L)$ et s'étend jusqu'à l'infini. Soit e un point dans ∂R qui est plus proche du segment $[a, b]$ (cf. la figure 5.4.a). Les points a, b, e déterminent un cercle C de centre p . Les deux demi-droites issues de p et passant respectivement par a et b génèrent un secteur angulaire (partie ombrée) qui est montré dans la figure 5.4.b. Selon l'hypothèse, il existe une partie dans l'intérieur du secteur telle qu'elle est dans la région de Voronoï $V(R, L)$. Soit z un point arbitraire dans l'intérieur de ce secteur. Il n'est pas difficile de montrer que le cercle centré en z avec un rayon $r = \min\{d(a, z), d(b, z)\}$ ne contient pas le point e , c'est-à-dire que z est plus proche de a ou plus proche de b que de e . Par conséquent, z est plus proche de ∂L que de ∂R . Comme z est arbitraire dans l'intérieur du secteur considéré, il est donc impossible de trouver un tel point z dans le secteur qui appartienne à la région de Voronoï $V(R, L)$. Ceci est en contradiction avec l'hypothèse.

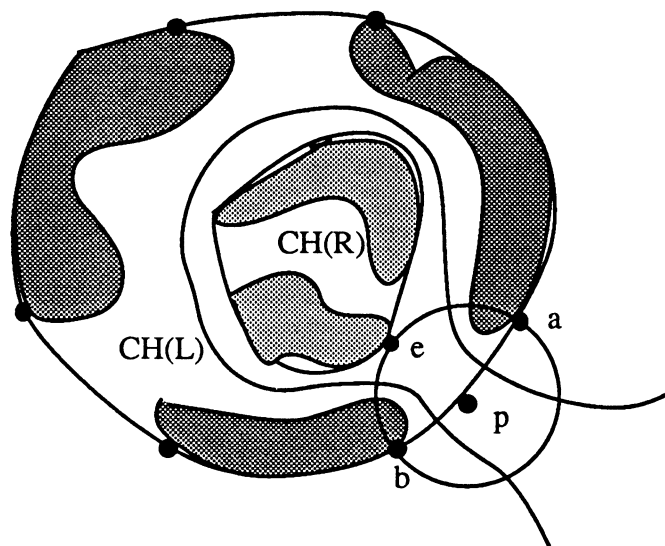


Figure 5.4.a. Le séparateur $Sep(L, R)$ est fermé quand $CH(R)$ est inclus dans $CH(L)$

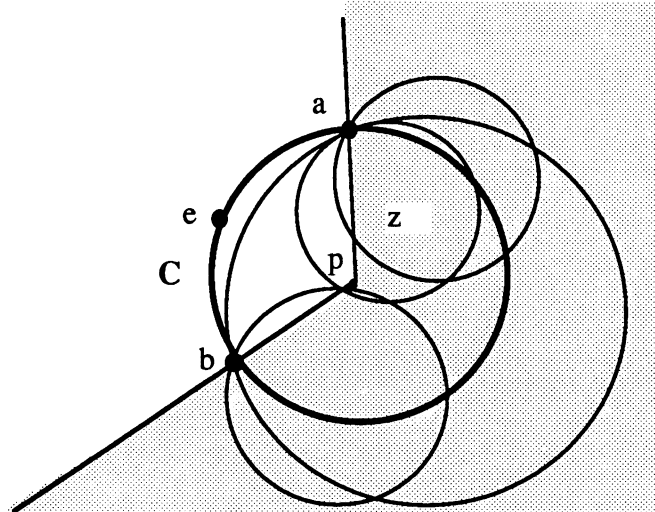


Figure 5.4.b. Aucun cercle centré dans la région ombrée n'inclut e qui est un point de ∂R

(b). Maintenant, nous démontrons que le séparateur entre L et R , $Sep(L, R)$ est ouvert si $CH(R)$ n'est pas inclus entièrement dans $CH(L)$. Pour le démontrer, on doit prouver que la région de Voronoï $V(L, R)$ ou $V(R, L)$ est non bornée. Comme $CH(R)$ n'est pas inclus entièrement dans $CH(L)$, il existe un élément, noté e , tel que e appartient à $\partial R \cap CH(L \cup R)$ mais non à $CH(L)$. Soit e_c une arête de $CH(R)$ qui contient l'élément e . Alors, une ligne passant par e_c divise le plan en deux demi-plans $h1$ et $h2$ (voir figure 5.5). Soit u un point de e . A partir de u , nous prenons une demi-droite s dans $h2$ perpendiculaire à la frontière du demi-plan. Soit v un point dans s . On a alors $d(u, v) = d(e, v) < d(e_l, v)$ pour $e_l \in \partial L$. Par conséquent, $v \in V(R, L)$. Comme v peut s'étendre jusqu'à l'infini, alors $V(R, L)$ est non borné.

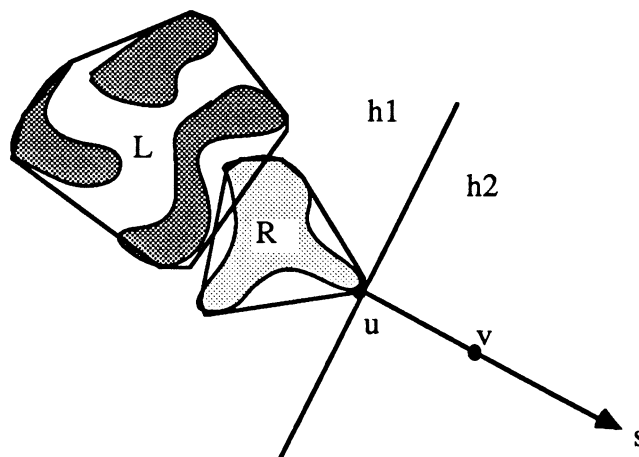


Figure 5.5. Illustration du lemme 5.2 (b)

D'après les lemmes 5.1 et 5.2, nous obtenons immédiatement:

Corollaire 5.1. La région de Voronoï d'un polygone P sur l'ensemble S , $V(P, S-P)$ est non bornée si et seulement si $\partial P \cap CH(S) \neq \emptyset$.

5.3 Calcul de la bissectrice de départ et de la bissectrice d'arrivée

Dans le précédent paragraphe, nous avons démontré que le séparateur $Sep(L, R)$ est formé d'une seule courbe continue (cf. le lemme 5.1). Comme nous l'avons déjà mentionné, c'est le prétraitement et l'utilisation de la stratégie d'incrémentation qui justifient le fait que le séparateur est formé d'une seule courbe non coupée. Rappelons qu'un séparateur est une séquence de bissectrices:

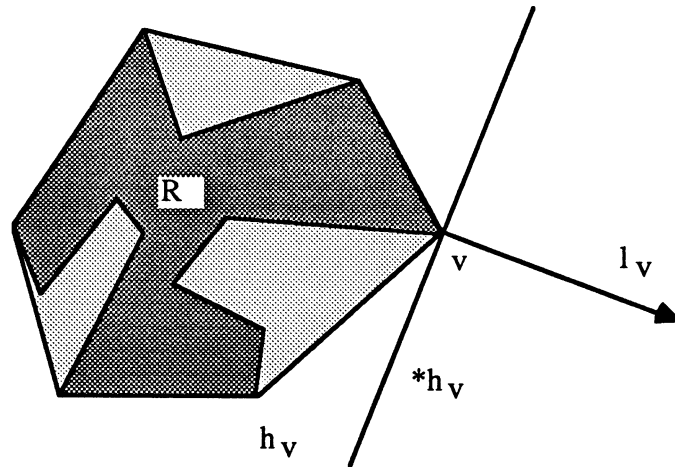
$$Sep(L, R) = \langle b_1, b_2, \dots, b_i, \dots \rangle$$

où chaque b_i est délimité par deux points d'extrémité p_{i-1} et p_i . Les points p_j sont appelés *points de transition* du séparateur. Ainsi, quand une orientation est imposée au séparateur, pour chaque b_i le point de départ et le point d'arrivée seront définis sans ambiguïté.

Jusqu'à présent, nous n'avons qu'une connaissance sur la forme d'un séparateur. Pour tracer un séparateur $Sep(L, R)$, il faut d'abord trouver un point appelé *point d'initialisation* appartenant au séparateur afin qu'à partir de ce point, tout le séparateur puisse être calculé morceau par morceau. Dans ce qui suit, nous allons montrer comment choisir le point d'initialisation pour deux catégories de séparateurs: les séparateurs fermés et les séparateurs ouverts.

5.3.1 Cas des séparateurs fermés

Supposons avoir construit à l'étape j d'incrémentation $VOD(L)$ où $L = \{P_1, P_2, \dots, P_{j-1}\}$, le diagramme de Voronoï pour les $j-1$ premiers polygones, et $VOD(R)$ où $R = \{P_j\}$, le diagramme de Voronoï pour le j -ème polygone. Soit $CONV(j-1)$ l'enveloppe convexe de L , et $CH(j)$ celle de R . Selon le lemme 5.2, nous savons que le séparateur $Sep(L, R)$ est fermé si $CH(j)$ est inclus dans $CONV(j-1)$. L'examen d'inclusion de $CH(j)$ dans $CONV(j-1)$ peut se faire en temps linéaire en utilisant l'algorithme proposé dans le chapitre 4. Dans le cas où le séparateur est fermé, comme ce que nous avons dit, n'importe quel point du séparateur peut être pris comme point de départ. En partant de ce point, nous construisons successivement tous les morceaux des bissectrices du séparateur jusqu'à ce que nous soyons revenus à ce point.

Figure 5.6. Construction d'une demi-droite l_v

Soit v un sommet sur la frontière de l'enveloppe convexe de R où $R = \{P_j\}$. Il est donc situé à droite du séparateur. (En pratique, v peut être l'un des quatre sommets extrêmes: le plus bas, le plus haut, le plus à gauche et le plus à droite, afin d'assurer que v est sur $\partial CH(R)$). Un demi-plan h_v peut être défini de manière à ce qu'il contienne $CH(R)$ et que sa frontière passe par le sommet v . Dans $*h_v$, le complément de h_v , une demi-droite l_v perpendiculaire à la frontière de h_v et partant de v peut être construite (Figure 5.6). Nous allons voir que l_v sera utile pour trouver un point sur le séparateur fermé.

Lemme 5.3. La demi-droite l_v est incluse entièrement dans la région de Voronoï du sommet v $V(v, \partial R - v)$.

Preuve. Soit z un point de l_v . Comme l_v est située dans le demi-plan $*h_v$ et perpendiculaire à la frontière de $*h_v$, on a alors $d(v, z) = \min_{p \in R} d(p, z) = d(R, z)$. Donc, z appartient à la région de Voronoï $V(v, \partial R - v)$. D'après le lemme 2.9, on sait que $V(v, \partial R - v)$ est non borné. De plus, le choix de z en l_v est arbitraire. On en conclut que la demi-droite l_v est incluse entièrement dans la région de Voronoï $V(v, \partial R - v)$.

Lemme 5.4. Soit P un ensemble de N polygones ordonnés. Soit L les $k-1$ ($k < N+1$) premiers polygones de P , R le k -ème polygone de P . La demi-droite l_v coupe $Sep(L, R)$ si $CH(R)$ est inclus dans $CH(L)$.

Preuve. D'après le lemme 5.2, le séparateur $Sep(L, R)$ est fermé. De plus ce séparateur fermé contient ∂R du fait que $CH(R)$ est inclus dans $CH(L)$. Par conséquent, le sommet v — point de départ de la demi-droite l_v — est aussi inclus dans l'intérieur du séparateur fermé $Sep(L, R)$. Il est certain qu'une courbe fermée coupe une demi-droite qui rayonne à partir d'un point situé à l'intérieur de cette courbe fermée. Donc, l_v coupe $Sep(L, R)$.

Le lemme 5.4 montre l'existence d'un point du séparateur situé sur la demi-droite l_v . Pourtant jusqu'à présent, nous ne savons pas comment trouver ce point parce que le séparateur $Sep(L, R)$ est inconnu.

Rappelons que le séparateur est composé d'une séquence de morceaux de bissectrices $B(e_L, e_R)$ où $e_L \in \partial L$ et $e_R \in \partial R$. Soit q le point d'intersection du séparateur $Sep(L, R)$ et de la demi-droite l_v . Il faut déterminer une paire d'éléments (e_L, v) où $e_L \in \partial L$ et $v \in \partial R$ et tel que la bissectrice $B(e_L, v)$ coupe la demi-droite l_v au point q . En outre, il faut signaler que le diagramme de Voronoï généralisé de L , $VOD(L)$, forme une partition du plan excepté l'intérieur de l'ensemble L . Donc, il existe un élément e_L tel que la région de Voronoï associée $V(e_L, \partial L - e_L)$ sur l'ensemble L contient le point q . Nous avons le lemme suivant:

Lemme 5.5. Soit $q = l_v \cap Sep(L, R)$. Soit t le point d'intersection de $B(e_L, v)$ et de l_v . Si $t \in V(e_L, \partial L - e_L)$, alors $q = t$.

Preuve. $t = B(e_L, v) \cap l_v$, implique :

$$\text{et} \quad d(t, e_L) = d(t, v) \quad (5.1)$$

$$d(t, v) = \min_{a \in R} d(t, a) \quad (5.2)$$

La condition $t \in V(e_L, \partial L - e_L)$ implique:

$$d(t, e_L) = \min_{b \in L} d(t, b) \quad (5.3)$$

A partir de (5.1), (5.2) et (5.3), on a:

$$\min_{a \in R} d(t, a) = \min_{b \in L} d(t, b) \quad (5.4)$$

Donc,

$$\begin{aligned} t &\in Sep(L, R) \\ &= \{p \mid \min_{a \in R} d(p, a) = \min_{b \in L} d(p, b)\} \end{aligned} \quad (5.5)$$

Et comme $t \in l_v$, on a immédiatement:

$$t \in l_v \cap Sep(L, R) = q$$

Ce dernier lemme présente une approche pour trouver le point q , dit point de départ du séparateur. Cette approche consiste à ce que pour tout élément $e_L \in \partial L$, on calcule d'abord

l'intersection d'une bissectrice $B(e_L, v)$ et de la demi-droite l_v , et ensuite que l'on vérifie si ce point d'intersection est situé dans la région de Voronoï associée à l'élément e_L , $V(e_L, \partial L - e_L)$. Si nous avons confirmé que le point d'intersection est dans $V(e_L, \partial L - e_L)$, nous prenons alors la bissectrice $B(e_L, v)$ comme première bissectrice, sinon nous réexaminons un autre élément de ∂L et la région de Voronoï correspondante jusqu'à ce que la bissectrice entre ce nouvel élément et le sommet v intersecte l_v dans la région de Voronoï contenant le nouvel élément. Une fois que nous avons trouvé la première bissectrice du séparateur, nous pouvons construire successivement tous les morceaux de bissectrices du séparateur $Sep(L, R)$ à partir de cette première bissectrice jusqu'à ce que soyons revenus à la première bissectrice $B(e_L, v)$. Comme le fait de trouver les éléments e_L et v définis ci-dessus constitue la clé de la construction d'un séparateur fermé, nous les appelons *initialisateurs* du séparateur. Ici, nous pouvons résumer l'algorithme pour trouver deux initialisateurs d'un séparateur fermé.

Algorithme INITIALISATORS_OF_CLOSED_SEPARATOR

entrée: $VOD(L), R$;

Sortie: e_{lo}, e_{ro} ; { deux initialisateurs }

Début

1. trouver un sommet de ∂R situé dans $CH(R)$, noté par v ; puis transformer v en la représentation par élément. On a alors l'initialisateur à droite du séparateur, noté e_{ro} . A partir de v , la demi-droite l_v est construite ;

2. tant que *non trouvé* faire

début

a. prendre un élément e_L tel que $e_L \in \partial L$, trouver t , l'intersection de $B(e_L, e_{ro})$ et de la demi-droite l_v ;

b. si $t \in V(e_L, \partial L - e_L)$ alors

début

1). *trouvé* \leftarrow vrai ;

2). $e_{lo} \leftarrow e_L$;

fin ;

sinon

reprendre un autre élément de ∂L comme e_L ;

fin ;

Fin ;

Comme nous l'avons dit précédemment, en pratique le sommet v dans l'étape 1 peut être choisi parmi l'un des quatre sommets extrêmes de ∂R . Par conséquent, la demi-droite l_v sera soit une demi-droite verticale soit une demi-droite horizontale. De plus, le calcul de l'intersection d'une bissectrice avec une demi-droite verticale ou une demi-droite horizontale est plus facile que celui de l'intersection d'une bissectrice avec une demi-droite arbitraire.

Le temps de calcul de cet algorithme peut être analysé. Soit n_l le nombre de sommets de ∂L et n_r celui de ∂R . L'étape 1 est en fait une initialisation. Un sommet extrême de R peut être trouvé en comparant les coordonnées des sommets de R . Ceci peut se faire en temps proportionnel au nombre de sommets de R . L'étape 2 est une boucle qui calcule le point d'intersection de $B(e_L, e_{ro})$ et l_v et qui détermine si ce point d'intersection est situé dans $V(e_L, \partial L - e_L)$. Dans le pire des cas, elle peut intervenir sur tous les éléments de ∂L . Donc, cette boucle peut s'exécuter n_l fois. Sans perte de généralité, nous supposons $n_l > n_r$. Le temps total pour l'algorithme est alors en $O(n_l)$.

5.3.2 Cas des séparateurs ouverts

Le séparateur $Sep(L, R)$ sera ouvert si $CH(R)$ n'est pas inclus dans l'intérieur de $CH(L)$ (voir lemme 5.2.). L'idée de générer une demi-droite à partir d'un sommet de ∂R situé dans $\partial CH(R)$ est inutile pour trouver un point d'un séparateur ouvert parce que le séparateur ne possède plus de propriété de fermeture. Donc, il faut trouver une autre approche pour calculer un point d'un séparateur ouvert. Par ailleurs, dans le cas où le séparateur est ouvert, si nous avons un point de départ non rejeté à l'infini, nous sommes obligés de tracer le séparateur selon deux directions inverses à partir de ce point, parce que nous ne pouvons pas retourner au point de départ comme dans le cas des séparateurs fermés.

Méthode basée sur un pont

Pour trouver un point dans le séparateur une méthode basée sur le concept de "pont" peut être utilisée. L'idée de cette méthode est de trouver un segment $[b_l, b_r]$, où $b_l \in \partial L$ et $b_r \in \partial R$, appelé *pont* tel que $[b_l, b_r]$ n'intersecte aucun point interne de $L \cup R$. On peut prouver facilement qu'un tel pont intersecte le séparateur en au moins un point. Le problème restant est d'identifier cette intersection sans avoir tracé le séparateur.

Grâce au lemme 2.13 qui montre que deux éléments $e_l \in \partial L$, $e_r \in \partial R$ partagent une arête de Voronoï commune si et seulement si $V(e_l, \partial L - e_l) \cap V(e_r, \partial R - e_r) \cap Sep(L, R) \neq \emptyset$, nous pouvons construire le pont selon le schéma suivant :

- (a) D'abord trouver $e_l \in \partial L$ et $e_r \in \partial R$ tels que $e_l' = e_l \cap V(e_r, \partial R - e_r) \neq \emptyset$ et $e_r' = e_r \cap V(e_l, \partial L - e_l) \neq \emptyset$;
- (b) Puis, en prenant respectivement un point de e_l' comme b_l et un point de e_r' comme b_r créer un pont $[b_l, b_r]$;
- (c) Enfin, trouver $B(e_l, e_r) \cap [b_l, b_r]$ qui va générer un point du séparateur (voir figure 5.7). Si l'une des extrémités d'un segment est dans une région de Voronoï, il est évident que l'intersection de ce segment avec la région de Voronoï est non vide.

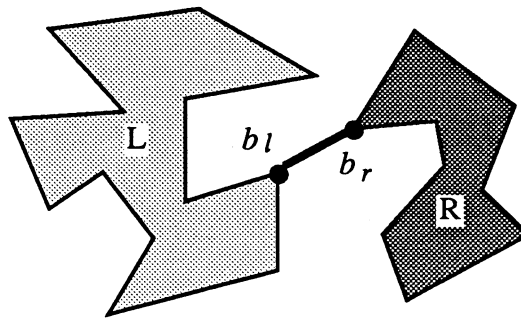


Figure 5.7 Un pont $[b_l, b_r]$ entre deux polygones

Méthode basée sur deux lignes d'appui

Pour éviter de tracer le séparateur en deux fois dans des sens inverses, une autre idée peut être utilisée. Cette idée consiste à trouver deux arêtes de Voronoï d'initialisation d'un séparateur ouvert, dites *bissectrice de départ* et *bissectrice d'arrivée*, qui contiennent respectivement un point d'extrémité à l'infini. Alors, le séparateur ouvert peut se construire à partir de l'un des points "infini" jusqu'à l'autre point "infini". Soit $B_s = B(e_{ls}, e_{rs})$, la bissectrice de départ, et $B_f = B(e_{lf}, e_{rf})$ la bissectrice d'arrivée. Donc, le problème du calcul du séparateur se ramène d'abord à trouver la bissectrice de départ et la bissectrice d'arrivée qui sont définies par deux paires d'éléments. Dans ce qui suit, nous allons montrer que B_s et B_f peuvent être déterminées comme étant uniques.

Rappelons qu'une droite d'appui commune à deux polygones P et Q est une arête de $\partial CH(P \cup Q)$ qui passe par au moins deux points, appelés points d'appui, dont l'un appartient à ∂P et l'autre à ∂Q (cf. figure 5.8).

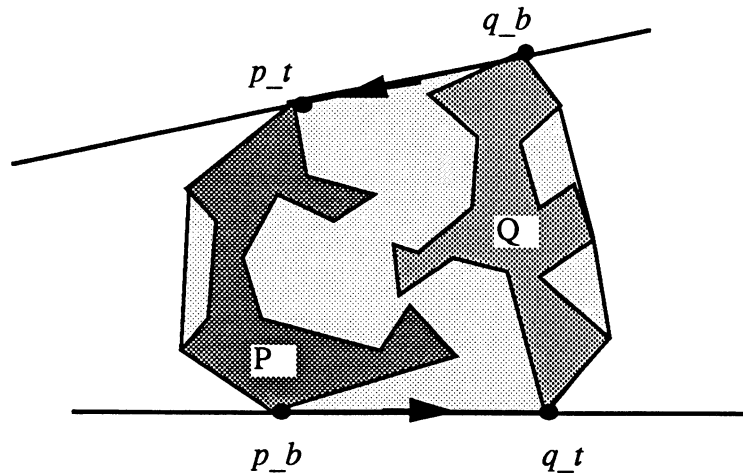


Figure 5.8 Lignes d'appui communes de deux polygones

Il n'y a que deux droites d'appui communes si $CH(P)$ n'est pas inclus dans $CH(Q)$ ou le contraire. Il y a quatre points d'appui correspondant aux deux droites d'appui communes. Soit p_b et p_t les deux points d'appui appartenant à ∂P , et q_t, q_b ceux appartenant à ∂Q . De plus, supposons que p_b, q_t, q_b et p_t sont ordonnés en sens CCW dans $\partial CH(P \cup Q)$ (voir la figure 5.8). Nous supposons que la direction du séparateur $Sep(P, Q)$ est définie de manière à ce que le polygone P soit situé toujours à gauche du séparateur et le polygone Q à droite de celui-ci. Nous allons voir que la bissectrice déterminée par p_b et q_t sera la bissectrice de départ du séparateur, et la bissectrice déterminée par p_t et q_b sera celle d'arrivée.

Lemme 5.6 Soit P et Q deux polygones simples tels que $CH(Q)$ n'est pas inclus entièrement dans $CH(P)$. Alors, $B(p_b, q_t) \cap V(p_b, \partial P - p_b) \cap V(q_t, \partial Q - q_t)$ est la première composante du séparateur orienté $Sep(P, Q)$ et $B(p_t, q_b) \cap V(p_t, \partial P - p_t) \cap V(q_b, \partial Q - q_b)$ est la dernière composante du séparateur $Sep(P, Q)$.

Preuve. Comme la dernière composante du séparateur est symétrique par rapport à la première composante du séparateur, nous ne présentons que le cas de la dernière composante du séparateur. Examinons la figure 5.9. Une droite d'appui commune l_1 passant par p_t et q_b définit un demi-plan h qui ne comprend aucun polygones. On note h^* son complémentaire. Tout d'abord, nous remarquons que $V(p_t, \partial P - p_t) \cap V(q_b, \partial Q - q_b) \neq \emptyset$, sinon, p_t ou q_b n'est plus un point d'appui. On définit par T la demi-droite contenue dans h supportée par la bissectrice de p_t et q_b . Il est évident que T peut être coupé tout au plus en trois morceaux par les arêtes des régions de Voronoï, $V(p_t, \partial P - p_t)$ et $V(q_b, \partial Q - q_b)$. Ces trois morceaux sont:

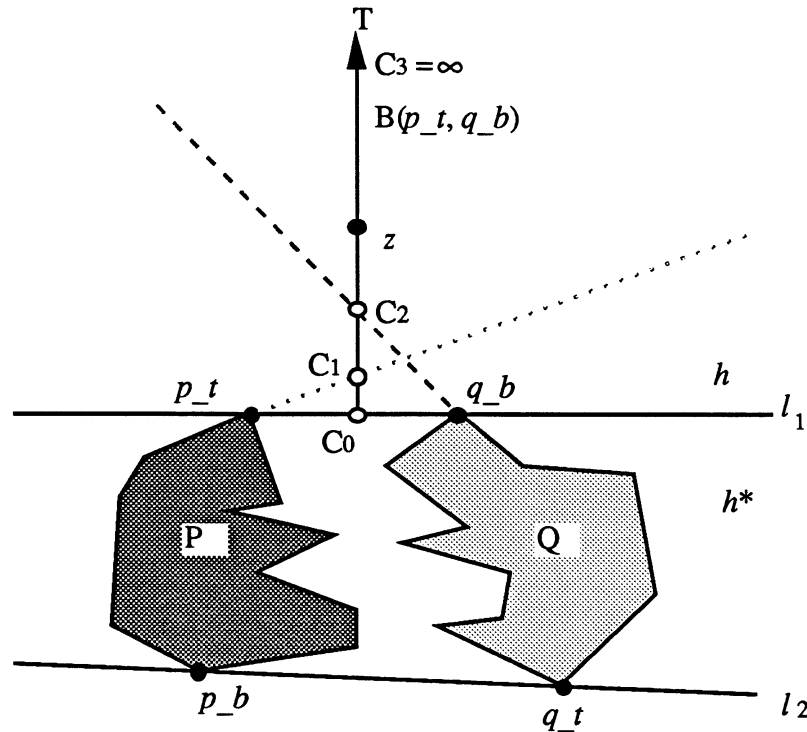


Figure 5.9. Illustration du lemme 5.6

- (1) (c_0, c_1) situé en dehors des régions $V(p_t, \partial P - p_t)$ et $V(q_b, \partial Q - q_b)$;
- (2) (c_1, c_2) situé dans $V(p_t, \partial P - p_t)$ mais hors de $V(q_b, \partial Q - q_b)$ ou le contraire; et
- (3) $(c_2, c_3) = B(p_t, q_b) \cap V(p_t, \partial P - p_t) \cap V(q_b, \partial Q - q_b)$. Pour un point z de (c_2, c_3) , on a alors :

$$d(p_t, z) = \min_{a \in P} d(a, z) = d(P, z) \tag{5.6}$$

et

$$d(q_b, z) = \min_{b \in Q} d(b, z) = d(Q, z) \tag{5.7}$$

Or $B(p_t, q_b) \supseteq (c_2, c_3)$, donc :

$$\min_{a \in P} d(a, z) = \min_{b \in Q} d(b, z) \tag{5.8}$$

alors,

$$\begin{aligned} z \in \{q \mid \min_{a \in P} d(a, q) &= \min_{b \in Q} d(b, q)\} \\ &= \text{Sep}(P, Q) \end{aligned} \tag{5.9}$$

Puisque z peut être rejeté à l'infini, (c_2, c_3) est alors la dernière composante du séparateur $\text{Sep}(P, Q)$.

Corollaire 5.2. Etant donné un ensemble de N polygones ordonnés $P=\{P_1, P_2, P_i, \dots, P_N\}$. Soit $L=\{P_1, P_2, P_i, \dots, P_{K-1}\}$ et $R=\{P_K\}$ où $K \leq N$. Supposons que $CH(R)$ n'est pas entièrement inclus dans $CH(L)$. Alors, $B(l_b, r_t) \cap V(l_b, \partial L - l_b) \cap V(r_t, \partial R - r_t)$ est la première composante du séparateur orienté $Sep(L, R)$ et $B(l_t, r_b) \cap V(l_t, \partial L - l_t) \cap V(r_b, \partial R - r_b)$ est la dernière composante du séparateur $Sep(L, R)$ où l_b, r_t, r_b et l_t sont les points d'appui de $CH(L)$ et $CH(R)$ tels qu'une suite ordonnée $\langle l_b, r_t, r_b, l_t \rangle$ est définie quand $\partial CH(L \cup R)$ est traversée en sens CCW.

Preuve. Facile à partir du lemme 5.6.

D'après le lemme 4.10, $CH(L)$ et $CH(R)$ sont d'intersection simple. En utilisant l'algorithme HULL_TWO_SIMPLE_INTERSECTION_CONVEX présenté dans le paragraphe 4.4.2 et appliqué à $CH(L)$ et $CH(R)$, les points d'appui (ou leurs équivalents, paire d'éléments associés pour déterminer une bissectrice initialisant le séparateur $Sep(L, R)$ et autre paire d'éléments associés pour déterminer une autre bissectrice terminant le séparateur), peuvent être déterminés en temps linéaire avec un prétraitement.

5.4 Tracé d'un séparateur en temps linéaire

Etant donné un ensemble de N polygones ordonnés $P=\{P_1, P_2, \dots, P_i, \dots, P_N\}$. Soit $L=\{P_1, P_2, \dots, P_i, \dots, P_{K-1}\}$ et $R=\{P_K\}$ où $K \leq N$. Après avoir déterminé (a) une paire d'initialisateurs pour un séparateur fermé; ou (b) une bissectrice de départ et une bissectrice d'arrivée pour un séparateur ouvert, nous pouvons construire le séparateur composante par composante.

Rappelons que le séparateur $Sep(L, R)$ peut s'exprimer par :

$$Sep(L, R) = \bigcup_{i=1}^M b_i \quad (5.10)$$

où b_i est une portion (morceau) de bissectrice contribuant au séparateur $Sep(L, R)$, et M est le nombre total de composantes. Remarquons que chaque portion de bissectrice est orientée et qu'un séparateur $Sep(L, R)$ est une courbe continue. Donc, le point d'arrivée de b_{i-1} est confondu avec le point de départ de b_i . On détermine une nouvelle arête de Voronoï, à chaque point de transition $b_i \cap b_{i-1}$, et ainsi de suite.

La figure 5.10 montre un séparateur de deux polygones convexes. Dans ce cas présent, le séparateur est composé de sept portions de bissectrices. Il débute par b_1 et se termine par b_7 .

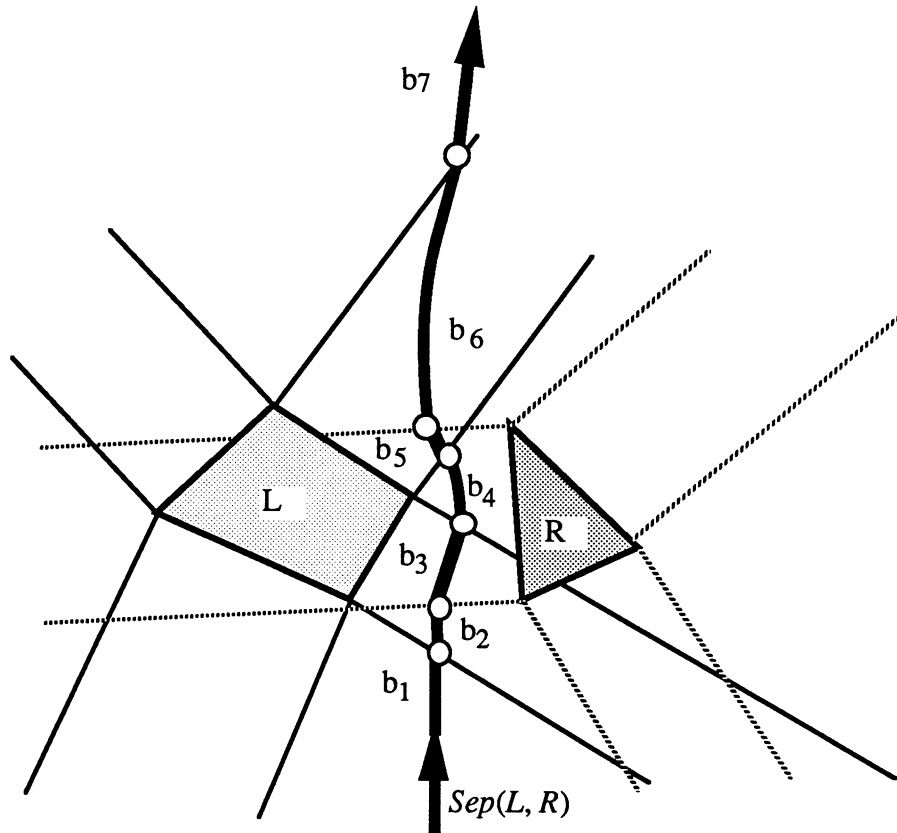


Figure 5.10 Séparateur entre deux polygones convexes

A tout moment de la construction d'un séparateur, nous avons à tracer une portion de bissectrice (arête de Voronoï). Cette portion de bissectrice est en fait contenue dans l'intersection de deux régions de Voronoï (voir le lemme 2.13):

$$b_i = B(e_L^i, e_R^i) \cap V(e_L^i, \partial L - e_L^i) \cap V(e_R^i, \partial R - e_R^i) \quad (5.11)$$

Ceci implique qu'à chaque étape de la construction du séparateur, on ne dispose que d'un triplet: $B(e_L^i, e_R^i)$, $V(e_L^i, \partial L - e_L^i)$ et $V(e_R^i, \partial R - e_R^i)$. Par conséquent, le processus entier de la construction d'un séparateur consiste à déterminer une suite de triplets. Supposons que nous disposons du point de départ de b_i à l'étape i . Il faut trouver le point d'arrivée de b_i . Ce point d'arrivée vérifie une des trois conditions suivantes:

- (a) il est situé sur une arête de $V(e_L^i, \partial L - e_L^i)$;
- (b) il est situé sur une arête de $V(e_R^i, \partial R - e_R^i)$;
- (c) il est à l'intersection d'une arête de $V(e_L^i, \partial L - e_L^i)$ avec une autre arête de $V(e_R^i, \partial R - e_R^i)$.

Il est alors nécessaire de balayer respectivement les arêtes de Voronoï de $V(e_L^i, \partial L - e_L^i)$ et celles de $V(e_R^i, \partial R - e_R^i)$ afin de trouver le point d'arrivée de b_i qui est le point d'intersection le plus

proche du point de départ. Comme Lee-Drysdale [LEE 81] l'indiquaient, une approche irréflechie—explorer toutes les arêtes d'une région de Voronoï chaque fois (retour arrière)—va induire un temps non linéaire de calcul pour construire un séparateur. Ceci est dû aux éléments suivants :

(a). le nombre d'arêtes de $V(e_L^i, \partial L - e_L^i)$ peut être en $O(N_L)$ où N_L est le nombre d'éléments dans ∂L , celui de $V(e_R^i, \partial R - e_R^i)$ peut être en $O(N_R)$ où N_R est le nombre d'éléments dans ∂R ;

(b). le séparateur peut rester dans une région de Voronoï $V(e_L^\beta, \partial L - e_L^\beta)$ en tournant de nombreuses fois avant de sortir de cette région. C'est-à-dire, dans une région de Voronoï $V(e_L^\beta, \partial L - e_L^\beta)$, il y aurait $O(N_R)$ portions de bissectrice: $c_i^\beta = B(e_L^\beta, e_R^i) \cap V(e_L^\beta, \partial L - e_L^\beta) \cap V(e_R^i, \partial R - e_R^i)$ ($i = 1, \dots, N_R$) où N_R est le nombre d'éléments de ∂R . Dans ce cas-là, si nous explorions toutes les arêtes de $\partial V(e_L^\beta, \partial L - e_L^\beta)$ à chaque fois pour trouver un point de sortie du séparateur de $V(e_L^\beta, \partial L - e_L^\beta)$, il faudrait effectuer beaucoup trop d'examen d'arêtes.

(c). Il y a au total $O(N_L + N_R)$ portions de bissectrices composant le séparateur.

Donc, pour chaque portion de bissectrice, ceci va prendre $O(N_L + N_R) = O(N)$ en temps pour trouver le point d'arrivée de la bissectrice correspondante si une approche simple de balayage est adoptée. Par conséquent, ceci peut conduire à un temps $O(N^2)$ pour construire le séparateur entier.

Cependant, le schéma suivant de balayage peut éviter une opération de retour arrière.

Schéma de balayage CW-CCW de la frontière d'une région de Voronoï [LEE 81]:

En partant d'un point où $B(e_L^\beta, e_R^\beta)$ entre dans une région de Voronoï, explorer les arêtes d'une région courante de $VOD(L)$ dans le sens inverse des aiguilles d'une montre (CCW) et balayer les arêtes d'une région courante de $VOD(R)$ dans le sens des aiguilles d'une montre (CW).

Lemme 5.7. Si une région de Voronoï, $V(e_L^\beta, \partial L - e_L^\beta)$, contient k portions de bissectrices entre e_L^β ($\in \partial L$) et e_R^i ($\in \partial R$) ($i = 1, \dots, k \geq 1$), alors, b_{i+1}^β ($i = 1, \dots, k-1$), portion du séparateur dans la région de Voronoï $V(e_R^i, \partial R - e_R^i)$, satisfait: $B(e_L^\beta, e_R^i) \cup h(e_L^\beta, e_R^i) \supset b_{i+1}^\beta$. On a une propriété similaire si $V(e_R^\beta, \partial R - e_R^\beta)$ contient plusieurs portions de bissectrices du séparateur.

Preuve. Selon le lemme 2.13, la $(i+1)$ -ème portion du séparateur dans la région de Voronoï $V(e_L^\beta, \partial L - e_L^\beta)$ peut s'exprimer par:

$$b_{i+1}^\beta = B(e_L^\beta, e_R^{i+1}) \cap V(e_L^\beta, \partial L - e_L^\beta) \cap V(e_R^{i+1}, \partial R - e_R^{i+1}) \quad (5.12)$$

Soit q un point de b_{i+1}^β ,

on a alors:

$$q \in B(e_L^\beta, e_R^{i+1}) \quad (5.13)$$

$$q \in V(e_L^\beta, \partial L - e_L^\beta) \quad (5.14)$$

et

$$q \in V(e_R^{i+1}, \partial R - e_R^{i+1}) \quad (5.15)$$

(5.13) implique :

$$d(e_L^\beta, q) = d(e_R^{i+1}, q) \quad (5.16)$$

(5.15) implique :

$$d(e_R^{i+1}, q) \leq d(e_R^i, q) \quad (5.17)$$

D'après (5.16) et (5.17), on a:

$$d(e_L^\beta, q) \leq d(e_R^i, q) \quad (5.18)$$

où l'égalité est due au fait que z peut être situé sur la frontière de $V(e_R^{i+1}, \partial R - e_R^{i+1})$.

D'où

$$q \in h(e_L^\beta, e_R^i) \cup B(e_L^\beta, e_R^i) \quad (5.19)$$

Le choix de q étant arbitraire, on a alors

$$B(e_L^\beta, e_R^i) \cup h(e_L^\beta, e_R^i) \supset b_{i+1}^\beta \quad (5.20)$$

Quand $V(e_R^\alpha, \partial R - e_R^\alpha)$ contient plusieurs morceaux de bissectrices, la preuve pour montrer l'inclusion $B(e_L^j, e_R^\alpha) \cup h(e_L^j, e_R^\alpha) \supset b_{j+1}^\alpha$ est similaire.

Le lemme ci-dessus présente le fait que b_{i+1}^β , morceau de bissectrice suivant de b_i^β , va s'étendre à gauche de ou sur la bissectrice courante $B(e_L^\beta, e_R^i)$ si la région de Voronoï $V(e_L^\beta, \partial L - e_L^\beta)$ contient plusieurs morceaux de bissectrices b_i^β ($i=1, 2, \dots, k$).

Remarque : le demi-plan utilisé ici est celui défini au sens généralisé, c'est-à-dire que sa frontière n'est pas nécessairement définie par une droite, mais par une bissectrice entre deux éléments qui sont soit deux points, soit un point et un segment, soit deux segments.

Corollaire 5.3. Si une région de Voronoï de $VOD(L)$, $V(e_L^\beta, \partial L - e_L^\beta)$, contient k (>1) morceaux de bissectrice du séparateur $Sep(L, R)$, alors on a $b_i^\beta = B(e_L^\beta, e_R^i) \cap V(e_L^\beta, \partial L - e_L^\beta) \cap V(e_R^i, \partial R - e_R^i)$ avec $i=1, 2, \dots, k$. Si t_i désigne l'intersection de $B(e_L^\beta, e_R^i)$ et de $\partial V(e_L^\beta, \partial L - e_L^\beta)$, alors $B(e_L^\beta, e_R^i)$ sort de $V(e_L^\beta, \partial L - e_L^\beta)$ aux points t_i . $t_1, t_2, \dots, t_i, \dots, t_k$ sont alors dans l'ordre CCW le long de $\partial V(e_L^\beta, \partial L - e_L^\beta)$. Une propriété similaire est aussi vérifiée si une région de $VOD(R)$ contient plusieurs morceaux de bissectrices.

Preuve. Soit z_1, z_2, \dots, z_k les points de transition du séparateur $Sep(L, R)$ dans $V(e_L^\beta, \partial L - e_L^\beta)$. Nous allons montrer que la portion de $B(e_L^\beta, e_R^{i+1})$ commençant en z_i est située à gauche de $B(e_L^\beta, e_R^i)$. Il est évident que cette portion de $B(e_L^\beta, e_R^{i+1})$ s'étend entièrement dans un demi-plan déterminé par une arête de $\partial V(e_R^i, \partial R - e_R^i)$ sur laquelle z_i est situé. Nous notons une telle portion de $B(e_L^\beta, e_R^{i+1})$ par $B_{z_i}(e_L^\beta, e_R^{i+1})$. Alors, nous avons:

$$B_{z_i}(e_L^\beta, e_R^{i+1}) = B(e_L^\beta, e_R^{i+1}) \cap h^*(e_R^{i+1}, e_R^i) \quad (5.21)$$

où h^* représente un demi-plan (frontière comprise). En supposant qu'un point q appartient à $B_{z_i}(e_L^\beta, e_R^{i+1})$, nous avons:

$$d(q, e_L^\beta) = d(q, e_R^{i+1}) \quad (5.22)$$

et

$$q \in h^*(e_R^{i+1}, e_R^i) \quad (5.23)$$

c'est-à-dire :

$$d(q, e_R^{i+1}) \leq d(q, e_R^i) \quad (5.24)$$

Alors

$$d(q, e_L^\beta) \leq d(q, e_R^i) \quad (5.25)$$

Ceci démontre que le point q est dans $h^*(e_L^\beta, e_R^i)$. Par conséquent, $B_{z_i}(e_L^\beta, e_R^{i+1})$ est soit à gauche de $B(e_L^\beta, e_R^i)$ soit dans $B(e_L^\beta, e_R^i)$.

Examinons la figure 5.11. Si nous partons de t_i ($i=1, 2, \dots, k-1$) dans le sens CCW suivant $\partial V(e_L^\beta, \partial L - e_L^\beta)$, t_{i+1} sera alors rencontré en premier. Par conséquent, t_1, t_2, \dots, t_k sont ordonnés en sens CCW suivant $\partial V(e_L^\beta, \partial L - e_L^\beta)$.

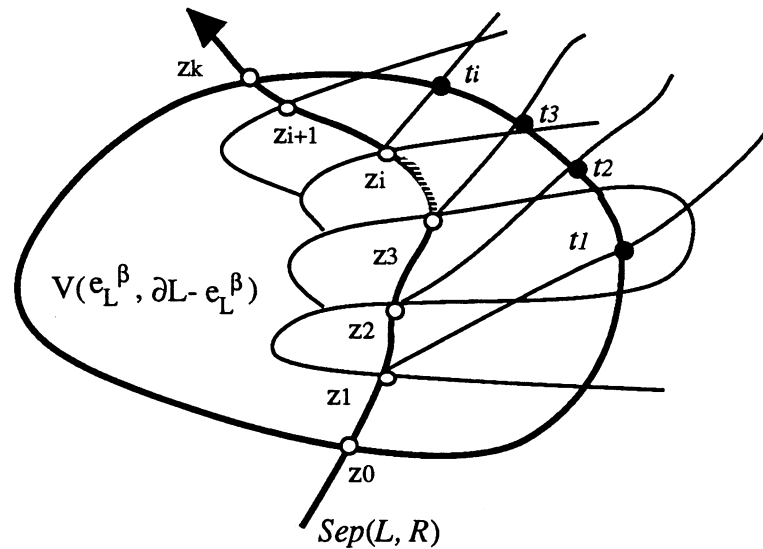


Figure 5.11. Quand un séparateur tourne plusieurs fois dans une même région de Voronoï, les intersections de la frontière de la région avec les bissectrices correspondant à chaque composante du séparateur sont ordonnées suivant la frontière de la région

Corollaire 5.4. Même si $V(e_L^\beta, \partial L - e_L^\beta)$ contient k ($k > 1$) morceaux de bissectrices, $b_1^\beta, b_2^\beta, \dots, b_i^\beta, \dots, b_k^\beta$, le point de sortie du séparateur de $V(e_L^\beta, \partial L - e_L^\beta)$ peut être trouvé sans l'opération de retour arrière en utilisant un schéma de balayage CCW. On a une propriété similaire aussi pour une région de Voronoï de $VOD(R)$ mais il faut explorer la frontière de la région en sens CW.

Preuve. A chaque étape de la construction d'un séparateur, on doit considérer un triplet: $B_{z(i-1)}(e_L^\beta, e_R^i)$, $V(e_L^\beta, \partial L - e_L^\beta)$ et $V(e_R^i, \partial R - e_R^i)$ (voir la figure 5.10). Quand nous avons vérifié que $B_{z(i-1)}(e_L^\beta, e_R^i)$ ($i=1, 2, \dots, k-1$) va sortir de $V(e_R^i, \partial R - e_R^i)$ avant qu'il sorte de $V(e_L^\beta, \partial L - e_L^\beta)$, nous étiquetons une arête courante de $V(e_L^\beta, \partial L - e_L^\beta)$ que $B_{z(i-1)}(e_L^\beta, e_R^i)$ coupe. Ensuite, nous n'explorons $\partial V(e_L^\beta, \partial L - e_L^\beta)$ en sens CCW qu'à partir de cette arête courante si bien qu'une intersection nouvelle de $B_{z_i}(e_L^\beta, e_R^{i+1})$ et de $\partial V(e_L^\beta, \partial L - e_L^\beta)$ peut être trouvée. Nous continuons un tel processus jusqu'à ce que le point de sortie du séparateur de la région de Voronoï $V(e_L^\beta, \partial L - e_L^\beta)$ soit trouvé. Ceci assure qu'il n'y a pas d'opération de retour arrière.

D'après le lemme 2.8, pour un ensemble de N éléments, le nombre d'arêtes de Voronoï et le nombre de sommets de Voronoï sont en $O(N)$. En combinant ce fait avec le schéma de balayage CW-CCW, on a:

Théorème 5.1. Le séparateur $Sep(L, R)$ peut être construit en $O(N)$ où N est le nombre d'éléments de $\partial(L \cup R)$.

Preuve. D'après le corollaire 5.4, il n'y a pas d'opération de retour arrière pour trouver les points de transition. Donc, pendant la construction du séparateur, le nombre d'examen d'arêtes est au plus égale au nombre total des arêtes de Voronoï dans $VOD(L \cup R)$ qui est $O(N)$.

5.5 Fusion de deux sous-diagrammes de Voronoï

Une fois que nous avons construit le séparateur entre L situé à gauche, et R situé à droite, nous pouvons obtenir le diagramme final de l'ensemble $L \cup R$ selon le lemme qui suit.

Lemme 5.8. $VOD(L \cup R)$ consiste en la réunion de la portion de $VOD(L)$ qui est située à gauche du séparateur $Sep(L, R)$ et de la portion de $VOD(R)$ qui est située à droite du séparateur.

Preuve. Selon le lemme 2.3, nous pouvons représenter une région de Voronoï par combinaison de demi-plans:

$$V(e_i, S-e_i) = \bigcap_{e_j \in S-e_i} h(e_i, e_j) \quad (5.26)$$

Tout point à gauche de $Sep(L, R)$ peut s'exprimer par :

$$H(L, R) = \bigcup_{e_L \in \partial L, e_R \in \partial R} \bigcap h(e_L, e_R) \quad (5.27)$$

De même, tout point à droite de $Sep(L, R)$ peut s'écrire :

$$H(R, L) = \bigcup_{e_R \in \partial R, e_L \in \partial L} \bigcap h(e_R, e_L) \quad (5.28)$$

De façon similaire:

$$VOD(L \cup R) = \bigcup_{e_i \in \partial(L \cup R)} V(e_i, \partial L \cup \partial R - e_i)$$

$$= [\cup_{e_i \in \partial L} V(e_i, \partial L \cup \partial R - e_i)] \cup [\cup_{e_i \in \partial R} V(e_i, \partial L \cup \partial R - e_i)] \quad (5.29)$$

Soit

$$DIAG_L = \cup_{e_i \in \partial L} V(e_i, \partial L \cup \partial R - e_i) \quad (5.30)$$

$$DIAG_R = \cup_{e_i \in \partial R} V(e_i, \partial L \cup \partial R - e_i) \quad (5.31)$$

En utilisant (5.26) et (5.27), on a :

$$\begin{aligned} DIAG_L &= \cup_{e_i \in \partial L} \cap_{e_j \in \partial(L \cup R) - e_i} h(e_i, e_j) \\ &= \cup_{e_i \in \partial L} [\{ \cap_{e_j \in \partial L - e_i} h(e_i, e_j) \} \cap \{ \cap_{e_j \in \partial R} h(e_i, e_j) \}] \\ &= [\cup_{e_i \in \partial L} \{ \cap_{e_j \in \partial L - e_i} h(e_i, e_j) \}] \cap [\cup_{e_i \in \partial L} \{ \cap_{e_j \in \partial R} h(e_i, e_j) \}] \\ &= [\cup_{e_i \in \partial L} V(e_i, \partial L - e_i)] \cap [\cup_{e_i \in \partial L} \cap_{e_j \in \partial R} h(e_i, e_j)] \\ &= VOD(L) \cap H(L, R) \end{aligned} \quad (5.32)$$

De même, à l'aide de (5.26) et (5.28):

$$\begin{aligned} DIAG_R &= \cup_{e_i \in \partial R} \cap_{e_j \in \partial(L \cup R) - e_i} h(e_i, e_j) \\ &= \cup_{e_i \in \partial R} [\{ \cap_{e_j \in \partial R - e_i} h(e_i, e_j) \} \cap \{ \cap_{e_j \in \partial L} h(e_i, e_j) \}] \\ &= [\cup_{e_i \in \partial R} \{ \cap_{e_j \in \partial R - e_i} h(e_i, e_j) \}] \cap [\cup_{e_i \in \partial R} \{ \cap_{e_j \in \partial L} h(e_i, e_j) \}] \\ &= [\cup_{e_i \in \partial R} V(e_i, \partial R - e_i)] \cap [\cup_{e_i \in \partial R} \cap_{e_j \in \partial L} h(e_i, e_j)] \\ &= VOD(R) \cap H(R, L) \end{aligned} \quad (5.33)$$

Ceci achève la démonstration du lemme.

Le lemme ci-dessus nous permet d'obtenir le diagramme final $VOD(L \cup R)$ à partir de deux sous-diagrammes $VOD(L)$ et $VOD(R)$ en ôtant la portion de $VOD(L)$ qui est située à droite de $Sep(L, R)$ et la portion de $VOD(R)$ qui s'étend à gauche de $Sep(L, R)$. Un tel processus de fusion peut être effectué sans nécessiter d'opération supplémentaire. En effet, le séparateur $Sep(L, R)$ traverse toujours quelques régions de Voronoï de $VOD(L)$ ainsi que quelques régions de $VOD(R)$. Si, pendant la fusion, le séparateur entre dans une région de $VOD(L)$, $V(e_L, \partial L - e_L)$, par une arête de $\partial V(e_L, \partial L - e_L)$ appelée *arête d'entrée*, et sort de cette région par une autre arête de $\partial V(e_L, \partial L - e_L)$ appelée *arête de sortie* (voir figure 5.12), il faut alors, pour modifier $V(e_L, \partial L - e_L)$ en $V(e_L, \partial L \cup \partial R - e_L)$, remplacer la chaîne d'arêtes $\langle C_1, \dots, C_2 \rangle$ par la portion du séparateur qui est située dans $V(e_L, \partial L - e_L)$ selon sa direction propre. Un tel remplacement n'est qu'une opération de pointeur sur des arêtes. Ceci est une opération très simple!

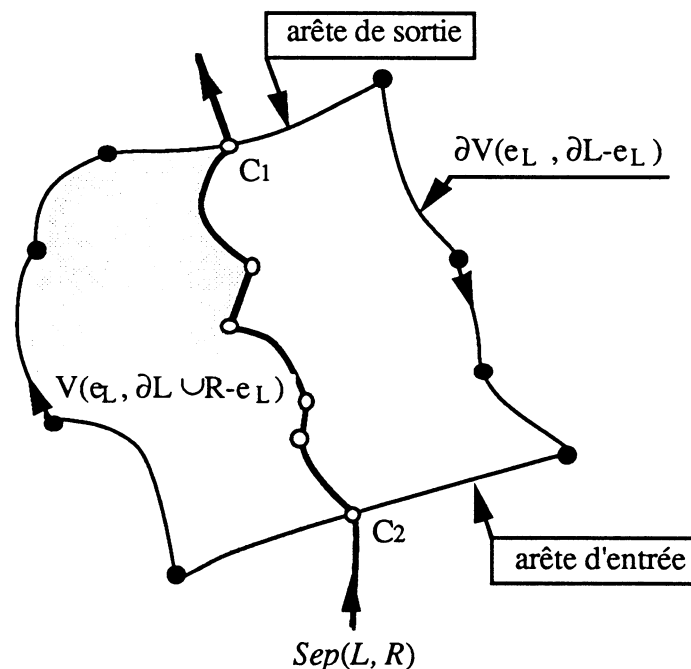


Figure 5.12 Renouvellement d'une région de Voronoï

5.6 Illustration

La figure 5.13 présente un exemple sorti du programme. Cet exemple illustre le processus de la construction du diagramme de Voronoï généralisé de trois polygones P_1 , P_2 et P_3 (voir plus loin l'algorithme EXVODMP au chapitre 6). Dans une telle construction, le séparateur entre P_1 et P_2 est fermé (figure 5.13(d)); et le séparateur entre $P_1 \cup P_2$ et P_3 est ouvert (figure 5.13(g)).

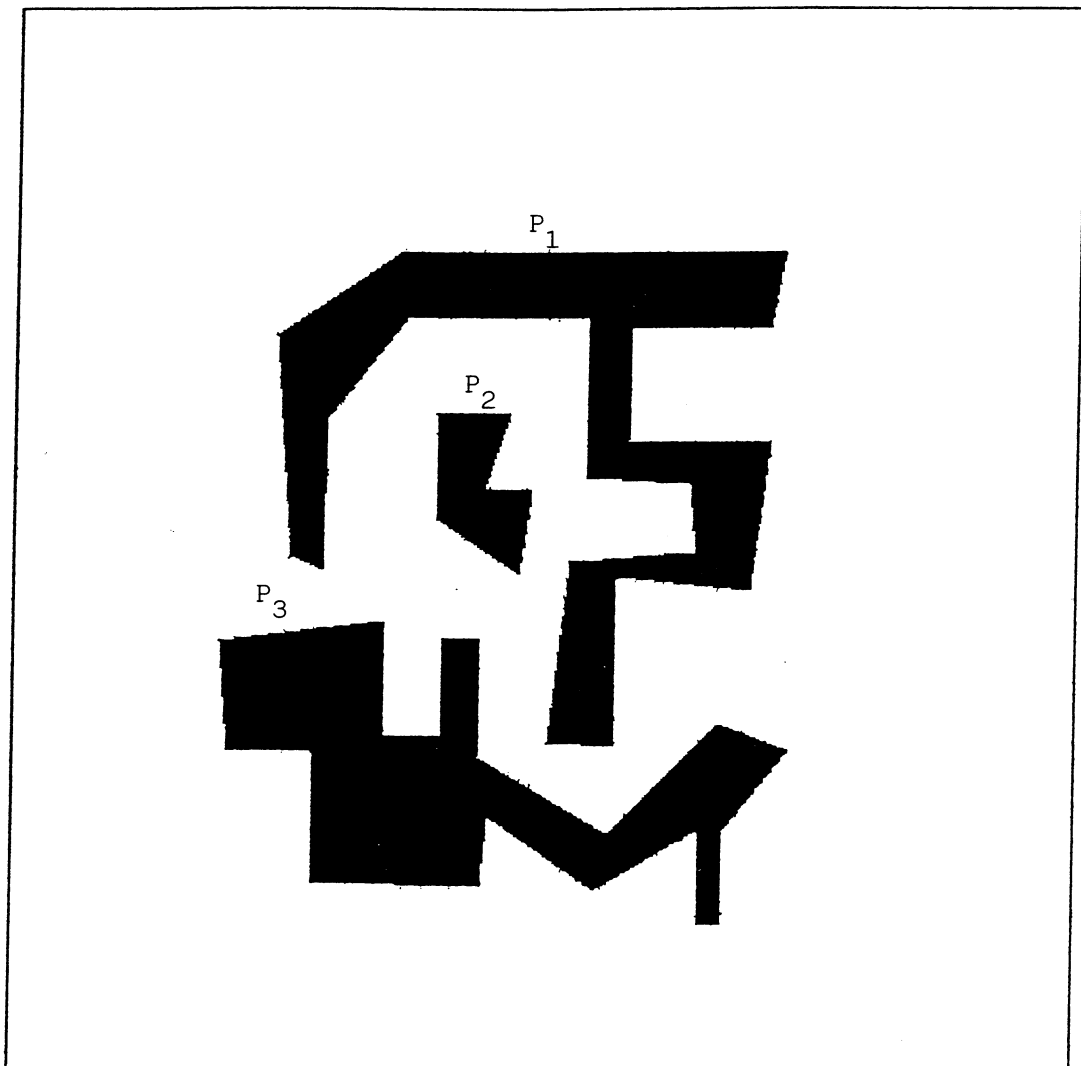


Figure 5.13 (a) Trois polygones initiaux P_1 , P_2 et P_3

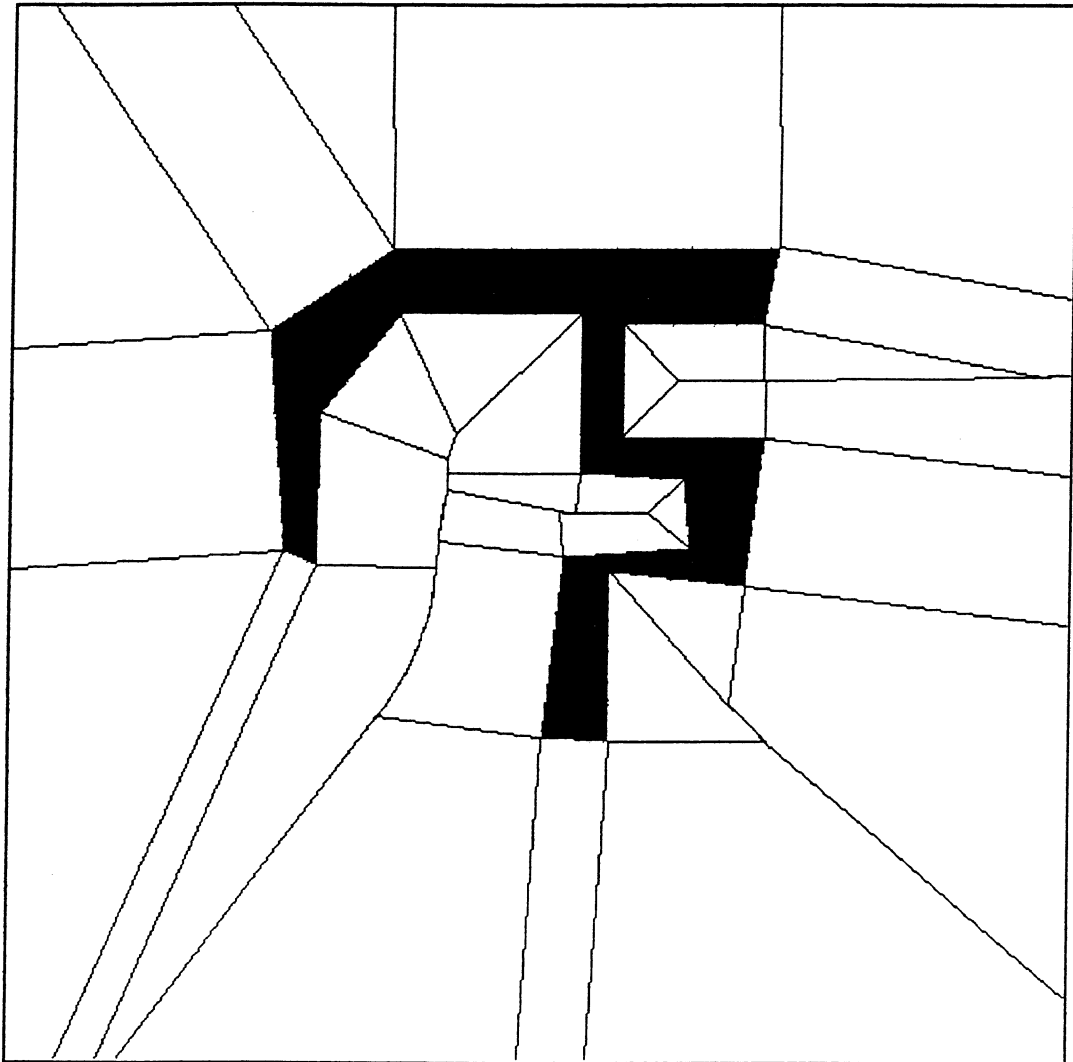


Figure 5.13 (b) $VOD_e(P_I)$ — DVG basé sur les éléments pour l'extérieur du polygone P_I

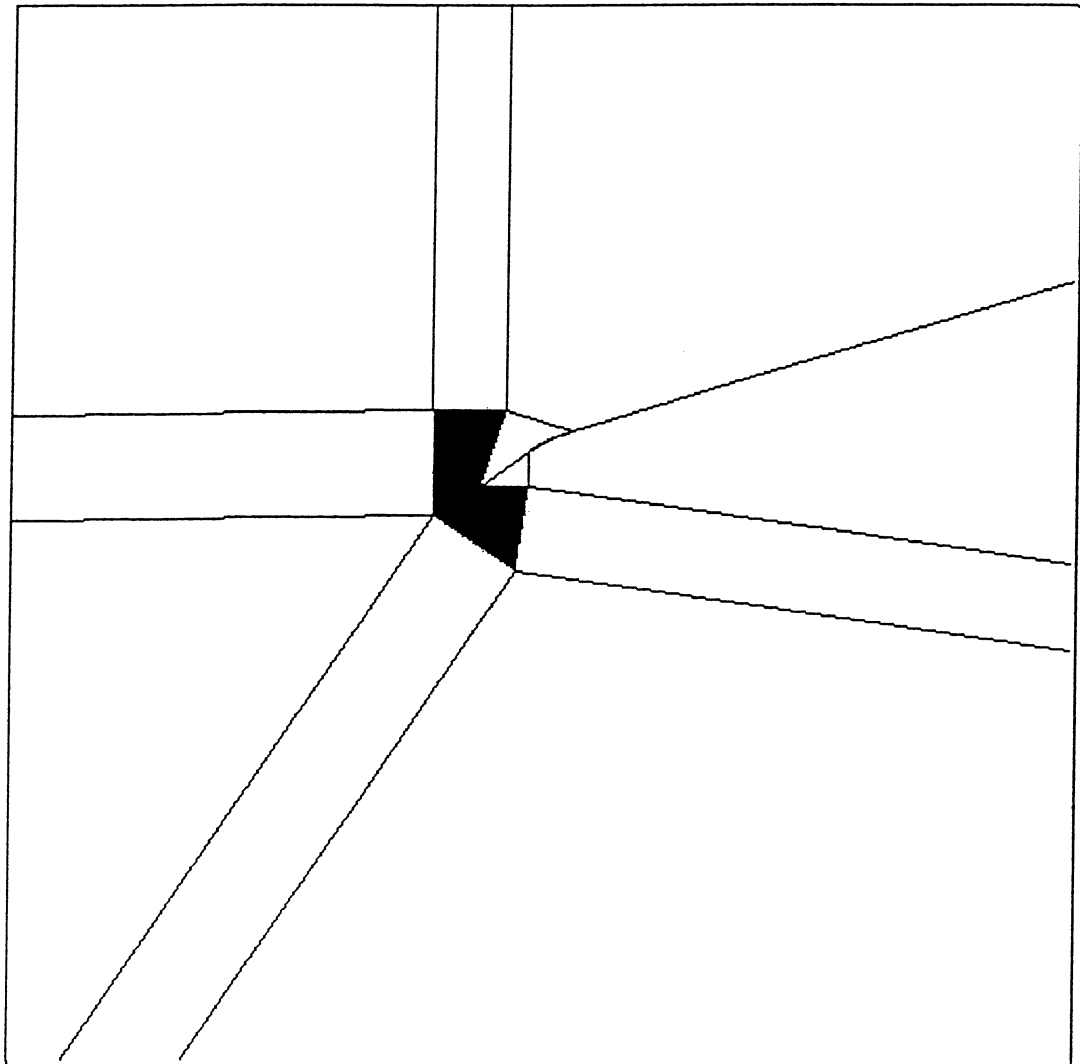


Figure 5.13 (c) $VOD_e(P_2)$ — DVG basé sur les éléments pour l'extérieur du polygone P_2

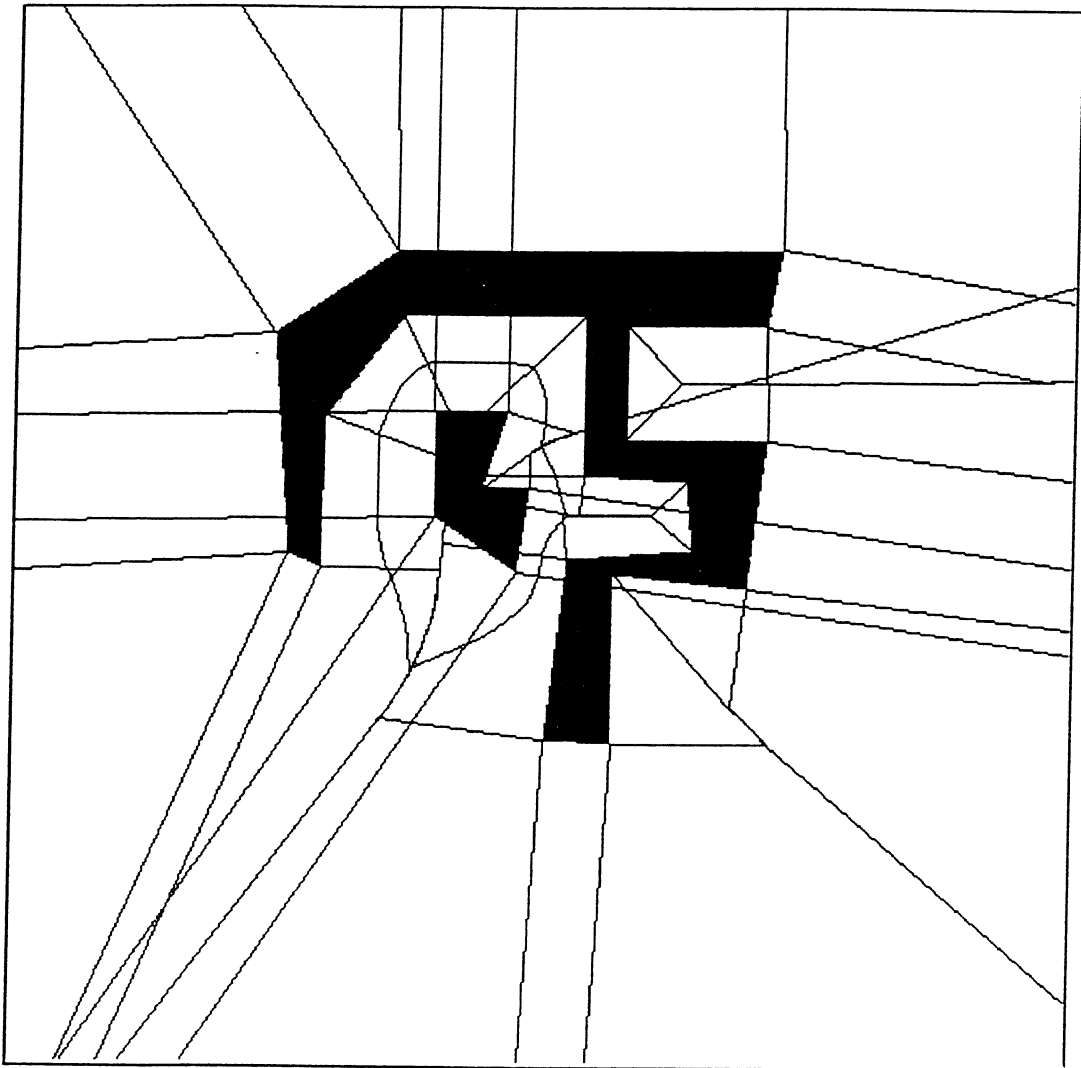


Figure 5.13 (d) Superposition de $VOD_e(P_1)$ et $VOD_e(P_2)$, et représentation du séparateur $Sep(P_1, P_2)$ (fermé)

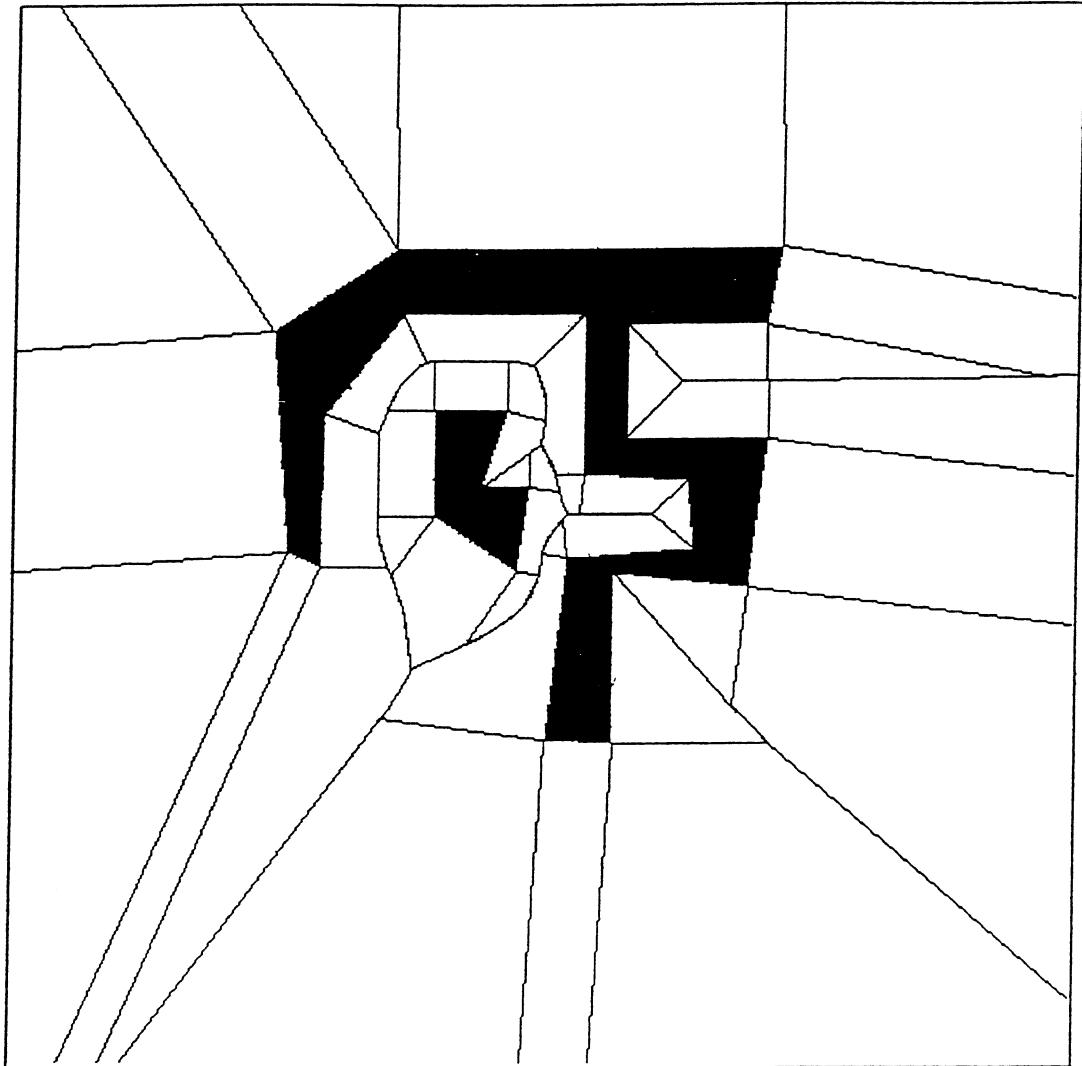


Figure 5.13 (e) $VOD_e(P_1 \cup P_2)$

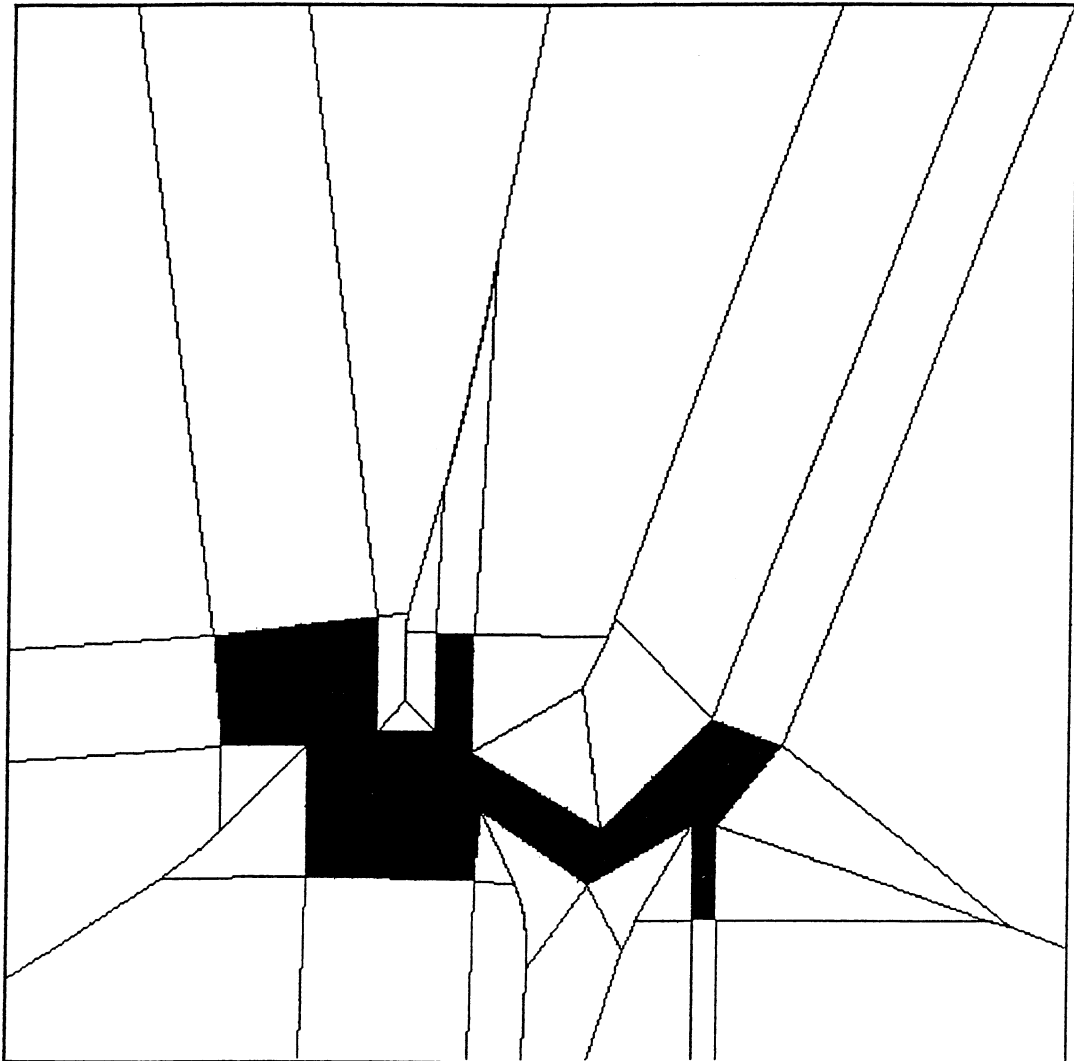


Figure 5.13 (f) $VOD_e(P_3)$

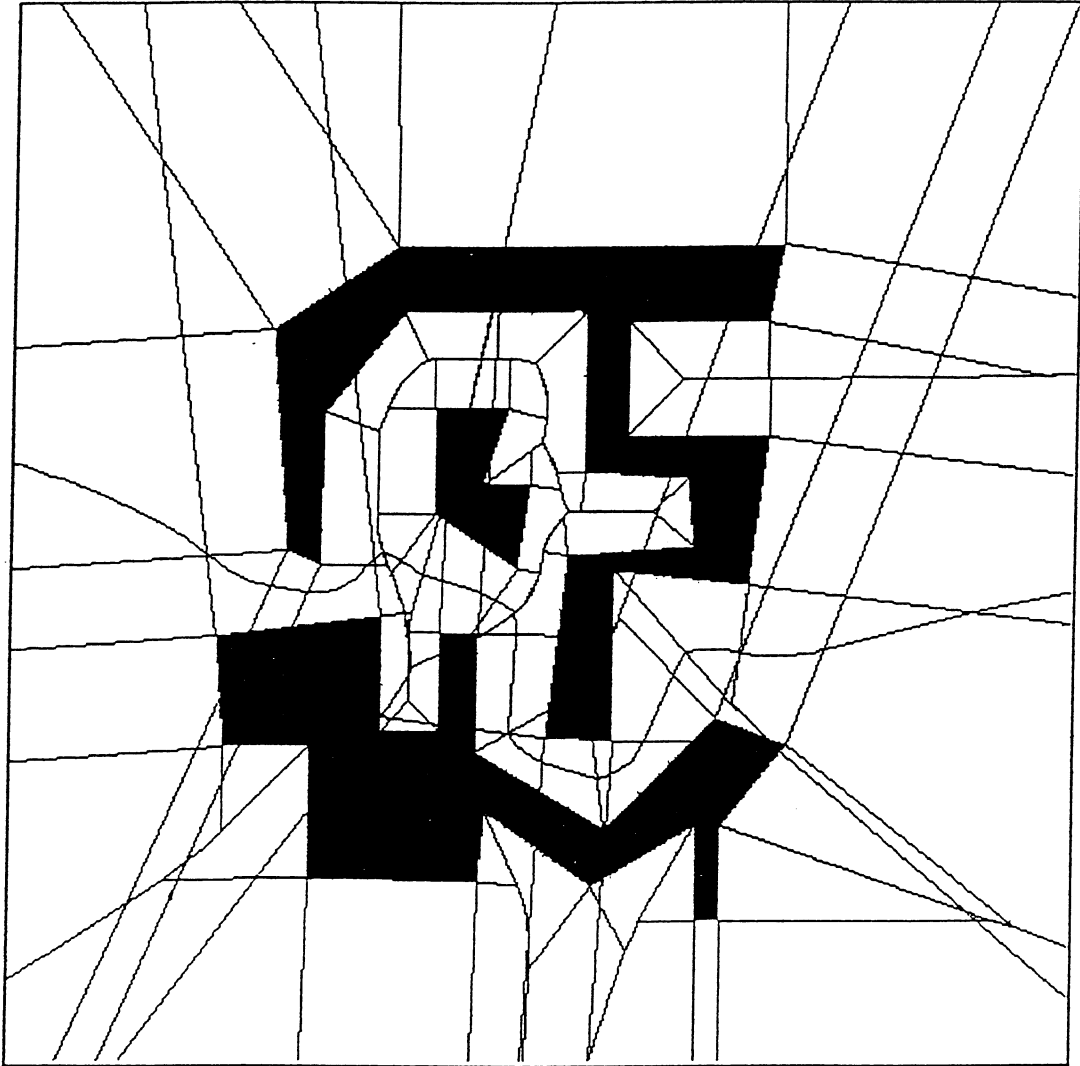


Figure 5.13 (g) Superposition de $VOD_e(P_1 \cup P_2)$ et $VOD_e(P_3)$, et représentation du séparateur $Sep(P_1 \cup P_2, P_3)$ (ouvert)

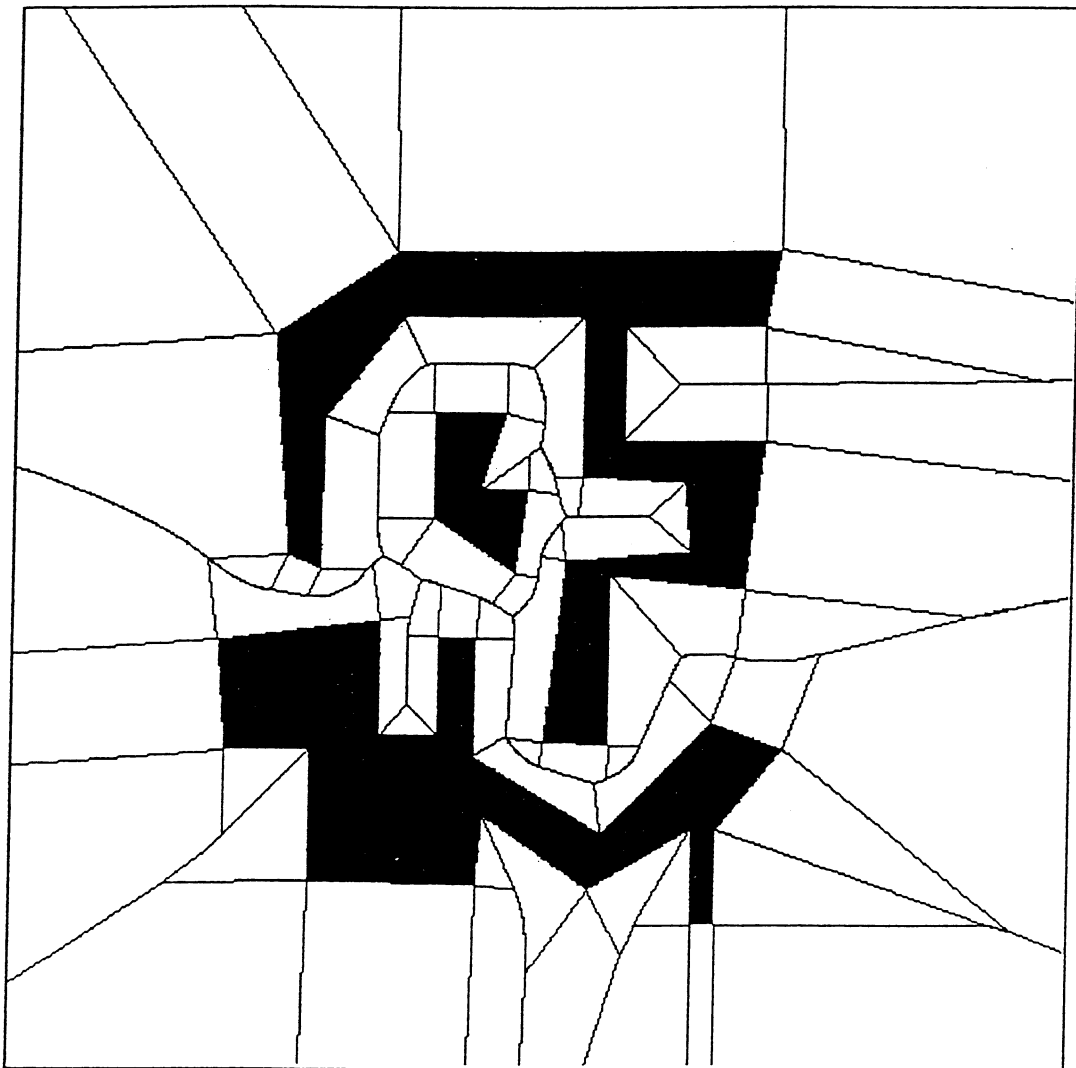


Figure 5.13 (h) $VOD_e(P_1 \cup P_2 \cup P_3)$ — le diagramme final basé sur les éléments

Chapitre 6

Diagramme de Voronoï Généralisé d'un Ensemble de Polygones

6.1 Introduction

Au chapitre 3, nous avons discuté du problème du diagramme de Voronoï d'un polygone basé sur les éléments. Suite à cela, nous allons étudier le problème de la construction du diagramme de Voronoï généralisé de polygones :

$VOD_o(G)$ — étant donné un ensemble G de N polygones simples disjoints du plan R^2 , trouver une partition de R^2 en N régions telles que tout point d'une région soit plus proche du polygone associé à cette région que de n'importe quel autre polygone.

Nous appelons $VOD_o(G)$ le diagramme de Voronoï généralisé basé sur les objets. Comme dans les cas de points, nous pouvons imaginer que beaucoup de problèmes géométriques se rapportent à l'abstraction du problème ci-dessus. Des exemples de problèmes géométriques sont donnés par :

- (a). **Calcul de distances.** Etant donné deux objets quelconques, trouver la distance minimum et la distance maximum entre les deux objets ;
- (b). **Paire d'objets les plus proches.** Etant donné N objets polygonaux du plan, trouver deux objets dont la distance soit minimale ;
- (c). **les plus proches voisins des objets.** Etant donné N objets dans le plan, trouver le voisin le plus proche de chacun;
- (d). **PATHFIND.** Etant donné N objets (obstacles) et une configuration initiale et une configuration finale d'un autre objet O du plan, déterminer s'il existe un mouvement continu de

l'objet O de sa configuration initiale à sa configuration finale, et déterminer un tel mouvement s'il existe. etc....

Ce dernier problème est important et crucial en robotique où un chemin sans obstacles doit être trouvé pour qu'un robot puisse réaliser un mouvement libre dans un environnement en présence d'obstacles.

Ce chapitre constitue l'aboutissement des différentes notions et algorithmes décrits au travers des précédents chapitres et paragraphes. Une approche originale est ainsi proposée pour la construction du diagramme de Voronoï généralisé d'un ensemble de polygones.

6.2 Schéma de calcul

La construction du diagramme de Voronoï généralisé basé sur les objets est plus complexe que celle du diagramme de Voronoï pour un ensemble de points. Un modèle de calcul direct à partir d'un ensemble d'objets donnés est difficile à concevoir. D'une part du fait que la bissectrice entre deux objets généralisés n'est pas aussi facilement exprimable que celle entre deux points ni que celle entre deux éléments; et d'autre part pour la raison que même si on peut exprimer la bissectrice entre deux objets généralisés sous quelque forme générale, il n'est pas facile de la calculer.

Cependant, nous allons voir dans le lemme 6.1 ci-dessous que le diagramme de Voronoï généralisé basé sur les objets est fondé sur le diagramme de Voronoï basé sur les éléments associés aux contours des objets donnés. Plus précisément, le calcul du diagramme de Voronoï généralisé basé sur les objets se ramène d'abord au calcul du diagramme de Voronoï basé sur les éléments correspondants et ensuite à l'extraction d'un sous diagramme de ce dernier. Donc, un schéma global de calcul du diagramme de Voronoï généralisé basé sur les objets peut être présenté selon le schéma donné en figure 6.1 où $VOD_e(G)$ (respectivement $VOD_o(G)$) désigne le diagramme de Voronoï généralisé basé sur les éléments (respectivement les objets).

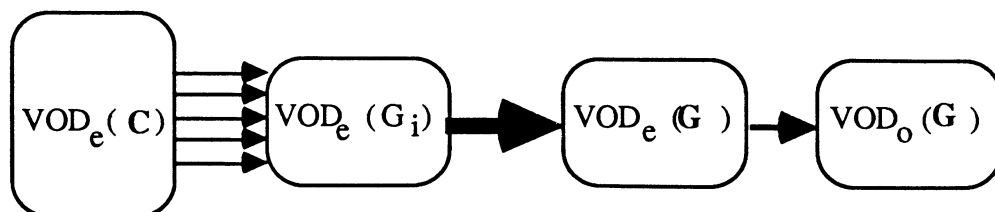


Figure 6.1 Schéma global pour calculer $VOD_o(G)$

C : chaîne d'éléments, $G=\{G_i\}$: l'ensemble d'objets

Lemme 6.1 Le graphe du diagramme de Voronoï d'un ensemble de polygones G basé sur les objets, $VOD_o(G)$, est un sous-graphe de celui basé sur les éléments noté $VOD_e(G)$ et obtenus par décomposition de la frontière de G .

Preuve. Le graphe de $VOD_o(G)$ peut s'exprimer par :

$$\partial[VOD_o(G)] = \{p \mid \min_{G_i \in G} d(p, G_i) = \min_{G_j \in G} d(p, G_j), \forall i \neq j\} \quad (6.1)$$

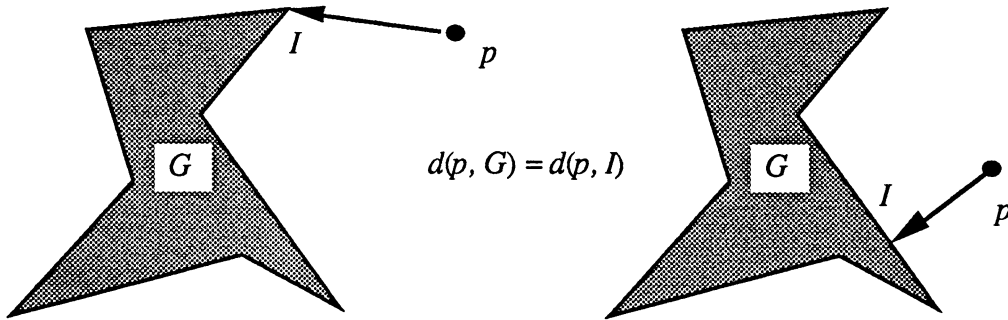


Figure 6.2 Distance entre un point p et un polygone G

En se référant à la figure 6.2, l'image $I(p, G)$ d'un point p sur un polygone (cf. la définition 2.4 au chapitre 2) est soit un sommet convexe soit située sur un segment ouvert du polygone. La distance $d(p, I)$ définit la distance entre point p et polygone G . D'après la définition de l'élément, on sait que l'image $I(p, G)$ est certainement incluse dans quelque élément de ∂G . Donc, (6.1) devient :

$$\partial[VOD_o(G)] = \{p \mid \min_{e_k \in \partial G_i} d(p, e_k) = \min_{e_l \in \partial G_j} d(p, e_l), \forall i \neq j\} \quad (6.2)$$

D'autre part, on a

$$\begin{aligned} \partial[VOD_e(G)] &= \{p \mid \min_{e_k \in \partial G} d(p, e_k) = \min_{e_l \in \partial G} d(p, e_l), \forall k \neq l\} \\ &= \{p \mid \min_{e_k \in \partial G_i} d(p, e_k) = \min_{e_l \in \partial G_j} d(p, e_l), \forall k \neq l, \forall i \neq j\} \cup \\ &\quad \{q \mid \min_{e_u \in \partial G_i} d(q, e_u) = \min_{e_v \in \partial G_i} d(q, e_v), u \neq v, \forall G_i \in G\} \\ &= \partial[VOD_o(G)] \cup E \end{aligned} \quad (6.3)$$

où

$$E = \{q \mid \min_{e_u \in \partial G_i} d(q, e_u) = \min_{e_v \in \partial G_j} d(q, e_v), u \neq v, \forall G_i \in \mathcal{G}\} \quad (6.4)$$

Il est évident que E n'est pas vide. Donc

$$VOD_e(\mathcal{G}) \supset VOD_o(\mathcal{G}).$$

Corollaire 6.1 Le graphe de $VOD_o(\mathcal{G})$ est composé des arêtes de $VOD_e(\mathcal{G})$ moins les arêtes partagées par deux éléments appartenant à un même polygone.

Preuve. Dans (6.3) on a :

$$\partial[VOD_e(\mathcal{G})] \cap E = \emptyset$$

où E représente toutes les arêtes de Voronoï qui sont partagées par deux éléments appartenant à un même objet. Donc,

$$VOD_o(\mathcal{G}) = VOD_e(\mathcal{G}) - E \quad (6.5)$$

Corollaire 6.2 Si $VOD_e(\mathcal{G})$ peut être construit en temps $O(T)$, alors $VOD_o(\mathcal{G})$ peut aussi être construit en temps $O(T)$ où T est une fonction du nombre des éléments dans \mathcal{G} .

Preuve. Examiner si deux éléments associés à une arête de Voronoï commune appartiennent à un même polygone est un test très simple. De plus, il y a $O(N)$ arêtes de Voronoï dans $VOD_e(\mathcal{G})$ où N est le nombre des éléments de \mathcal{G} (voir le lemme 2. 8). Un tel examen d'arêtes de Voronoï peut se faire au plus en $O(N)$. Ceci n'apporte pas de complexité supplémentaire à la complexité du calcul de $VOD_e(\mathcal{G})$.

En raison du fait que $VOD_o(\mathcal{G})$ est un sous-graphe de $VOD_e(\mathcal{G})$ nous allons nous consacrer à la construction du diagramme de Voronoï généralisé basé sur les éléments au cours de la suite de cette partie. Plus particulièrement, nous ne considérerons qu'un domaine défini par la frontière d'un polygone qui est soit à l'intérieur soit à l'extérieur de ce polygone.

6.3 Choix d'un modèle algorithmique

La stratégie "Divide-and-Conquer" consiste à diviser un ensemble original de taille K en deux sous-ensembles de tailles presque égales. Ensuite, la solution du problème original peut être obtenue en fusionnant les deux solutions des sous-problèmes. Le succès d'un algorithme basé

sur le schéma “Divide-and-Conquer” repose sur le fait que les solutions des deux sous-problèmes doivent être combinées en un temps linéaire afin de donner une solution pour le problème entier. Dans le cas où nous avons plusieurs polygones $G=\{G_i, i=1, 2, \dots, K\}$, il est difficile d'utiliser la stratégie “Divide-and-Conquer”. Donnons des raisons principales de cette remarque.

(a). Ordonner un ensemble de polygones n'a pas de signification évidente comme c'est le cas pour un ensemble de points. Si on examine la figure 6.3, on trouve qu'il est difficile de définir la position relative des polygones, notamment, P_2 est-il au dessus de P_1 et P_3 est-il situé à gauche de P_2 ?

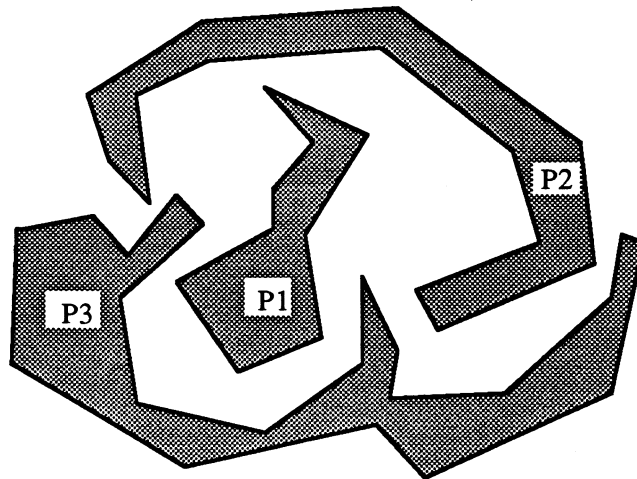


Figure 6.3 Les positions relatives des polygones sont ambiguës

En fait, les projections des polygones sur les axes OX et OY se chevauchent. Donc, il est difficile de bien préciser les positions relatives entre polygones.

(b). Même si on peut ordonner les polygones selon un critère simple, par exemple, une X-coordonnée minimale ou une Y-coordonnée minimale de chaque polygone, on ne peut pas assurer l'existence d'un algorithme en un temps linéaire qui combine les solutions des deux sous-problèmes pour résoudre le problème entier quand on utilise la stratégie de “Divide-and-Conquer”. La figure 6.4 illustre un tel exemple. Supposons que les polygones sont ordonnés par la Y-coordonnée maximale de chacun d'eux. Le schéma de “Divide-and-Conquer” va diviser $S=\{P_1, P_2, P_3, P_4\}$ en deux sous-ensemble, $S_1=\{P_1, P_2\}$ et $S_2=\{P_3, P_4\}$. Si le diagramme de Voronoï de S_1 , $VOD(S_1)$, et celui de S_2 , $VOD(S_2)$ sont déjà calculés, examinons l'étape de fusion de $VOD(S_1)$ et $VOD(S_2)$ pour obtenir le diagramme de Voronoï $VOD(S)$ des quatre polygones initiaux. En ce cas-ci, le séparateur entre S_1 et S_2 sera composé de deux courbes discontinues, C_1 et C_2 . Le calcul d'une des deux courbes du séparateur peut être effectué en un temps linéaire. Ici, les deux courbes doublent peut-être le temps de calcul.

De plus, dans le cas d'un ensemble de N polygones on peut avoir au plus $N/2$ courbes discontinues comme ceci est illustré sur la figure 6.5.

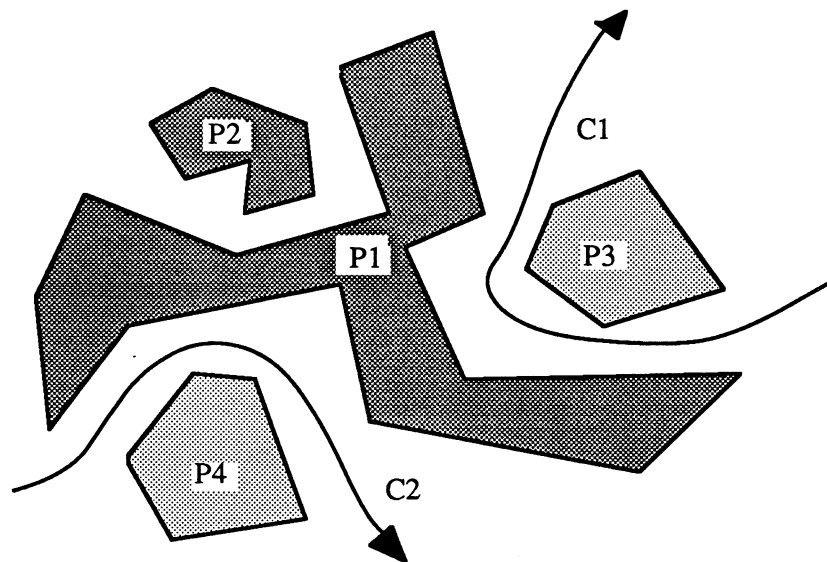


Figure 6.4 Le séparateur est composé de deux morceaux

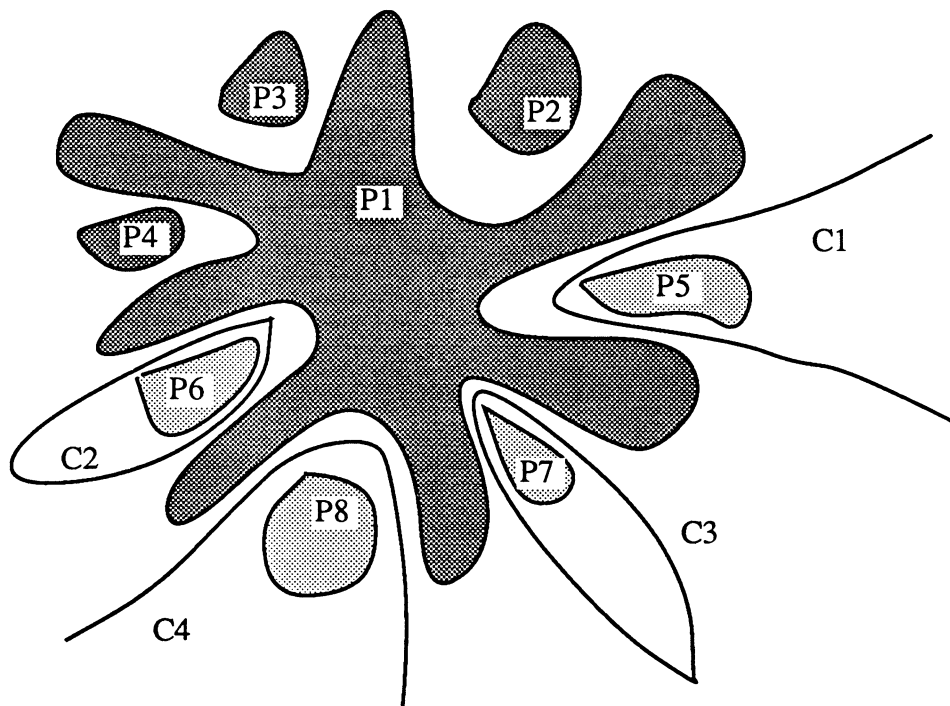


Figure 6.5 N polygones peuvent apporter un séparateur de $N/2$ courbes discontinues

(c). Par ailleurs, chaque courbe d'un séparateur est soit fermée soit ouverte. Par conséquent, même si on peut déterminer le nombre de courbes discontinues du séparateur par quelque algorithme simple, on doit examiner encore la fermeture de chaque courbe.

On doit donc s'attendre à rencontrer certaines difficultés quand on applique la stratégie de "Divide-and-Conquer" au cas de la construction du diagramme de Voronoï d'un ensemble de polygones. Par contre, on va voir dans le paragraphe suivant que l'utilisation de l'approche incrémentale combinée avec un prétraitement bien choisi peut assurer d'avoir toujours un séparateur connexe à chaque étape de la construction.

6.4 Modèle incrémental de construction

Dans ce qui suit, nous allons proposer un algorithme basé sur un modèle de calcul itératif pour construire le diagramme de Voronoï généralisé d'un ensemble de polygones. Soit G un ensemble de N polygones. Après un prétraitement qui ordonne l'ensemble des polygones donnés selon un critère spécifié, nous introduisons à l'étape K le K -ième polygone, à partir du diagramme de Voronoï des $K-1$ premiers polygones, et effectuons une modification de l'ancien diagramme de Voronoï pour obtenir le diagramme de Voronoï des K polygones. Le diagramme de Voronoï est mis à jour jusqu'à ce que tous les polygones donnés soient insérés. Du fait que nous n'insérons qu'un polygone ordonné à chaque étape du calcul, nous assurons que le séparateur entraîné à chaque étape sera constitué d'un seul morceau de courbe continue (voir le lemme 5.1).

Critère de prétraitement :

Soit $G = \{G_i\}$, $i=1, 2, \dots, K$ ($K > 2$), un ensemble de polygones initiaux. Nous introduisons un critère d'ordonnement des polygones durant l'étape de prétraitement de notre algorithme :

Soit $Y_{\min}(G)$ la coordonnée en y minimale des sommets du polygone G .

- (a) $Y_{\min}(G_i) < Y_{\min}(G_{i+1})$;
- (b) $X_{\min}(G_i) < X_{\min}(G_{i+1})$ si $Y_{\min}(G_i) = Y_{\min}(G_{i+1})$

En fait, nous pouvons utiliser l'un des quatre sommets extrêmes (le plus haut, le plus bas, le plus à droite et le plus à gauche) comme étiquette de chaque polygone.

Soit $CONV(m)$ l'enveloppe convexe des m premiers polygones ordonnés, $CH(i)$ celle du polygone de rang i (voir figure 6.6). Alors, d'après le lemme 4.10, nous savons que $CONV(m)$ et $CH(i)$ s'intersectent d'une "manière simple". De plus, bénéficiant de l'algorithme proposé dans le paragraphe 4.4.2 pour calculer l'enveloppe convexe de deux polygones d'intersection simple, nous pouvons résoudre les deux problèmes :

- (i) $CH(i)$ est-il inclus dans $CONV(i-1)$?
- (ii) Si non, trouver l'enveloppe convexe de la réunion de $CH(i)$ et $CONV(i-1)$.

Evidemment, toutes les propriétés d'un séparateur présentées au chapitre 5 tiennent aussi pour le séparateur introduit à chaque étape d'itération si les polygones sont ordonnés selon le critère de prétraitement. Par exemple, le lemme 5.2 permet d'accéder à la forme du séparateur de deux ensembles des polygones, L et R , quand leurs enveloppes convexes $CH(L)$ et $CH(R)$ sont d'intersection simple. La forme du séparateur de L et R est soit fermée si $CH(R)$ est inclus dans $CH(L)$, soit ouverte si $CH(R)$ n'est pas inclus dans $CH(L)$. Ici, nous avons $L=\{G_1, G_2, \dots, G_{i-1}\}$ et $R=\{G_i\}$. Donc, le lemme 5.2 peut y s'appliquer. Avec quelle complexité pouvons nous déterminer la forme d'un séparateur? Le lemme suivant donne la réponse.

Lemme 6.2 Etant donné un ensemble de N polygones, $G=\{G_1, G_2, \dots, G_N\}$, ordonnés selon le critère de prétraitement précité. Soit $L=\{G_1, G_2, \dots, G_{i-1}\}$ et $R=\{G_i\}$. Si $VOD(L)$ et $VOD(R)$ sont disponibles, la forme du séparateur $Sep(L, R)$ peut être déterminée en temps linéaire.

Preuve. Le lemme 2.9 montre qu'un élément e_i d'un ensemble S est situé dans l'enveloppe convexe de S si et seulement si $V(e_i)$ est non borné. Donc une des approches pour trouver $CONV(i-1)=CH(L)$ consiste à examiner simplement si une région de Voronoï est non bornée en examinant toutes les régions de Voronoï de $VOD(L)$. Ceci peut être fait en un temps proportionnel au nombre d'éléments de $VOD(L)$. De même, $CH(R)$ peut aussi être trouvé en temps linéaire. L'algorithme qui est proposé dans le paragraphe 4.4.2 peut résoudre à la fois le problème d'inclusion de $CH(R)$ dans $CH(L)$ et celui de l'enveloppe convexe de $CH(L \cup R)$ — ce qui est effectué en un temps linéaire avec un prétraitement. D'après le lemme 5.2, ceci est identique à la détermination de la forme d'un séparateur.

Rappelons l'algorithme proposé dans § 4.4.2 pour calculer l'enveloppe convexe de la réunion de deux polygones convexes d'intersection simple. Il consiste à trouver quatre points d'appui ordonnés suivant l'enveloppe convexe résultante. Une simple modification de cet algorithme donnera une ou deux paires d'éléments cruciaux. Ce sont d'une part la paire d'éléments de départ et celle d'arrêt pour un séparateur ouvert, et d'autre part une paire d'éléments d'initialisation pour un séparateur fermé. Une telle modification se présente dans l'algorithme FINDKEYS suivant : (voir la figure 6.6 pour illustration).

Algorithme FINDKEYS ;

entrée : $L=\{G_1, G_2, \dots, G_{i-1}\}, R=\{G_i\}$;

$VOD(L), VOD(R)$;

sortie : $e_{ls}, e_{rs}, e_{lf}, e_{rf}$;

Début

1. trouver $CH(R)$ et $CH(L)$;

2. trouver deux points d'appui dans $CH(L)$, notés par p_{lf} et p_{ls} (respectivement le point d'appui positif et le point d'appui négatif introduits par la définition 4.10), en analysant la séquence $L_{CH(R)}(v)$ ($v \in CH(L)$);

3. trouver deux points d'appui dans $CH(R)$, notés par p_{rf} et p_{rs} (respectivement le point d'appui positif et le point d'appui négatif introduits par la définition 4.10), en analysant la séquence $L_{CH(L)}(v)$ ($v \in CH(R)$);

4. Si $p_{lf} = \emptyset$ et $p_{ls} \neq \emptyset$ alors

début

a. trouver le point d'ancrage q_l dans $CH(L)$;

b. $p_{lf} \leftarrow q_l; p_{ls} \leftarrow q_l$;

fin

Sinon si $p_{rs} = \emptyset$ et $p_{lf} \neq \emptyset$ alors

début

a. trouver le point d'ancrage q_r dans $CH(R)$;

b. $p_{rf} \leftarrow q_r; p_{rs} \leftarrow q_r$;

fin

sinon si $p_{lf} = \emptyset$ et $p_{rs} = \emptyset$ alors {un séparateur fermé}

si $L_{CH(R)}(v)$ ($v \in CH(L)$)=1 alors { $CH(R)$ est dans $CH(L)$ }

trouver une paire d'éléments d'initialisation ;

5. $e_{ls} \leftarrow$ la représentation en élément de p_{ls} ;

$e_{lf} \leftarrow$ la représentation en élément de p_{lf} ;

$e_{rs} \leftarrow$ la représentation en élément de p_{rs} ;

$e_{rf} \leftarrow$ la représentation en élément de p_{rf} ;

Fin ;

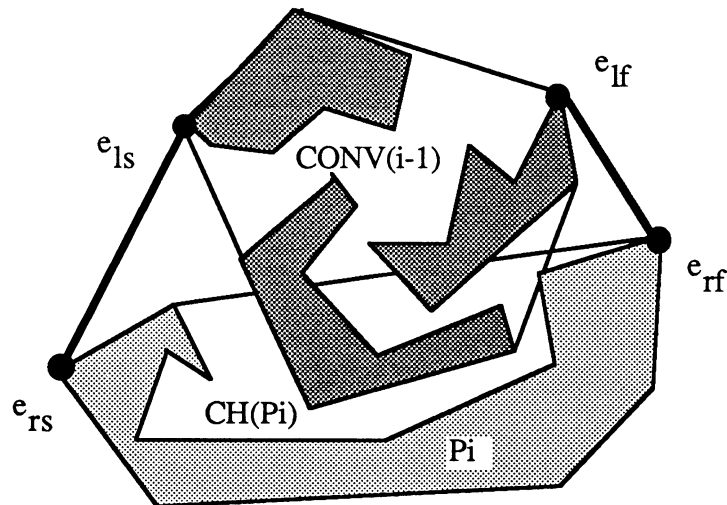


Figure 6.6 Illustration pour l'algorithme FINDKEYS

Lorsque nous avons obtenu la forme d'un séparateur, nous pouvons construire ce séparateur par la méthode proposée au chapitre 5. Si le séparateur est fermé, nous prenons un élément e_r de G_i dans $CH(G_i)$, puis nous examinons toutes les régions de $VOD(L)$ afin de déterminer un élément e_l , tel que l'intersection de la bissectrice $B(e_l, e_r)$ avec la demi-droite dans la région de Voronoï $V(e_r, \partial G_i - e_r)$ soit comprise dans la région de Voronoï $V(e_l, \partial L - e_l)$ (voir le lemme 5.5). L'élément e_l peut être trouvé en un temps proportionnel au nombre total de régions de Voronoï dans $VOD(L)$. D'autre part, quand le séparateur est ouvert, les éléments correspondant aux deux paires de points d'appui sont identiques aux éléments de départ et éléments d'arrêt. (voir le lemme 5.6). A partir de la bissectrice de départ déterminée par les éléments de départ, nous construisons le séparateur morceau par morceau jusqu'à ce que la courbe du séparateur soit entièrement constituée. Ce processus de fusion est semblable à celui adopté pour le diagramme de Voronoï de l'intérieur d'un polygone discuté au chapitre 3.

En résumé l'algorithme d'incrémental de calcul du diagramme de Voronoï d'un ensemble de polygones peut se formuler comme ci-dessous. Pour simplifier, nous utilisons les notations suivantes :

$$G(K) = \{G_1, G_2, \dots, G_K\};$$

$$\partial G(K) = \{\partial G_1, \partial G_2, \dots, \partial G_K\};$$

$$\varepsilon G(K) = \{\varepsilon G_1, \varepsilon G_2, \dots, \varepsilon G_K\};$$

$$VOD_i = VOD(\partial G_i \cap \varepsilon G_i);$$

$$VOD(K) = VOD(\partial G(K) \cap \varepsilon G(K));$$

Algorithme EXVODMP ;

entrée : $G(N) = \{G_1, G_2, \dots, G_N\};$

sortie : $VOD(N)=VOD(G(N))$;

Début

1. Prétraitement. Ordonner tous les polygones originaux selon le critère de prétraitement;
2. Calculer VOD_1 , le diagramme de Voronoï du premier polygone par l'algorithme EXVOD (cf. le chapitre 3) ;
3. pour $j=2$ à K faire
 - début
 - a. calculer VOD_j par l'algorithme EXVOD ;
 - b. fusionner $VOD(j-1)$ et VOD_j pour obtenir $VOD(j)$;
 - fin ;

Fin ;

L'étape 3.b consiste à calculer d'abord le séparateur entre $G(j-1)$, constitué des $j-1$ premiers polygones, et G_j , le polygone courant, et ensuite faire une modification locale du diagramme de Voronoï obtenu. Le détail de cette étape est présenté par la suite.

Algorithme MERGE_EX;

entrée : $VOD(j-1), VOD_j$;

sortie : $VOD(j)$;

Début

1. Trouver $CONV(j)$, l'enveloppe convexe de l'union de $CONV(j-1)$ et de $CH(j)$, par l'algorithme HULL_TWO_SIMPLE_INTERSECTION_CONVEX présenté dans le paragraphe 4.4.2. Cet algorithme donne à la fois (a) l'information si $CH(j)$ est inclus dans $CONV(j-1)$; et (b) quatre points d'appui qui déterminent une bissectrice de départ et une bissectrice d'arrivée pour un séparateur ouvert ;
2. Si $CH(j)$ est inclus dans $CONV(j-1)$, alors construire un séparateur fermé ;
3. Sinon construire un séparateur ouvert ;
4. Modifier toutes les régions de $VOD(j-1)$ et de VOD_j rencontrées par le séparateur entre $G(j-1)$ et G_j ;

Fin ;

La complexité de l'algorithme ci-dessus est facile à analyser. Après la première étape qui prend un temps proportionnel au nombre de sommets dans $CONV(j-1)$ et $CH(j)$, l'algorithme exécute alternativement l'étape 2 ou l'étape 3. Si l'étape 2 est sélectionnée, il faut d'abord trouver une paire d'initialisateurs et effectuer le calcul du séparateur à partir de cette paire d'éléments d'initialisation. D'après l'analyse de complexité dans § 5.3.1 et le théorème 5.1, nous savons que ceci peut être fait en un temps linéaire. Dans l'autre cas, si l'étape 3 est exécutée, nous avons seulement besoin de calculer un séparateur ouvert, ceci peut se faire également en temps linéaire (cf. le théorème 5.1). L'étape 4 est une opération simple sur pointeur. En effet cette étape peut être exécutée en même temps que la construction du séparateur. Par conséquent, l'algorithme MERGE_EX est de complexité linéaire, c'est-à-dire, l'étape 3.b. dans l'algorithme EXVODMP peut se calculer en un temps $O(\sum_{i=1}^j n_i)$ où n_i est le nombre des éléments composant le contour du polygone G_i .

A présent nous analysons la complexité de l'algorithme EXVODMP. Le prétraitement dans l'étape 1 peut être fait en calculant d'abord les y-coordonnées minimales parmi les sommets de chaque polygone. Ceci peut prendre au total un temps en $O(\sum_{i=1}^K n_i)$. Ensuite on ordonne ces polygones selon l'ordre ascendant, ce qui peut se faire en un temps $O(K \log K)$. Donc, le temps total pour l'exécution du prétraitement est en $O(K \log K + \sum_{i=1}^K n_i)$. Remarquons que cette étape est indépendante du reste de l'algorithme. L'étape 3 est une boucle itérative. L'étape 3.a. dans l'algorithme est un calcul de diagramme de Voronoï de l'extérieur du j -ième polygone ordonné. Cette étape peut être effectuée en $O(n_j \log n_j)$ où n_j est le nombre des éléments du j -ième polygone (voir le lemme 3.5). Donc un pas d'itération dans l'étape 3 nécessite au total $O(n_j \log n_j + \sum_{i=1}^j n_i)$ opérations. Supposons $K \geq 2$. Nous pouvons écrire la formule de récurrence suivante pour exprimer le temps de calcul de la boucle itérative dans l'étape 3 :

$$T(K) \leq T(K-1) + C_1 (n_K \log n_K) + C_2 (\sum_{i=1}^K n_i) \quad (6.6)$$

où $T(K)$ représente le temps d'exécution sur un ensemble de K polygones, C_1 et C_2 sont des constantes positives. La quantité $(n_K \log n_K)$ est due au calcul de VOD_K . La quantité $\sum_{i=1}^K n_i$ est la contribution nécessaire pour la fusion de $VOD(K)$ et VOD_K , et n_i représente le nombre d'éléments composant le contour du polygone G_i .

En remplaçant K par $K-1$, (6.6) devient :

$$T(K) \leq T(K-2) + C_1 (n_{K-1} \log n_{K-1} + n_K \log n_K) + C_2 (\sum_{i=1}^{K-1} n_i + \sum_{i=1}^K n_i) \quad (6.7)$$

En répétant jusqu'à $K=1$, on obtient :

$$T(K) \leq T(1) + C_1 (\sum_{i=2}^K n_i \log n_i) + C_2 (\sum_{h=2}^K \sum_{i=1}^h n_i) \quad (6.8)$$

où $T(I)$ vérifie l'inégalité :

$$T(I) \leq C_1 (n_I \log n_I) \quad (6.9)$$

Donc, nous avons :

$$T(K) \leq C_1 (\sum_{i=1}^K n_i \log n_i) + C_2 (\sum_{h=2}^K \sum_{i=1}^h n_i) \quad (6.10)$$

Soit $N = \sum_{i=1}^K n_i$ le nombre total des éléments de K polygones. Alors, nous avons le lemme suivant qui donne le temps de calcul de l'algorithme EXVODMP :

Lemme 6.3 Etant donné un ensemble de K polygones quelconques, noté $G(K)$. Le diagramme de Voronoï généralisé des polygones basé sur éléments, $VOD_e(G(K))$ par l'algorithme EXVODMP peut être construit dans le cas le plus défavorable en $O(\sum_{i=1}^K n_i \log n_i + \sum_{h=2}^K \sum_{i=1}^h n_i)$ où n_i est le nombre des éléments composant le contour du i -ème polygone.

Preuve. Le temps total d'exécution de l'algorithme EXVODMP est composé du temps de prétraitement et du temps de l'itération. Le temps de prétraitement est $O(K \log K + \sum_{i=1}^K n_i)$. Remarquons que l'inégalité ci-dessous tient si $K > 2$:

$$(K \log K + \sum_{i=1}^K n_i) \leq (\sum_{i=1}^K n_i \log n_i + \sum_{h=2}^K \sum_{i=1}^h n_i) \quad (6.11)$$

Donc, la complexité de l'algorithme EXVODMP est celle de l'itération qui est

$$O(\sum_{i=1}^K n_i \log n_i + \sum_{h=2}^K \sum_{i=1}^h n_i) \quad (6.12)$$

Corollaire 6.3 Le diagramme de Voronoï basé sur les objets de K polygones quelconques peut être construit par l'algorithme EXVODMP en temps $O(\sum_{i=1}^K n_i \log n_i + \sum_{h=2}^K \sum_{i=1}^h n_i)$ dans le cas le plus défavorable où n_i est le nombre des éléments composant le contour du i -ième polygone.

Preuve. Immédiate à partir du corollaire 6.2 et du lemme 6.3.

6.5 Cas particulier — diagramme de Voronoï de polygones convexes

Bien que nous avons présenté un algorithme d'incrémental combinée avec l'approche l'approche Divide-and-Conquer en temps $O(\sum_{i=1}^K n_i \log n_i + \sum_{h=2}^K \sum_{i=1}^h n_i)$ dans le cas le plus défavorable pour construire le diagramme de Voronoï d'un ensemble de polygones quelconques, nous allons obtenir une solution plus performante si tous les polygones sont convexes. Dans ce cas, la construction de VOD_j décrite dans l'étape 3.a de l'algorithme

EXVODMP ne prend qu'un temps linéaire. Par conséquent, le temps de calcul pour le diagramme de Voronoï de K polygones convexes peut s'exprimer en remplaçant $(n_K \log n_K)$ par n_K dans (6.6). On a alors :

$$T(K) \leq T(K-1) + C_3 (n_K) + C_4 (\sum_{i=1}^K n_i) \quad (6.13)$$

En développant les calculs d'une manière similaire, nous obtenons :

$$T(K) \leq C_3 (\sum_{i=1}^K n_i) + C_4 (\sum_{h=2}^K \sum_{i=1}^h n_i) \quad (6.14)$$

Lemme 6.4 Etant donnés K polygones convexes, $G = \{G_1, G_2, \dots, G_K\}$, le diagramme de Voronoï généralisé (pour l'extérieur) basé sur les éléments, en utilisant l'algorithme EXVODMP, $VOD_e(G)$ peut être trouvé en temps $O(\sum_{h=1}^K \sum_{i=1}^h n_i)$ dans le cas le plus défavorable où n_i est le nombre de sommets du polygone G_i .

Preuve. Soit m_i le nombre de sommets du polygone G_i , et n_i le nombre d'éléments de G_i . Pour un polygone convexe, on a $n_i = 2m_i$. (6.14) devient donc :

$$\begin{aligned} T(K) &\leq 2C_3 (\sum_{i=1}^K m_i) + 2C_4 (\sum_{h=2}^K \sum_{i=1}^h m_i) \\ &\leq 2C_3 (\sum_{i=1}^K m_i) + 2C_4 (\sum_{h=1}^K \sum_{i=1}^h m_i) \end{aligned} \quad (6.15)$$

Si $K \gg 1$, alors :

$$(\sum_{i=1}^K m_i) \ll (\sum_{h=1}^K \sum_{i=1}^h m_i)$$

On a donc :

$$O(T(N)) \leq (\sum_{h=1}^K \sum_{i=1}^h m_i).$$

Corollaire 6.4 Le diagramme de Voronoï généralisé basé sur les objets de K polygones convexes peut être construit par l'intermédiaire de l'algorithme EXVODMP dans le cas le plus défavorable en temps $O(T(N)) = (\sum_{h=1}^K \sum_{i=1}^h m_i)$ où m_i est le nombre de sommets du polygone G_i .

Preuve. Immédiate à partir du corollaire 6.2 et du lemme 6.4.

Cas de segments de droite :

Un segment de droite est l'exemple du polygone convexe le plus simple. Nous allons voir qu'une meilleure solution que (6.15) peut être obtenue pour ce cas particulier. Soit $s = [a, b]$ ($a \neq b$) un segment de droite orienté. Ajoutons un autre segment de droite $[b', a']$ au segment original afin que s soit considéré comme un polygone convexe ayant deux sommets et deux

arêtes (voir figure 6.7). Autrement dit, un segment de droite $[a, b]$ est considéré comme un ensemble de quatre éléments: a , (a, b) , b , et (b, a) . En fait, quand nous appliquons l'algorithme EXVODMP à un ensemble de segments, nous considérons toujours que chaque segment est composé de quatre éléments différents définis comme ci-dessus. Pour ce cas-ci, nous avons le lemme suivant.

Lemme 6.5 Le diagramme de Voronoï de K segments de droite peut être construit en un temps $O(K^2 / 2)$ dans le cas le plus défavorable.

Preuve. Soit $I(K)$ le temps d'itération de l'algorithme EXVODMP, $P(K)$ le temps de prétraitement. D'après (6.14), nous avons :

$$\begin{aligned} I(K) &\leq 4C_3 K + 4C_4 (K^2 + K - 2) / 2 \\ &\leq C_a K^2 / 2 + C_b K \end{aligned} \tag{6.16.a}$$

où $C_a = 4C_4$ et $C_b = 4(C_3 + C_4 / 2)$.

Le temps de prétraitement est :

$$P(K) \leq C_1 K \log K + C_2 K \tag{6.16.b}$$

Donc, le temps total de calcul est :

$$\begin{aligned} T(K) &= I(K) + P(K) \\ &\leq C_a K^2 / 2 + C_b K + C_1 K \log K + C_2 K \\ &\leq C_a K^2 / 2 + C_1 K \log K + C_{b'} K \end{aligned} \tag{6.16.c}$$

où $C_{b'} = C_b + C_2$. Par conséquent, $O(T(K)) \leq O(K^2 / 2)$.

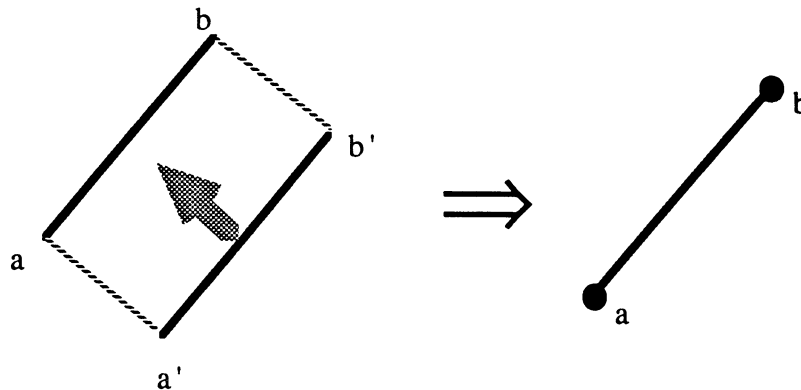


Figure 6.7 Un segment de droite est un polygone le plus simple. Il est composé de quatre éléments différents.

6.6 Construction semi-dynamique : incrémentation aléatoire d'un objet

6.6.1 Introduction

Une *construction dynamique* identifie une opération qui permet d'insérer et de supprimer un objet de façon aléatoire. Dans cette partie, nous allons discuter du problème de la construction semi-dynamique. Nous parlons de la construction semi-dynamique car nous allons présenter des algorithmes qui peuvent supporter seulement l'opération d'insertion d'un objet. Nous appelons aussi une telle opération, le calcul d'incrémentation aléatoire. Le problème est en général exprimé comme suit :

Supposons que le diagramme de Voronoï d'un ensemble d'objets S , $VOD(S)$, ait été construit à un instant donné. Quand un nouvel objet O est inséré dans l'ensemble S , on veut construire le diagramme de Voronoï pour l'ensemble $S \cup \{O\}$, noté $VOD(S \cup \{O\})$.

La construction d'incrémentation aléatoire a récemment attiré l'attention et l'intérêt de certains chercheurs. Ce n'est pas seulement parce que les algorithmes d'incrémentation sont en principe simples, faciles à programmer et efficaces en pratique, mais, c'est aussi parce que la construction d'incrémentation aléatoire est issue d'un besoin lié à la réalité. Par exemple, dans un environnement de travail en ligne, il est inévitable d'utiliser la technique de la construction d'incrémentation aléatoire.

Soit $VOD(S)$ le diagramme de Voronoï pour l'ensemble de S , $VOD(O)$ le diagramme de Voronoï pour l'objet à insérer O . Le problème de la construction de $VOD(S \cup \{O\})$ peut être considéré comme une opération $*$ entre $VOD(S)$ et $VOD(O)$:

$$VOD(S \cup \{O\}) = VOD(S) * VOD(O) \quad (6.17)$$

Si $S \cap \{O\} = \emptyset$, alors la région de Voronoï de S par rapport à O , $V(S, O)$, et celle de O par rapport à S , $V(O, S)$ seront définies sans ambiguïté puisque la distance euclidienne entre deux points appartenant respectivement à S et O est bien définie.

Lemme 6.6 Si $S \cap \{O\} = \emptyset$, alors $V(S, O) \cap V(O, S) = \emptyset$.

Preuve. Supposons que $V(S, O) \cap V(O, S) \neq \emptyset$, on a immédiatement que $S \cap \{O\} \neq \emptyset$.

Corollaire 6.5 Si $S \cap \{O\} = \emptyset$, alors $R^2 = V(S, O) \cup V(O, S) \cup Sep(S, O)$.

Preuve. Immédiate à partir du lemme 6.6.

Corollaire 6.6 Si $S \cap \{O\} = \emptyset$, alors $VOD(S \cup \{O\}) = \{VOD(S) \cap V(S, O)\} \cup \{VOD(O) \cap V(O, S)\}$.

Preuve. Immédiate d'après le lemme 5.8 et le corollaire 6.5.

Dans ce qui suit, nous allons étudier trois sortes d'insertions aléatoires : l'insertion d'un point, l'insertion d'un segment de droite, et l'insertion d'un polygone convexe. Il sera toujours supposé que $S \cap \{O\} = \emptyset$ dans une insertion aléatoire.

6.6.2 Insertion aléatoire d'un point

Soit $CH(S)$ l'enveloppe convexe d'un ensemble d'objets existant S . Il n'y a alors que deux possibilités pour la position d'un point à insérer p par rapport à S . Ce sont (a) $p \in CH(S)$; (b) le contraire. Selon la méthode présentée au chapitre 5, le séparateur entre S et p est facile à construire. D'après le lemme 2.9, nous savons qu'un élément est dans une enveloppe convexe si la région de Voronoï associée est non bornée. Donc, si nous avons localisé une région de Voronoï contenant le point p à insérer, nous pouvons déterminer si ce point appartient ou non à l'enveloppe convexe. Par conséquent, l'insertion aléatoire d'un point est simple. Un algorithme pour insérer un point est donné ci-après.

Algorithme RANDOM_POINT_INSERTION ;

entrée : S , l'ensemble d'objets existant ;

$VOD(S)$, diagramme de Voronoï de S ;

p , un point à insérer ;

sortie : $VOD(S \cup \{p\})$, diagramme de Voronoï de $S \cup \{p\}$;

Début

1. Localiser p dans une région de Voronoï de $VOD(S)$. Soit $V(e, S - \{e\})$ une telle région ;
2. Si $V(e, S - \{e\})$ est borné, alors calculer le séparateur fermé $Sep(S, p)$
sinon calculer le séparateur ouvert $Sep(S, p)$;
3. Modifier localement $VOD(S)$ pour obtenir $VOD(S \cup \{p\})$ en mettant à jour toutes les régions de $VOD(S)$ par lesquelles passe le séparateur ;

Fin;

L'étape 1 peut se faire en balayant toutes les régions de Voronoï dans $VOD(S)$. Ceci peut être fait dans le cas le plus défavorable en un temps $O(N)$ où N est le nombre d'éléments de S . L'étape 2 s'exécute en temps linéaire d'après le théorème 5.1. La modification d'une région de

Voronoi est seulement une opération sur pointeur. Donc, l'insertion aléatoire d'un point peut être réalisée en un temps linéaire dans le cas le plus défavorable.

6.6.3 Insertion aléatoire d'un segment

Soit L un segment à insérer, p_1 et p_2 les deux points d'extrémités de L , et S un ensemble d'objets originaux. Il est clair que les positions de L par rapport à $CH(S)$ sont :

- (a) POSITION1. $L \cap \zeta CH(S) \neq \emptyset$, et p_1 et p_2 sont inclus dans $CH(S)$, ce qui implique que L est à l'intérieur de $CH(S)$;
- (b) POSITION2. $L \cap \zeta CH(S) \neq \emptyset$, p_1 est dans $CH(S)$ mais p_2 n'est pas dans $CH(S)$ ou le contraire. C'est-à-dire, une seule extrémité de L est dans $CH(S)$;
- (c) POSITION3. $L \cap \zeta CH(S) = \emptyset$, c'est-à-dire que L est en dehors de $CH(S)$;
- (d) POSITION4. $L \cap \zeta CH(S) \neq \emptyset$, $p_1 \cap \zeta CH(S) = \emptyset$ et $p_2 \cap \zeta CH(S) = \emptyset$, c'est-à-dire, L sépare S en deux sous-ensembles d'objets invisibles. p_1 et p_2 sont en dehors de $CH(S)$.

Les cas (a), (b), (c) concernent un séparateur composé d'une seule branche de courbe soit fermée soit ouverte. En utilisant la technique introduite au chapitre 5, nous pouvons construire facilement le séparateur entre S et L .

Le cas (d) est un peu plus compliqué que les autres. Dans ce cas-ci, on va voir que le séparateur $Sep(S, L)$ sera composé de deux branches de courbes déconnectées. Précisément, on a le lemme suivant :

Lemme 6.7 Soit S un ensemble d'objets existant, L un segment à insérer ayant deux extrémités : p_1 et p_2 . (i) Si $L \cap \zeta CH(S) \neq \emptyset$; (ii) et si $p_1 \cap \zeta CH(S) = \emptyset$ et $p_2 \cap \zeta CH(S) = \emptyset$ alors le séparateur $Sep(S, L)$ est formé de deux branches de courbes continues et ouvertes.

Preuve. Sous la condition donnée : p_1 et p_2 sont en dehors de $CH(S)$ et L sépare S en deux sous-ensembles invisibles S_1 et S_2 si bien que S_1 et S_2 sont répartis de chaque côté de L (voir figure 6.8). Comme p_1 et p_2 sont en dehors de S , ils sont aussi en dehors de S_1 et S_2 . De plus, il est évident que $CH(S_1) \cap L = \emptyset$ et $CH(S_2) \cap L = \emptyset$. D'après les lemmes 5.1 et 5.2, le séparateur entre S_1 et L , $Sep(S_1, L)$, est continu et ouvert. De même $Sep(L, S_2)$ est aussi continu et ouvert. Comme S_1 et S_2 sont invisibles, alors il n'existe pas d'arête partagée par un élément de S_1 et un autre élément de S_2 . Par conséquent, $Sep(S_1, L)$ et $Sep(L, S_2)$ ne sont pas identiques.

Une fois que nous avons déterminé qu'il y a intersection entre L et S et que les deux extrémités de L sont en dehors de $CH(S)$, nous pouvons calculer respectivement les deux branches de courbes du séparateur selon l'algorithme présenté au chapitre 5. Pour trouver $Sep(S_1, L)$, nous calculons d'abord deux points d'appui dans $CH(S_1)$ qui sont définis dans § 4.3.2, notés par

p_{ls} et p_{lf} (voir figure 6.8). A partir de la première bissectrice déterminée par une paire d'éléments $\langle p_{ls}, p_l \rangle$, la branche continue du séparateur $Sep(S_1, L)$ est construite morceau par morceau jusqu'à ce que la dernière bissectrice $B(p_{lf}, p_2)$ soit atteinte. Ensuite, en ôtant la portion de $V(S_1, S_2)$ qui est à droite du séparateur $Sep(S_1, L)$ et la portion de $VOD(L)$ qui est à gauche du séparateur $Sep(S_1, L)$, nous obtenons le diagramme de Voronoï $VOD(S_1 \cup L)$. Un processus similaire de fusion de $V(S_2, S_1)$ et $VOD(L)$ peut s'effectuer pour obtenir $VOD(S_2 \cup L)$. Finalement, le diagramme de Voronoï entier, $VOD(S \cup L) = \{VOD(S_1 \cup L)\} \cup \{VOD(S_2 \cup L)\}$ sera construit.

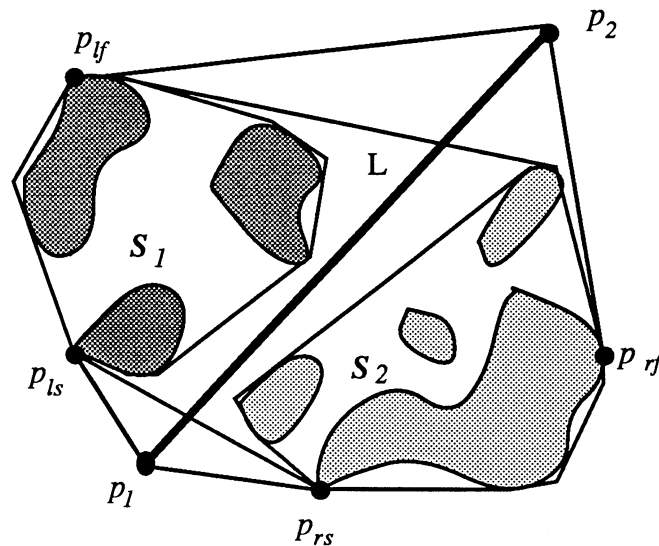


Figure 6.8 L divise S en deux sous-ensembles d'objets invisibles

Une double question liée à la fusion est la suivante : Est-ce que des régions de Voronoï de l'ensemble S_2 sont coupées par le séparateur entre S_1 et L ? De même, Est-ce que des régions de Voronoï de l'ensemble S_1 sont coupées par le séparateur entre S_2 et L ? La réponse est négative! Ceci est démontré dans le lemme 6.8.

Lemme 6.8 Le séparateur entre S_1 et L , $Sep(S_1, L)$ ne coupe aucune région de Voronoï associée à un élément du sous-ensemble S_2 . De façon similaire, $Sep(L, S_2)$ n'intersecte aucune région de Voronoï associée à un élément du sous-ensemble S_1 .

Preuve. La droite qui contient L divise le plan en deux demi-plans, h_1 et h_2 tels que S_1 est inclus entièrement dans h_1 et S_2 est inclus entièrement dans h_2 (voir figure 6.9). Soit z un point arbitraire dans $Sep(S_1, L)$. on a alors :

$$d(S_1, z) = d(L, z) \quad (6.18)$$

Supposons maintenant que ce point soit situé aussi dans une région de Voronoï associée à un élément de l'ensemble S_2 . Cette hypothèse implique :

$$d(S_2, z) < d(S_1, z) \quad (6.19)$$

D'après (6.18) et (6.19), on a :

$$d(S_2, z) < d(L, z) \quad (6.20)$$

Cependant, ceci est en contradiction avec le fait que $Sep(S_1, L)$ doit s'étendre complètement dans h_1 . Par conséquent, il n'existe pas de point situé dans une région de Voronoï associée à un élément de l'ensemble S_2 . De même, on a une conclusion similaire pour $Sep(L, S_2)$.

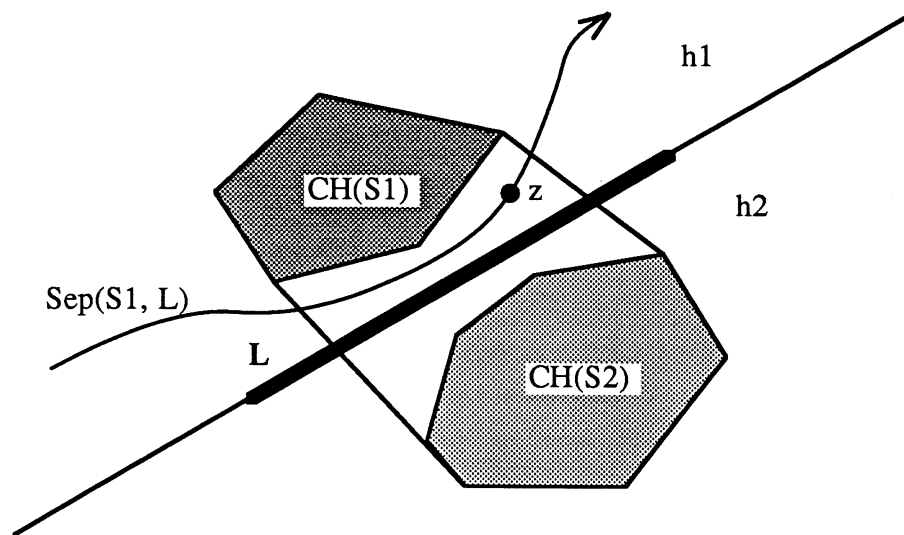


Figure 6.9 Illustration pour lemme 6.8

Maintenant, nous pouvons donner l'algorithme pour construire le diagramme de Voronoï $VOD(S \cup L)$ quand un segment de droite L est à insérer aléatoirement.

Algorithme RANDOM_SEGMENT_INSERTION ;

entrée : $VOD(S)$;

L , le segment à insérer ;

sortie : $VOD(S \cup L)$;

Début

1. déterminer la position de L par rapport à $CH(S)$, il y a quatre possibilités : POSITION1, POSITION2, POSITION3 et POSITION4 ;
2. a. Si POSITION1 alors calculer le séparateur fermé $Sep(S, L)$;
 b. sinon si POSITION2 ou POSITION3 alors calculer le séparateur ouvert $Sep(S, L)$;
 c. sinon si POSITION4 alors
 début

- 1). diviser S en deux sous-ensembles S_1 et S_2 par L tels que S_1 et S_2 sont répartis respectivement de chaque côté de L ;
- 2). calculer la première branche de $Sep(S, L)$, $Sep(S_1, L)$, qui est ouverte et calculer la deuxième branche de $Sep(S, L)$, $Sep(L, S_2)$, qui est aussi ouverte ; $Sep(S, L) = Sep(S_1, L) \cup Sep(L, S_2)$;

fin ;

3. mettre à jour toutes les régions de Voronoï par lesquelles passe le séparateur ;

Fin;

Le temps de calcul de cet algorithme est facile à analyser. Car chaque étape de l'algorithme peut se faire en temps linéaire, donc le temps de calcul total est aussi linéaire. Nous avons donc le lemme suivant.

Lemme 6.9 Etant donné le diagramme de Voronoï généralisé, $VOD(S)$, d'un ensemble d'objets S , le nouveau diagramme de Voronoï, $VOD(S \cup L)$ où L est un segment de droite à insérer aléatoirement, peut être construit en temps linéaire dans le cas le plus défavorable.

Corollaire 6.8 La construction d'incrémental aléatoire du diagramme de Voronoï généralisé de N segments de droite est dans $O(N^2 / 2)$ dans le cas le plus défavorable.

Preuve. Selon notre schéma de calcul, chaque segment contient quatre éléments (voir la figure 6.7). Pour insérer un nouveau segment de droite dans un ensemble de K segments existant, le temps de calcul est :

$$t(K+1) = 4C(K+1) \quad (6.21)$$

D'après le lemme 6.9, nous avons une formule itérative pour le temps de calcul du diagramme de Voronoï de N segments :

$$T(N) \leq T(N-1) + t(N) = T(N-1) + C'N \quad (6.22)$$

La solution de (6.22) est :

$$T(N) \leq C'(N/2 + N^2/2) \quad (6.23)$$

Donc,

$$O(T(N)) = O(N^2/2).$$

6.6.4 Insertion aléatoire d'un polygone convexe

Dans le paragraphe précédent, nous avons montré qu'il y a au plus deux branches de courbe pour un séparateur quand un nouveau segment de droite est inséré aléatoirement dans un ensemble d'objets existant. Que va-t-il se passer quand un polygone doit être inséré dans un ensemble d'objets existant ? Dans ce qui suit, nous allons discuter ce dernier problème. Plus particulièrement, nous nous contraindrons au cas d'insertion aléatoire d'un polygone convexe.

Soit $CONV(S)$ l'enveloppe convexe de l'ensemble d'objets existant S , $CH(P)$ l'enveloppe convexe du polygone P à insérer. Rappelons que la règle d'insertion est :

- (a). $\partial S \cap \partial P = \emptyset$ et
- (b). $\zeta P_i \cap \zeta P_j = \emptyset$ ($i \neq j$) pour $P_i, P_j \in S \cup \{P\}$.

Remarquons qu'il y a peut-être intersection entre $CH(P)$ et $CONV(S)$ bien que la frontière du polygone P à insérer n'intersecte pas celle de S . Dans tous les cas, l'intersection de $CH(P)$ et $CONV(S)$ est soit une *intersection simple* (cf. la définition 4.11) soit une intersection non simple. En outre, si $CH(P)$ et $CONV(S)$ s'intersectent d'une façon simple, la forme du séparateur entre P et S est alors facile à trouver par la méthode proposée au chapitre 5. Cependant l'intersection entre deux polygones convexes n'est pas toujours simple. Donc l'insertion aléatoire d'un polygone P convexe dans un ensemble d'objets S existant apportera en général un séparateur plus compliqué. Dans ce qui suit, nous ne considérons que le cas où $CH(P)$ et $CONV(S)$ sont d'intersection non simple. Comme nous l'avons dit, $\partial CONV(S)$ et $\partial CH(P)$ sont supposées avoir une orientation commune. Sans perte de généralité, nous supposons qu'elles aient une orientation CW.

Définition 6.1 L'*intersection positive* du polygone P , notée par I_p , est un point d'intersection duquel une arête de P entre dans l'intérieur de $CONV(S)$. L'*intersection négative* de P , notée par I_n , est un point d'intersection duquel une arête de P sort de $CONV(S)$.

La figure 6.10 montre un exemple d'intersection de deux polygones convexes. Dans cet exemple, P ayant 5 sommets possède 4 intersections positives et 4 intersections négatives. Donc, les intersections positives et les intersections négatives apparaissent en paire. En général, nous avons :

Lemme 6.10 Le polygone convexe P de N sommets a au plus N paires d'intersections positives et d'intersections négatives.

Preuve. Intuitive.

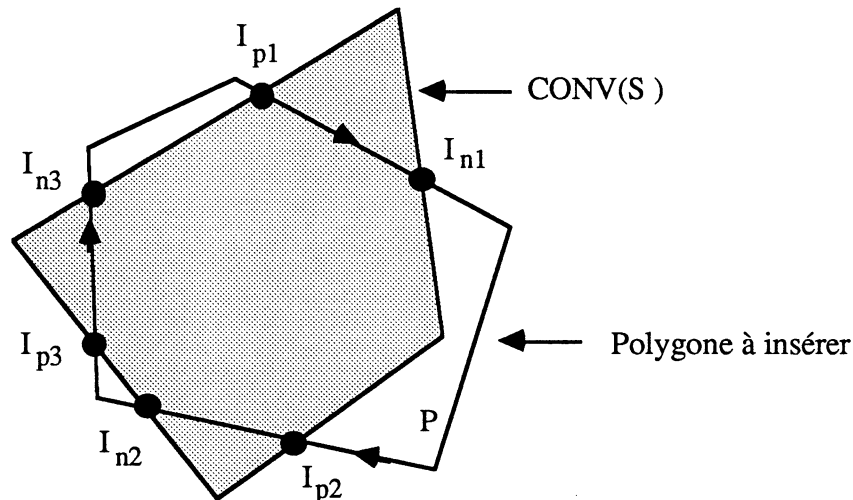


Figure 6.10 Les intersections positives et l'intersections négatives de deux polygones convexes.

Il faut noter que la règle d'insertion ne permet pas d'insérer un polygone P tel que $P \supset \text{CONV}(S)$. Par conséquent, si P et $\text{CONV}(S)$ sont d'intersection non simple, alors $\text{CONV}(S)$ sera coupé en $K+1$ parties dont l'une est $P \cap \text{CONV}(S)$ et K parties sont à l'extérieur de $P \cap \text{CONV}(S)$. Les K morceaux hors de $P \cap \text{CONV}(S)$ sont appelés les *faucilles* [O'ROURKE 82]. Ici K est délimité par les deux nombres : (a) le nombre des polygones dans S , N_S ; (b) le nombre des sommets de P , N_P :

$$K \leq \text{Min} \{ N_S, N_P \} \quad (6.24)$$

L'algorithme proposé par O'Rourke *et al* [O'ROURKE 82] permet de trouver l'intersection $P \cap Q$, de deux polygones convexes P et Q , en temps $O(L+M)$ où L et M sont respectivement le nombre de sommets de P et de Q . L'intérêt de cet algorithme est qu'il peut donner deux informations : une sur les points d'intersections (positive et négative) et une autre sur les faucilles. Soit $I_{p1}, I_{p2}, \dots, I_{pk}$, k intersections positives, $I_{n1}, I_{n2}, \dots, I_{nk}$, k intersections négatives correspondantes. Nous notons par F_{pi} la faucille associée à I_{pi} dans P et par F_{si} celle dans $\text{CONV}(S)$ (voir figure 6.11)

A chaque point d'intersection positif, nous pouvons former facilement deux polygones convexes dont les sommets forment une orientation CW, notés par L_i et R_i où $L_i = \text{CH}(F_{si})$ et $R_i = \text{CH}(F_{pi} \cup F_{pi+1})$. Nous prenons deux chaînes extérieures, celle de F_{pi} et celle de F_{pi+1} , complétées des segments (I_{pi}, I_{ni}) et (I_{pi+1}, I_{ni-1}) pour former L_i . Par contre, R_i est formé par la chaîne extérieure de F_{si} complétée du segment (I_{pi}, I_{ni}) . Soit S_i le sous-ensemble de polygones situés dans la faucille F_{si} . Le lemme suivant montre que chaque F_{si} ($i=1, \dots, k$) va correspondre à une branche de courbe ouverte pour le séparateur entre S et P .

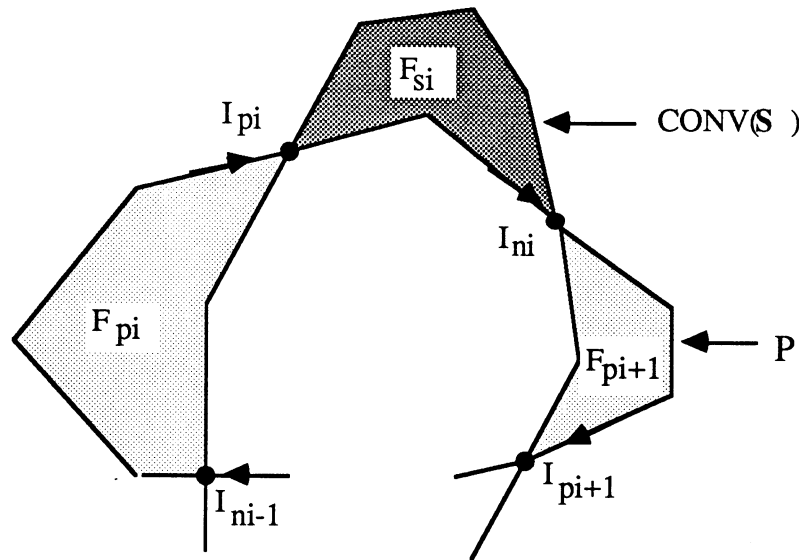


Figure 6.11 Chaque point d'intersection de P avec $CONV(S)$ est partagé par deux faucilles; l'une des faucilles est dans P et l'autre est dans $CONV(S)$

Lemme 6.11 F_{S_i} ($i=1, \dots, k$) correspond à une seule branche de courbe connexe et ouverte du séparateur entre S et P , notée (cette branche) par $Sep(S_i, P)$ où S_i est le sous-ensemble d'objets situés dans F_{S_i} . De plus la bissectrice de départ et celle d'arrivée de $Sep(S_i, P)$ sont déterminées par l'enveloppe convexe de L_i et R_i .

Preuve. Comme P et $CH(F_{S_i})$ ne s'intersectent qu'aux deux points, d'après les lemmes 5.1 et 5.2, le séparateur $Sep(S_i, P)$, entre S_i inclus dans $CH(F_{S_i})$ et P , est ouvert et composé d'une seule branche de courbe connexe. Il est évident que les deux lignes d'appui communes entre $L_i = CH(F_{S_i})$, et $R_i = CH(F_{p_i} \cup F_{p_{i+1}})$, sont identiques à celles entre S_i et le polygone P . De plus, d'après le lemme 5.6, on sait que la bissectrice de départ et celle d'arrivée de $Sep(S_i, P)$ sont déterminées par l'enveloppe convexe de L_i et R_i .

Corollaire 6.9 Soit S un ensemble de t polygones existant et P un polygone convexe à insérer composé de m sommets. Il y a au plus $K = \min \{t, m\}$ branches de courbes pour le séparateur entre S et P quand P est à insérer dans S .

Preuve. On doit considérer les quatre cas suivants :

- P est à l'intérieur de $CONV(S)$ ou le contraire ;
- $P \cap \zeta CONV(S) = \emptyset$, c'est-à-dire, P est complètement situé à l'extérieur de $CONV(S)$;
- P est coupé par $CONV(S)$ en deux parties dont l'une est à l'intérieur de $CONV(S)$ et l'autre est à l'extérieur de $CONV(S)$;

(d). P est coupé par $CONV(S)$ en $K+1$ ($K>1$) parties dont l'une est à l'intérieur de $CONV(S)$ et les K parties restantes sont à l'extérieur de $CONV(S)$. C'est le cas général.

Pour les cas (a), (b) et (c), on sait d'après le lemme 5.1 et le lemme 5.2 que le séparateur entre S et P est composé d'une seule branche de courbe connexe. Pour le cas (d), selon le lemme 6.11, il existe une seule branche ouverte de courbe connexe correspondant à chaque faucille dans l'ensemble d'objets existant. De plus, le nombre de faucilles dans S est délimité par le minimum de t et de m . Ceci achève la démonstration du lemme.

Maintenant nous pouvons donner un algorithme pour calculer le diagramme de Voronoï généralisé de $S \cup P$, $VOD(S \cup P)$, quand un polygone convexe P est inséré aléatoirement dans un ensemble d'objets existant dont le diagramme de Voronoï généralisé $VOD(S)$ est disponible. L'algorithme consiste d'abord à déterminer les faucilles, dans $CONV(S)$ et celles dans P correspondant à l'intersection de la frontière de l'enveloppe convexe de S et de la frontière de P . Ensuite pour chaque faucille F_{si} ($i = 1, \dots, k$) dans $CONV(S)$ calculer une branche de courbe connexe pour le séparateur $Sep(S_i, P)$. Puis chaque branche de courbe connexe du séparateur peut être calculée en temps linéaire en utilisant la méthode proposée au chapitre 5.

Algorithme RANDOM_CONVEX_POLYGON_INSERTION ;

entrée : $VOD(S)$ — le diagramme de Voronoï d'un ensemble d'objets S ,

P —le polygone convexe à insérer ;

sortie: $VOD(S \cup P)$;

Début

1. Initialisation;

(a). construire $VOD(P)$;

(b). trouver $CONV(S)$;

(c). trouver entre $CONV(S)$ et P , les intersections positives, $I_{p1}, I_{p2}, \dots, I_{pk}$, et les intersections négatives, $I_{n1}, I_{n2}, \dots, I_{nk}$; former les faucilles correspondantes dans P , $F_{p1}, F_{p2}, \dots, F_{pk}$, et celles dans $CONV(S)$, $F_{s1}, F_{s2}, \dots, F_{sk}$;

2. Si $k=0$ alors (P est inclus dans $CONV(S)$) construire une seule branche de courbe pour le séparateur fermé $Sep(S, P)$;

3. sinon pour $i=1$ à k faire

début

(a). $R_i \leftarrow F_{si} \cup (I_{ni}, I_{pi})$; $L_i \leftarrow \{F_{pi} \cup (I_{pi}, I_{ni-1})\} \cup \{F_{pi+1} \cup (I_{pi+1}, I_{ni})\}$; (voir la figure 6.11)

(b). A l'aide des quatre points d'appui dont deux sont dans R_i et autres dans L_i déterminer la bissectrice de départ et la bissectrice d'arrivée de $Sep(S_i, P)$ où S_i est le sous-ensemble d'objets contenu dans la faucille F_{si} . Puis calculer la branche de courbe connexe du séparateur $Sep(S_i, P)$;

fin;

4. mettre à jour toutes les régions de Voronoï dans $VOD(S)$ et $VOD(P)$ par lesquelles passe $Sep(S_i, P)$;

Fin ;

Dans l'algorithme précédent, les étapes 1, 2 et 4 peuvent être faites en temps linéaire. Donc le temps d'opération de l'étape 3 détermine la complexité de l'algorithme. Le lemme suivant va nous aider à estimer le temps de calcul de l'étape 3.

Lemme 6.12 Soit S un ensemble d'objets polygonaux existants, P un polygone convexe à insérer. Si S est divisé par P en K sous-ensembles d'objets, S_i ($i=1, 2, \dots, k$), compris dans une faucille F_{si} , le séparateur $Sep(S_i, P)$ ($i=1, 2, \dots, k$) n'intersecte alors aucune région de Voronoï associée à l'ensemble $S-S_i$.

Preuve. Ce que le lemme montre est :

$$Sep(S_i, P) \cap \bigcup_{e_j \in S-S_i} V(e_j, S-e_j) = \emptyset \quad (6.25)$$

Soit v_{i1} et v_{i2} deux extrémités des deux lignes d'appui communes entre $CH(S_i)$ et P qui sont dans la frontière de P (voir figure 6.12). Notons par C_i la chaîne polygonale de P débutant au sommet v_{i1} et se terminant au sommet v_{i2} en sens CW. De même, une chaîne polygonale de P associée à S_j est notée par C_j . Soit z un point appartenant à $Sep(S_i, P)$. Alors, on a :

$$d(z, C_i) < d(z, C_j) \quad i \neq j \quad (6.26)$$

Par ailleurs S_j est situé à l'extérieur de la chaîne C_j , c'est-à-dire :

$$d(z, C_j) < d(z, S_j) \quad (6.27)$$

donc,

$$d(z, C_i) < d(z, S_j) \quad i \neq j \quad (6.28)$$

Maintenant supposons que $Sep(S_i, P) \cap V(e_j, S-e_j) \neq \emptyset$ ($e_j \in S-S_i$). Soit q un point appartenant à cette intersection. On a alors :

$$d(q, S_i) = d(q, P) = d(q, C_i) \quad (6.29)$$

et

$$d(q, S_i) > d(q, e_j) = d(q, S_j) \quad i \neq j \quad (6.30)$$

D'après (6.29) et (6.30) on a :

$$d(q, C_i) > d(q, S_j) \quad i \neq j \tag{6.31}$$

Ceci est en contradiction avec (6.28). Donc, $Sep(S_i, P) \cap V(e_j, S - e_j) = \emptyset \ (e_j \in S - S_i)$.

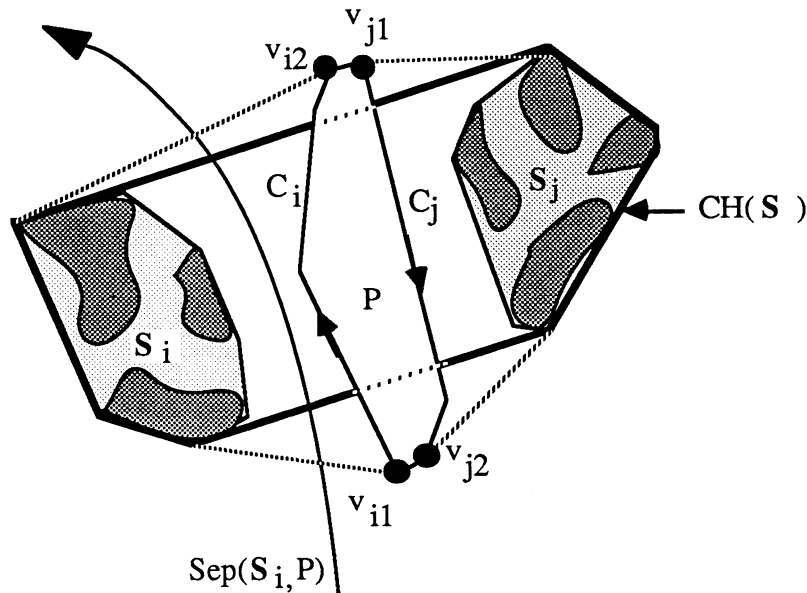


Figure 6.12 Illustration du lemme 6.12

Il est évident que le lemme 6.8 est inclus dans le lemme 6.12.

En conclusion nous avons :

Lemme 6.13 Etant donné le diagramme de Voronoï d'un ensemble d'objets $VOD(S)$, et un polygone convexe P à insérer, l'algorithme `RANDOM_CONVEX_POLYGON_INSERTION` est en $O(N)$ dans le cas le plus défavorable où N est le nombre d'éléments de $S \cup P$.

Preuve. Pendant la construction de $Sep(S, P)$, d'après le lemme 6.12, il n'y a aucune région de Voronoï dans $VOD(S)$ et $VOD(P)$ qui est utilisée plus d'une fois. L'étape 3 de l'algorithme prend donc un temps linéaire. Par conséquent, le calcul de tout l'algorithme prend un temps linéaire.

6.7 Exemples de diagrammes de Voronoï généralisés

Quelques sorties du programme d'après les algorithmes présentés précédemment pour des diagrammes de Voronoï généralisés sont donnés ci-dessous. La figure 6.13 montre des diagrammes de Voronoï généralisés d'un ensemble de segments. La figure 6.14 montre des diagrammes de Voronoï généralisés d'un ensemble de polygones convexes. Enfin, des diagrammes de Voronoï généralisés d'un ensemble de polygones quelconques sont illustrés en figure 6.15.

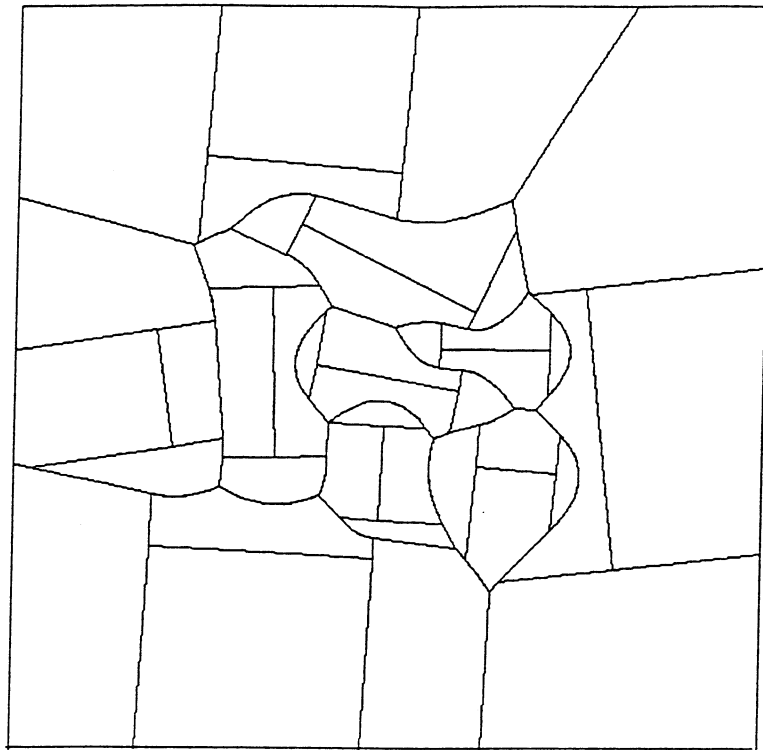


Figure 6.13 (a) Diagramme de Voronoï généralisé d'un ensemble de segments basé sur les éléments

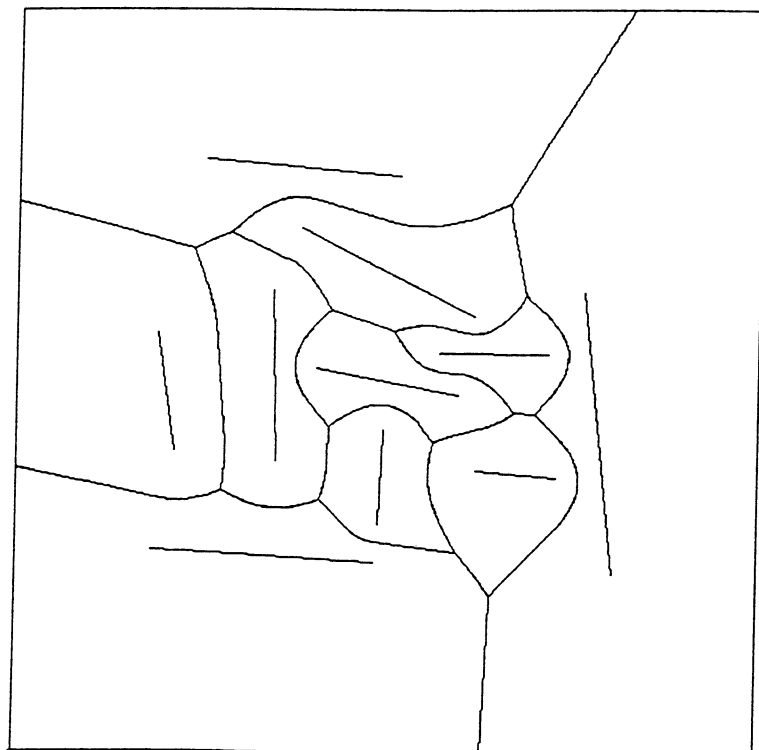


Figure 6.13 (b) Diagramme de Voronoï généralisé d'un ensemble de segments basé sur les objets

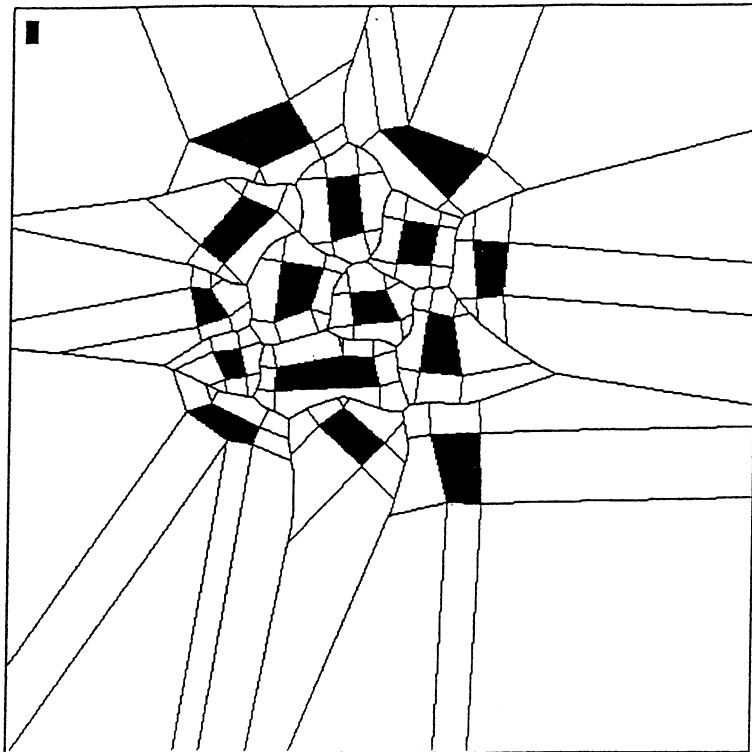


Figure 6.14 (a) Diagramme de Voronoï généralisé d'un ensemble de polygones convexes basé sur les éléments

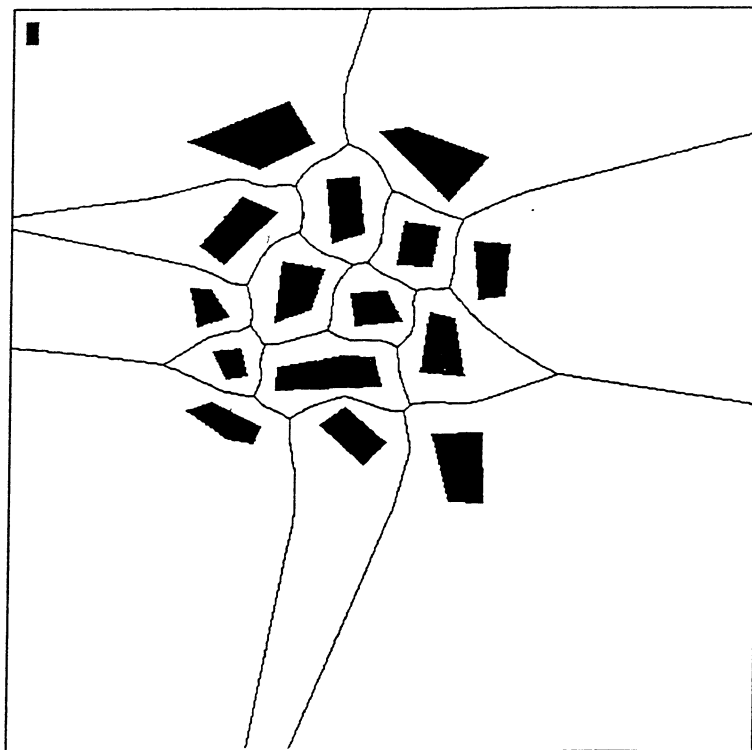


Figure 6.14 (b) Diagramme de Voronoï généralisé d'un ensemble de polygones convexes basé sur les objets

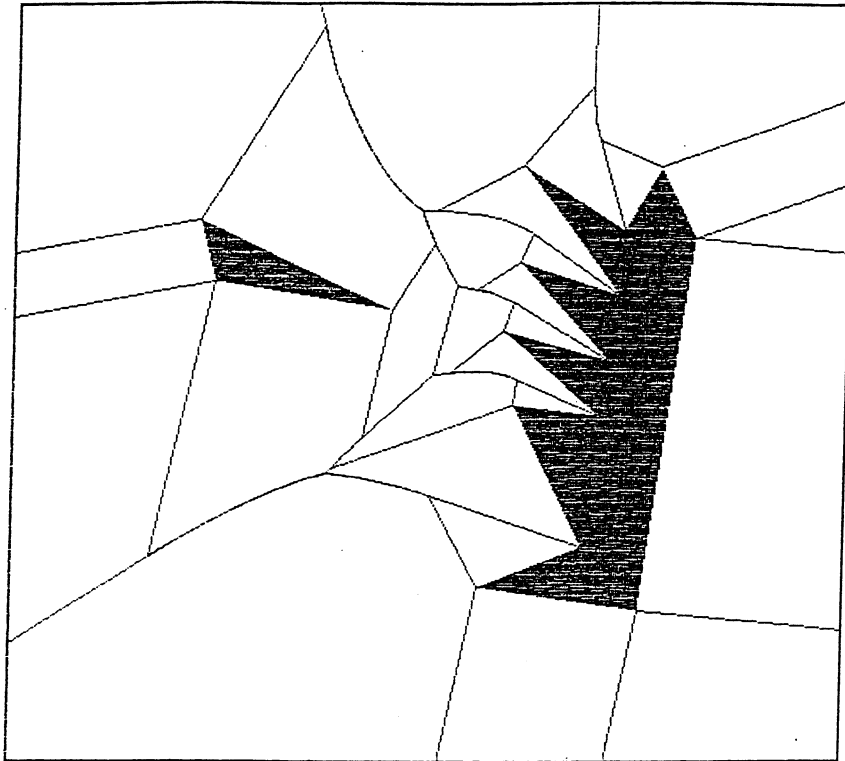


Figure 6.15 (a)

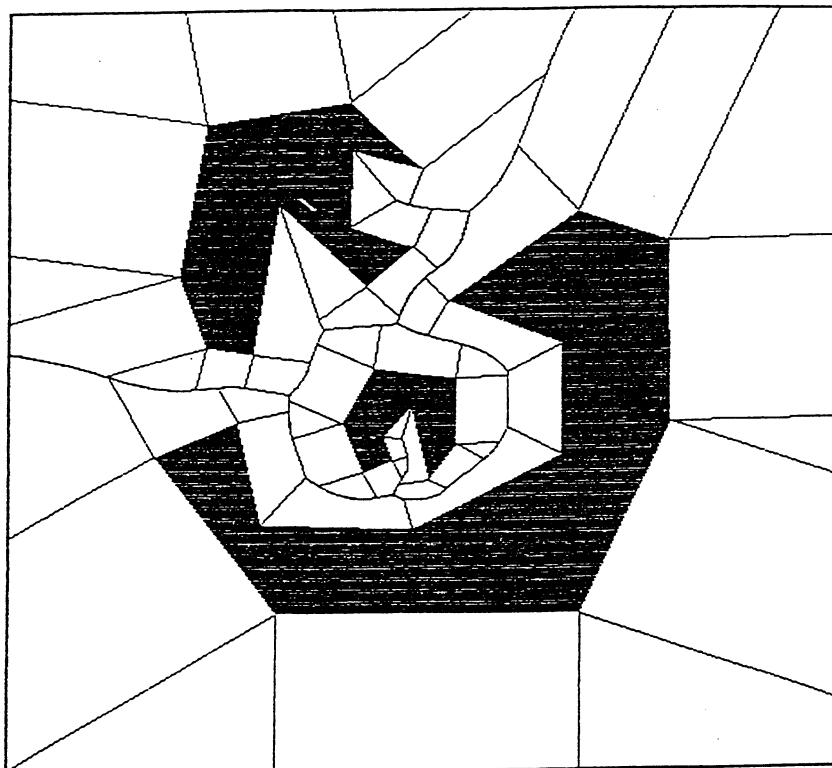


Figure 6.15 (b)

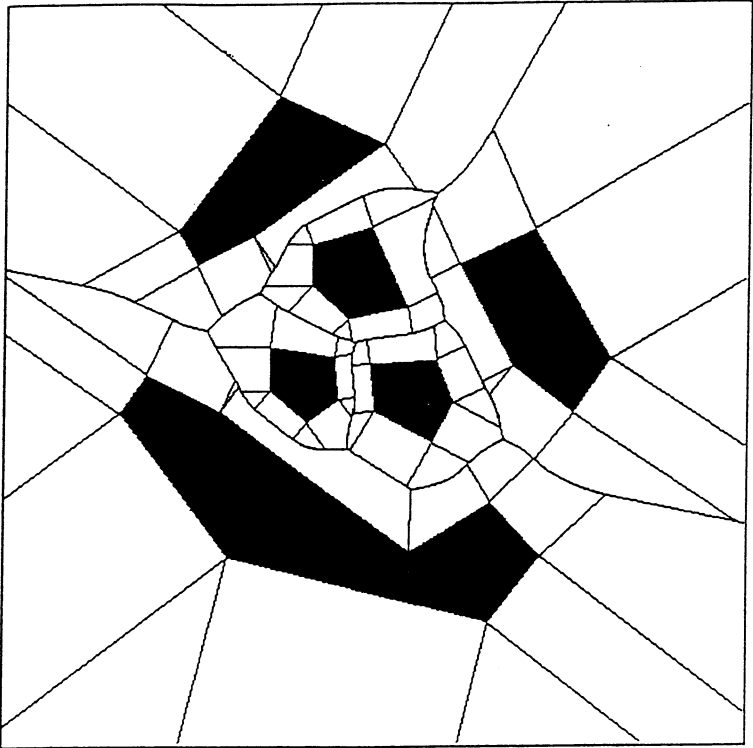


Figure 6.15 (c)

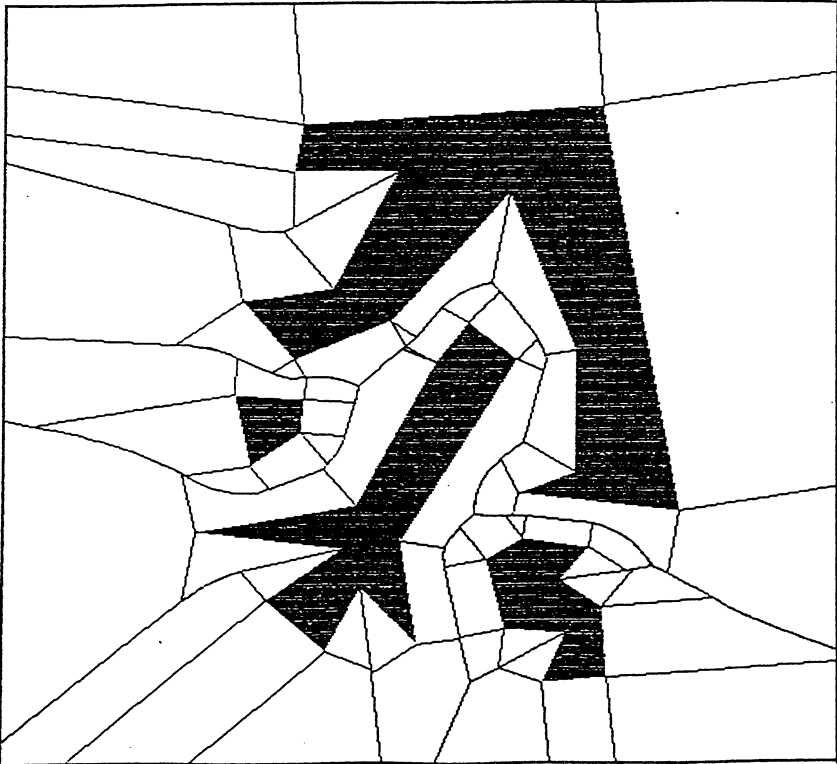


Figure 6.15 (d)

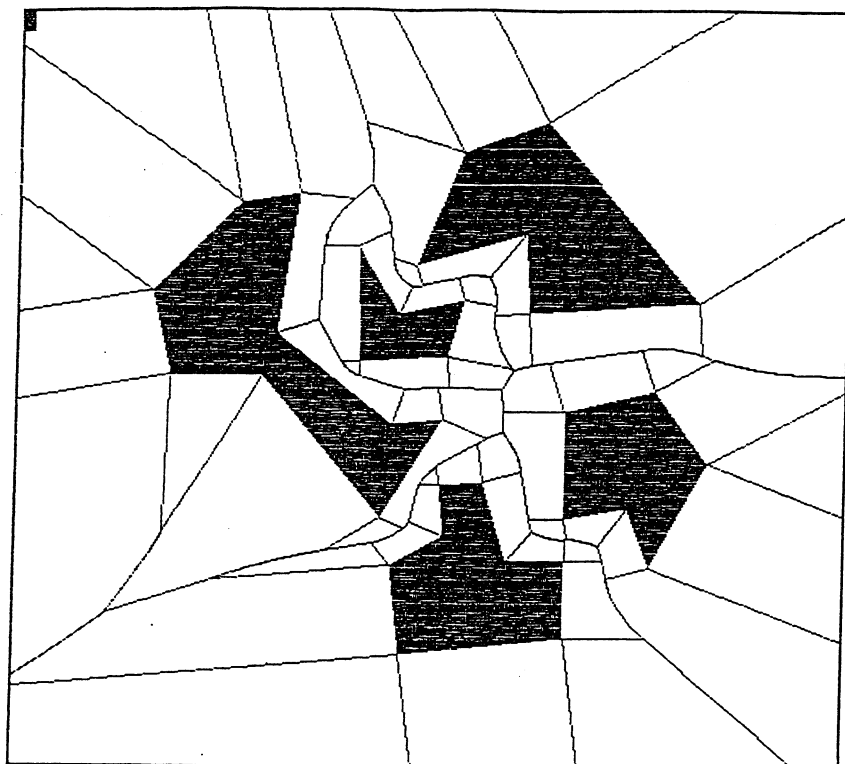


Figure 6.15 (e)

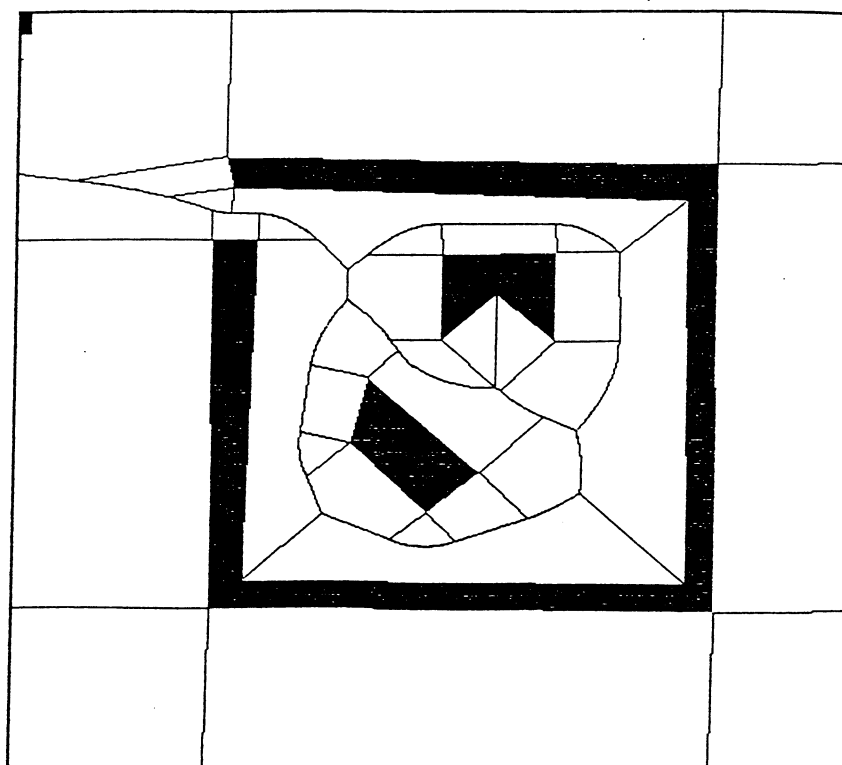


Figure 6.15 (f)

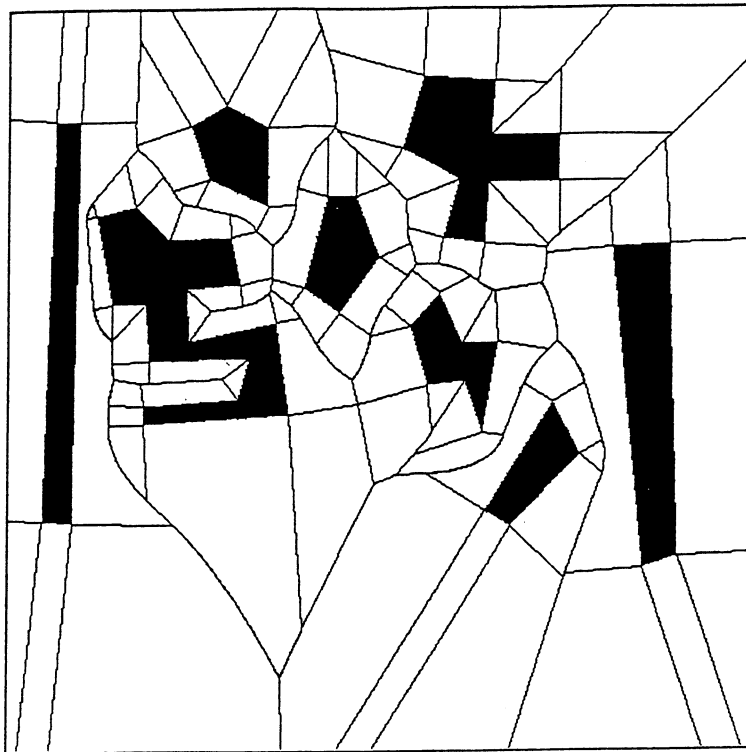


Figure 6.15 (g)

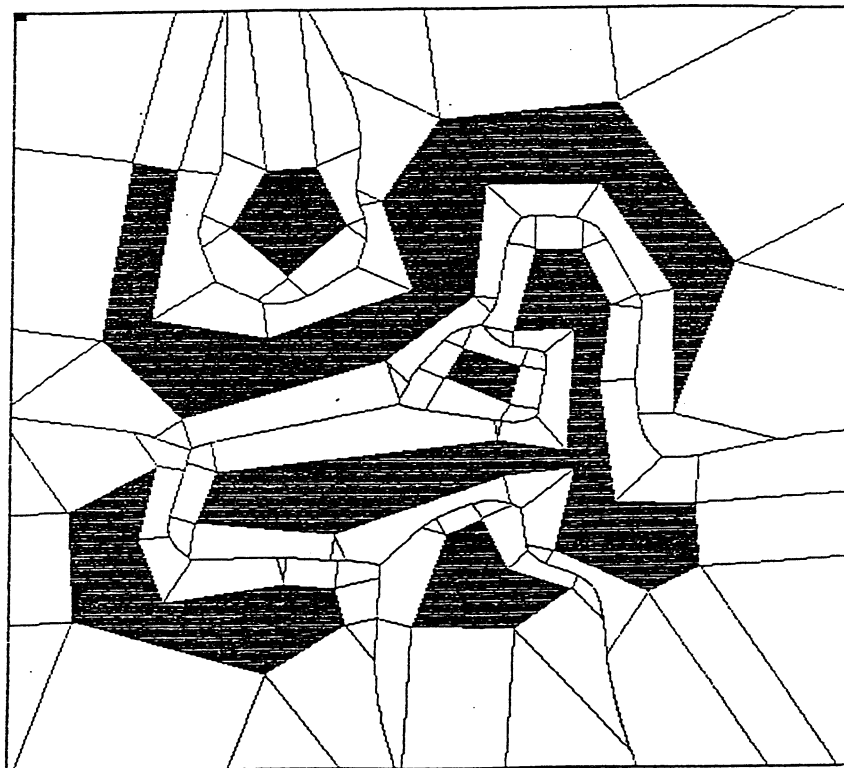


Figure 6.15 (h)

Figure 6.15 (a) - (h) Des diagrammes de Voronoï généralisés de polygones basé sur les éléments

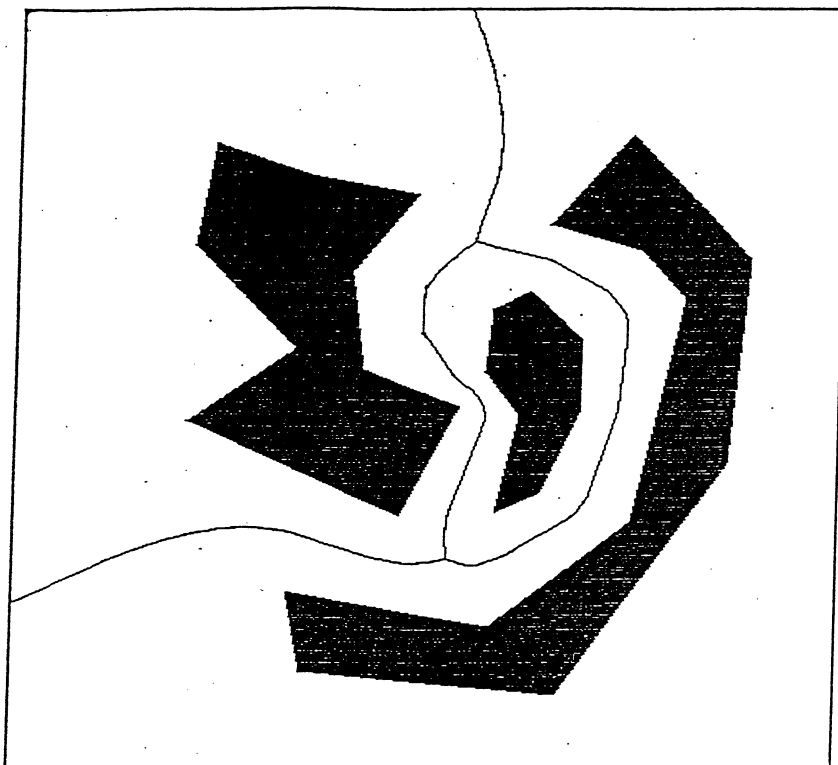


Figure 6.15 (i)

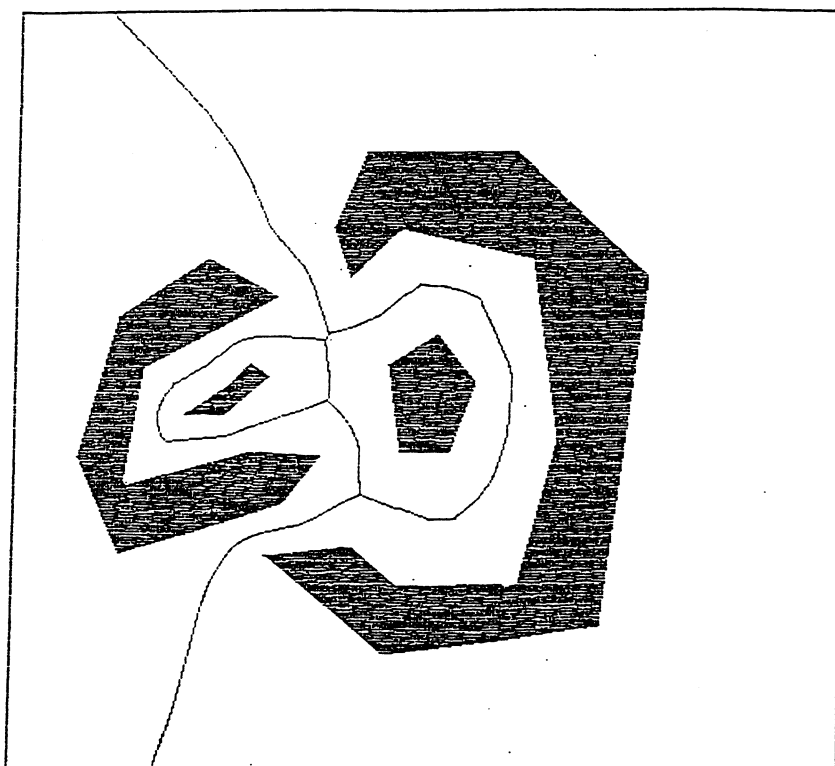


Figure 6.15 (j)

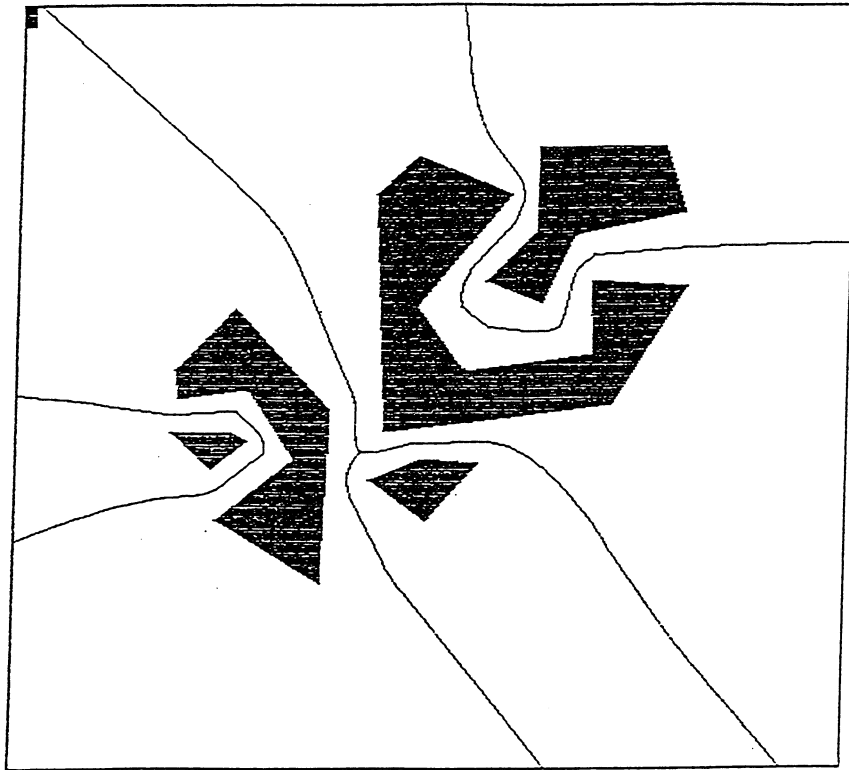


Figure 6.15 (k)

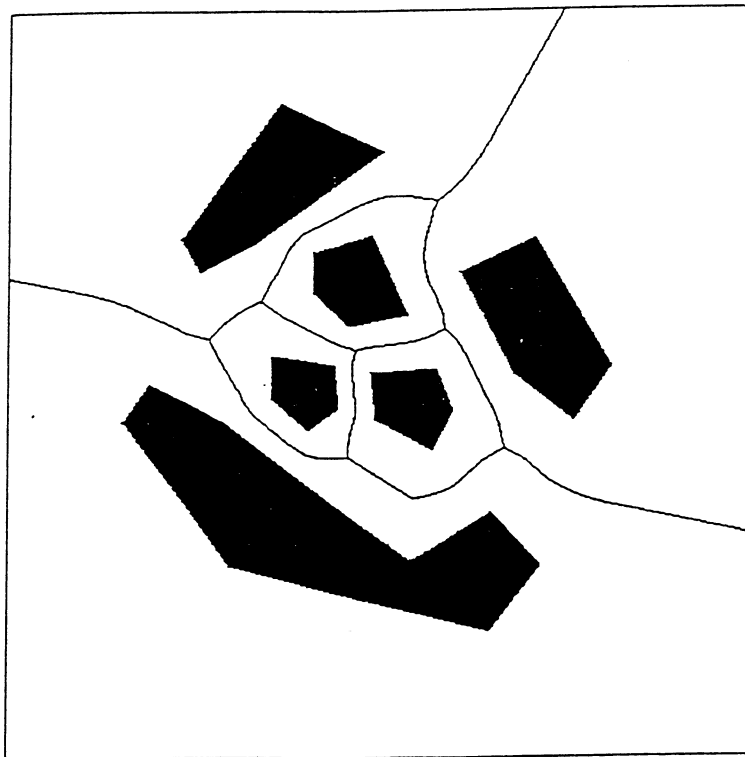


Figure 6.15 (l)

Figure 6.15 (i) - (l) Des diagrammes de Voronoï généralisés de polygones basé sur les objets

Chapitre 7

Axe médian, Exosquelette et SKIZ

7.1 Description de formes

L'analyse et la description de formes jouent un rôle important dans la plupart des systèmes de vision par ordinateur. Il n'est pas donc surprise que beaucoup de techniques nouvelles ont été introduites et appliquées aux problèmes de reconnaissance des formes. La majorité des techniques de l'analyse des formes peut être classée en deux catégories: les techniques fondées sur contour et les techniques fondées sur les régions. Comme exemple de la technique appartenant à la première catégorie on peut citer les descripteurs de Fourier uni-dimensionnel évalué sur la frontière. Les coefficients de la transformation de Fourier obtenus sont utilisés comme paramètres de formes pour la discrimination des formes [PERSON 77]. Une autre exemple de technique basée sur la frontière consiste à calculer un ensemble de points critiques en suivant la frontière. En principe, les points critiques correspondent aux points importants pour percevoir, par exemple, des discontinuités de courbure [PAVLIDIS 78] [ROSENFELD 73].

Par contre, les techniques fondées sur les régions représentent les formes à l'aide de leur structure *interne*. Deux approches principales de cette catégorie sont : (i) le *squelette* appelé aussi *transformation axe médian* [BLUM 64] et (ii) la décomposition d'une forme en *sous-ensembles convexes* [FENG 75] [SCHACHTER 78]. On peut remarquer que peu de méthodes existent qui utilisent à la fois l'information "interne" et "externe" pour analyser les formes.

7.2 Définitions existantes de squelette

Le squelette d'un objet est un concept intuitif. En traitement d'images, en vision par ordinateur et en reconnaissance de formes, la squelettisation d'un objet est une procédure cruciale. La représentation du squelette d'un objet peut non seulement réduire la place mémoire, mais peut permettre la mise en évidence de propriétés topologiques de l'objet. Il fournit une représentation centrée de la forme d'un objet. L'étude du concept de squelette a été introduite pour la première fois en 1964 par Blum [BLUM 64]. L'implémentation algorithmique et une théorie de squelette discret ont été ensuite développées par Rosenfeld *et al*, [ROSENFELD 66]

[PFLATZ 67], Montanari [MONTANARI 68] [MONTANARI 69]. Pendant ces deux dernières décennies, de grands efforts ont été consacrés à la squelettisation. Notamment, il existe plusieurs algorithmes d'amincissement travaillant dans le plan discret pour obtenir un squelette discret, mais comme Davies et Plummer [DAVIES 81] l'ont indiqué, la plupart des auteurs des algorithmes ne définissent pas un squelette, ne donnent pas non plus une exactitude pour le squelette obtenu.

En fait, les squelettes peuvent être définis de diverses façons mais équivalentes dans le plan continu ainsi que dans le plan discret. Nous ne voulons pas donner une définition générale du squelette. Par contre, nous préférons présenter différentes définitions. En effet, les différentes définitions vont apporter différents schémas de calcul de squelette. Citons ici trois définitions différentes mais équivalentes :

(1) **Simulation de la propagation d'une onde** [BLUM 64]. Soit w une onde. Le contour d'un objet est considéré comme source d'un front de l'onde. C'est-à-dire que l'onde est positionnée sur la frontière de l'objet quand la variable temporaire est égale à 0. Puis nous la laissons se propager à vitesse constante avec la contrainte qu'aucun point du contour ne peut être excité plus d'une fois (tous sont excités au même instant). Il existe des points où les fronts de l'onde provenant de parties différentes de la frontière se rencontrent. Les points de rencontre et un ensemble de la mesure du temps correspondant représentent le squelette (cf. figure 7.1).

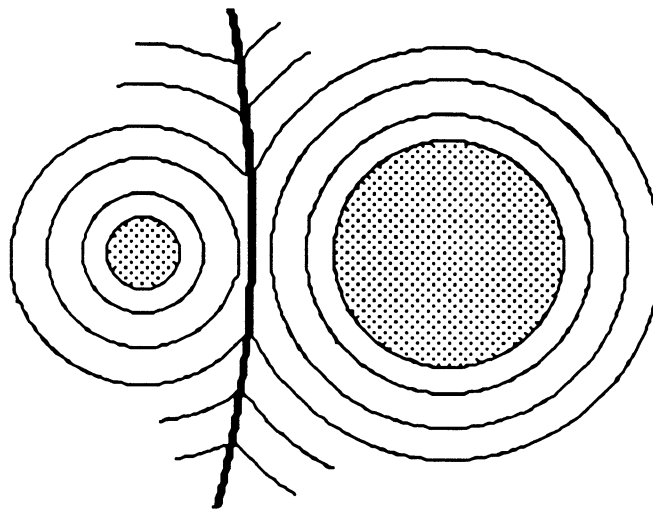


Figure 7.1 Squelette fondé sur la simulation de la propagation d'une onde

(2) **Centres de boules maximales** [BLUM 78] [ROSENFELD 66] [SERRA 82]. Soit $B(x, r)$ une boule centrée en x et du rayon r . Une boule $B(x, r)$ incluse dans l'objet X est dite maximale si et seulement si il n'existe aucune autre boule de X qui la contient. Le squelette est alors défini comme l'ensemble des centres des boules maximales:

$$S(X) = \bigcup \{x \in X, \exists r \geq 0 \text{ tel que } B(x, r) \text{ est maximale dans } X\}$$

Et l'objet peut alors se reconstruire par la réunion de toutes les boules maximales (cf. la figure 7.2).

Particulièrement, on a un *squelette intérieur* (respectivement *extérieur*) si on limite les boules maximales à l'intérieur (respectivement extérieur) de l'objet X . Le squelette extérieur contient l'exosquelette et le SKIZ [SERRA 82] (SKeleton by Zones of Influence), notions définies ultérieurement.

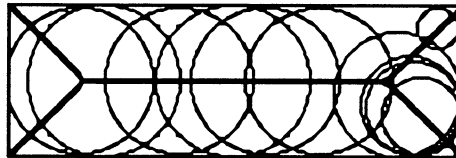


Figure 7.2. Squelette par boules maximales

(3) **Basée sur la transformation de distance.** Pour chaque point p de l'objet X , on définit une fonction :

$$L(p) = \min (d(p, X^c))$$

où X^c désigne le complémentaire de X . Cette fonction transforme l'espace de la représentation de l'objet en espace de distance. On appelle un tel processus *la transformation de distance* [ROSENFELD 68] [DANIELSSON 80] [BORGEFORS 86]. Une fois que l'espace de distance est construit, le squelette peut alors être défini comme l'ensemble des points auxquels $L(p)$ a des *maxima locaux*.

Bien que la définition du squelette soit donnée dans l'espace continu, elle peut être aussi formulée dans l'espace discret. En fait, la plupart des algorithmes de squelettisation sont développés dans le plan discret puisqu'une forme est toujours contenue dans une image numérique. Par conséquent, la propagation d'une onde sera simulée dans le plan discret ; la boule continue devient une boule numérique, et la distance utilisée dans la transformation de distance n'est pas toujours sous la métrique euclidienne réelle.

Ces trois définitions sont en fait équivalentes. Selon les définitions, étant donné le squelette d'un objet X et la valeur de distance associée à chaque point du squelette (rayon de la boule correspondante), on peut reconstruire l'objet X . Donc, la squelettisation est réversible.

Remarquons aussi que d'après les définitions, un squelette est assez mince pour que sa largeur soit ramenée à un seul point. De plus, les points du squelette sont généralement classés en trois catégories :

a. *Les points normaux*: la boule correspondante touche la frontière de l'objet en exactement

deux points distincts ;

b. *Les points de branchement* ou *points multiples*: la boule maximale correspondante touche la frontière de l'objet en trois points distincts ou plus ;

c. *Les points terminaux*. (cf. figure 7.3)

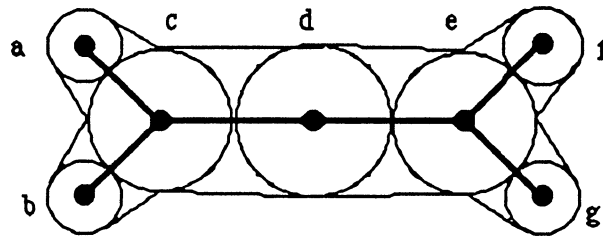


Figure 7.3. Trois catégories des points du squelette
d: point normal; *c, e*: points de branche; *a, b, f, g*: points terminaux

Quant au modèle de calcul du squelette, il existe de nombreux algorithmes fondés sur les trois définitions précédentes. En fait, les différentes définitions génèrent des méthodes différentes pour calculer les squelettes. Les méthodes les plus connues et les plus développées sont les suivantes : simulation de la propagation d'une onde, amincissement d'un objet et transformation axe médian. Par ailleurs, il existe une autre catégorie de méthodes de calcul : *méthodes basées sur la géométrie algorithmique*. Cette dernière catégorie de méthodes est peu étudiée par rapport aux trois premières. Nous la laissons en discussion dans les paragraphes suivants. Nous présentons d'abord les trois premières méthodes.

Approches par simulation de la propagation d'une onde

Intuitivement, cette catégorie d'approches est venue de la définition d'un squelette fondé sur un processus de propagation d'une onde, ou alternativement, la propagation de feu, puisque le processus de propagation du feu dans une prairie d'herbes sèches est similaire à celui d'une onde. Etant donné un objet binaire, la technique nous permet de travailler sur la frontière de l'objet. En supposant que le front d'une onde est connu au moment t , pour simuler la propagation de l'onde, il faut déterminer le nouveau front de l'onde au moment $t + \Delta t$. En traçant le front de l'onde courante, on marque simultanément les points appartenant au front de l'onde à venir. Les points de contour qui sont marqués en deux fois ou plus, sont alors les points brûlés. Ce sont ces points brûlés qui constituent le squelette de l'objet [XIA 89].

Approches par amincissement d'un objet

L'opération d'amincissement consiste à peler successivement le contour d'un objet discret jusqu'à arriver à un résultat d'épaisseur unité [PAVLIDIS 80]. L'amincissement se fait à condition de ne pas perdre la propriété de connexité [CHASSERY 79], c'est-à-dire, l'amincissement élimine itérativement des points du contour si l'élimination n'affecte pas la connexité de l'objet. L'examen d'un voisin local est suffisant pour vérifier si un point peut être supprimé [PFLATZ 67]. Il existe beaucoup de variations basées sur le même principe en utilisant des masques différents [CHASSERY 91].

Approches par transformation de distance

La transformation d'axe médian dans le plan discret a été proposée pour la première fois par Rosenfeld *et al* [ROSENFELD 68]. A partir de l'espace d'objet, la transformation de distance est d'abord effectuée. Les points correspondant aux maxima locaux représentent l'axe médian. Pour donner un "meilleur" squelette, il faut effectuer une transformation de distance qui donne une meilleure approximation de la distance euclidienne réelle [DANIELSSON 80] [BORGEFORS 86].

Pour réaliser cela, le calcul des transformations en distance discrète est utilisé. Les distances globales dans une image sont obtenues par propagation de distance locale qui est la distance discrète entre des pixels voisins. La propagation peut s'effectuer de manière séquentielle ou parallèle [ROSENFELD 66]. Différentes distances locales ont été proposées et comparées pour la transformations en distance. Elles sont: **City block** (d4); **Chessbord** (d8); **Octagonal**; **Chamfer 3-4**; **Chamfer 5-7-11**; et **Euclidienne** [BORGEFORS 86]. En raison de la discretisation, les axes médians obtenus à partir de la transformation de distance sont souvent déconnectés. Pour connecter les composantes de l'axe médian, une ligne médiane [MONTANVERT 87] peut être construite en rajoutant les points qui ont détruit la connexion.

7.3 Approximation de squelettes continus

Dans ce paragraphe, nous présentons l'approximation de *squelette continu* d'objets. Un squelette est dit continu s'il est défini et calculé dans l'espace continu. Donc, le squelette continu est *exact*. Un squelette exact implique qu'il satisfait quelques propriétés inhérentes existant dans la définition originale continue de Blum [BLUM 64] : par exemple l'invariance en translation, rotation, changement d'échelle etc. Ici, l'approximation du squelette continu sera faite en deux sens : (a) le modèle de calcul est effectué dans le plan continu ; (b) certains points

du squelette obtenu sont identiques à ceux du squelette exact.

7.3.1 Squelettes par la triangulation de Delaunay

Soit $S = \{p_1, p_2, \dots, p_i, \dots, p_N\}$ un ensemble de points du plan. Une triangulation de S est un ensemble maximal de segments de droite entre des points de S sans intersection 2 à 2. C'est-à-dire, une triangulation est un graphe planaire sur S ayant le maximum d'arêtes [PREPARATA 88] [SCHMITT 90]. Le dual du diagramme de Voronoï, obtenu en joignant deux germes dont les régions de Voronoï sont adjacentes, est appelé *la triangulation de Delaunay*. La figure 7.4. montre un exemple du diagramme de Voronoï et de la triangulation de Delaunay correspondante. La triangulation de Delaunay vérifie les propriétés suivantes:

- (1) La triangulation de Delaunay est une partition du plan en régions de triangle. Les sommets de chaque triangle sont les germes de l'ensemble S .
- (2) Pour chaque triangle de la triangulation de Delaunay, il existe un *cercle de Delaunay* qui circonscrit le triangle. Un cercle de Delaunay ne contient aucun germe en son intérieur.
- (3) Pour un germe p_i , il existe un polygone étoilé avec noyau p_i tel qu'il ne contient pas d'autre germe.
- (4) L'union de tous les triangles de Delaunay est égale à l'intérieur de l'enveloppe convexe de S .

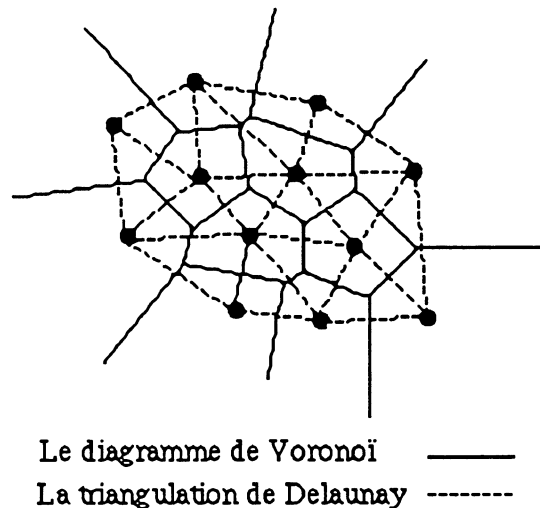


Figure 7. 4. Le diagramme de Voronoï d'un ensemble de points et la triangulation de Delaunay correspondante

Soit P un polygone. Quelquefois, nous avons le problème de *la triangulation contrainte*. Un exemple est la triangulation de l'intérieur (ou l'extérieur) du polygone P à partir de l'ensemble

des sommets du polygone P . Le problème est souvent appelé *la triangulation de polygone*.

Un autre exemple consiste en construction de la triangulation de Delaunay d'un ensemble de points répartis sur la frontière d'un polygone P [BOISSONNAT 84]. Par exemple, la discrétisation des arêtes d'un polygone fournit un tel ensemble. Pour réaliser un tel échantillonnage de la frontière d'un polygone, on utilise une mesure *densité de points*. Formellement, on appelle ensemble de points de densité $1/\varepsilon$ tout ensemble $S = \{p_1, p_2, \dots, p_i, \dots, p_N\}$ de points de ∂P tel que toute boule fermée $B(p, \varepsilon/2)$ centrée en un point de $p \in \partial P$ et de rayon $\varepsilon/2$ contienne au moins un point de S [LeBRAS 89]. Cette définition montre comment on peut échantillonner la frontière du polygone P aussi finement que l'on veut.

Soit S un ensemble de points échantillonnés sur la frontière d'un objet P , la densité de S , étant ε_S . Notons C_S l'ensemble des centres des cercles de Delaunay correspondant à la triangulation de S . Quand S est de plus en plus dense sur ∂P , c'est-à-dire, $\varepsilon_S \rightarrow +\infty$, alors C_S va former le squelette de P [FAUGERAS 90]. En particulier, si les triangles de Delaunay sont contraints dans l'intérieur de P , le squelette obtenu est l'axe médian. Similairement, les centres des cercles de Delaunay situés à l'extérieur donnent le squelette extérieur. Si la densité de points sur ∂P n'est pas infinie, une approximation du squelette peut être obtenue en reliant les centres des cercles de Delaunay voisins.

En résumé, un simple algorithme pour calculer une approximation du squelette à l'aide de la triangulation de Delaunay peut se formuler par :

Algorithme SQUELETTE_PAR_DELAUNAY ;

entrée : la représentation polygonale ou de la courbe de la frontière d'un objet P ;

sortie : approximation du squelette de P , $SQT(P)$;

Début

1. échantillonner ∂P en un ensemble de points S ;
2. construire la triangulation de Delaunay de S ;
3. pour chaque triangle de Delaunay, calculer le centre de cercle de Delaunay;
4. lier les centres des cercles de Delaunay voisins pour obtenir $SQT(P)$;

Fin;

7.3.2 Squelettes par diagramme de Voronoï classique

Comme le centre du cercle de Delaunay est équidistant des trois sommets du triangle de Delaunay associé, selon les corollaire 2.2 et 2.3 et la définition du diagramme de Voronoï classique, il est très facile de démontrer que le centre du cercle de Delaunay est précisément un sommet de Voronoï. On peut donc utiliser le diagramme de Voronoï classique pour faire une

approximation de squelette continu. En fait, la propriété indiquant que le squelette continu est formé de points situés à une distance égale à au moins deux points de la frontière est équivalente à indiquer que deux points de la frontière peuvent produire un point du squelette.

Par conséquent, si on échantillonne la frontière d'un objet en une suite d'ensemble de points, le graphe du diagramme de Voronoï classique de cet ensemble de points échantillonnés contient certains points du squelette [KUBLER 89]. De plus, les points du squelette sont compris dans les sommets de Voronoï qui sont les centres des cercles de Delaunay. Comme les sommets de Voronoï sont reliés entre eux par les arêtes de Voronoï, le calcul du squelette par diagramme de Voronoï classique consiste à enlever certaines arêtes de Voronoï inutiles dans le diagramme de Voronoï classique des points échantillonnés. Pour faire cela, un critère géométrique peut être utilisé portant sur le fait qu'une arête de Voronoï inutile a une intersection avec la frontière de l'objet [MELKEMI 90].

7.4 Squelettes continus d'après le diagramme de Voronoï généralisé

Dans ce qui suit, nous allons présenter des approches de calcul des squelettes exacts continus d'après le diagramme de Voronoï généralisé. Nous parlerons de trois types des squelettes: *axe médian*, *exosquelette* et *SKIZ*. Il sera montré que les squelettes continus d'objets polygonaux sont en fait des dérivées inhérentes au diagramme de Voronoï généralisé des objets.

7.4.1 Axe médian

Nous attirons le lecteur au fait que le terme d'axe médian est examiné au sens analogique et non discret. Il s'associe au terme de squelette pour l'intérieur d'un objet et du fait de la propriété de positionnement central au sein de la forme. Cette dénomination d'axe médian est plus significative que le simple terme de squelette.

Revenons aux définitions du squelette. Pour chaque point du squelette d'une forme (objet), il existe un cercle centré en ce point qui est tangent en au moins deux points différents de la frontière de l'objet. Soit S un objet dont la frontière ∂S est représentée sous la forme d'un polygone. Le squelette à l'intérieur de S ou l'axe médian de S est composé de points tels que les distances du point de l'axe médian à différents points de ∂S sont égales. Pour formuler tout cela, nous avons:

Définition 7.1 Soit S un ensemble d'objets qui sont représentés sous la forme de polygones. L'axe médian des objets S est défini par:

$$AM(S) = \{q \in S \mid \exists p_i, p_j \in \partial S \ (p_i \neq p_j) \text{ tels que } d(q, p_i) = d(q, p_j) = d(q, \partial S)\} \quad (7.1)$$

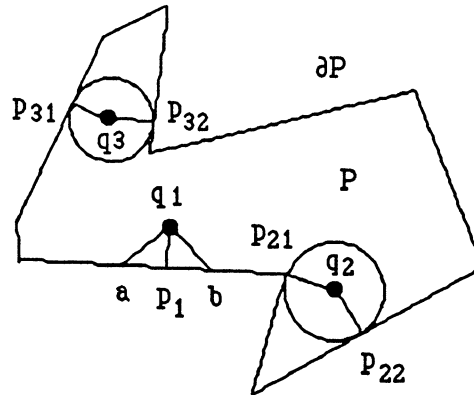


Figure 7.5 un point de l'axe médian est déterminé par trois quantités égales de distance

Cette définition est bien équivalente aux trois définitions du squelette citées précédemment, elle utilise un autre critère. En fait, l'égalité des trois distances $d(q, p_i)$, $d(q, p_j)$ et $d(q, \partial S)$ indique que les points p_i et p_j sont deux points parmi les plus proches de q sur ∂S . Ceci nous permet d'établir facilement un rapport entre l'axe médian et le diagramme de Voronoï. Remarquons que la distance entre un point et un objet est définie comme celle entre le point et l'image du point sur l'objet. Dans la figure 7.5, q_1 n'est pas un point de l'axe médian puisqu'on ne peut pas trouver deux points différents de la frontière du polygone ayant la même distance que celle entre q_1 et la frontière du polygone. Par exemple, bien que $d(q_1, a) = d(q_1, b)$, elles ne sont pas égales à $d(q_1, \partial P) = d(q_1, p_1)$. Par contre, q_2 et q_3 sont des points de l'axe médian car $d(q_2, p_{21}) = d(q_2, p_{22}) = d(q_2, \partial P)$ et $p_{21} \neq p_{22}$; et $d(q_3, p_{31}) = d(q_3, p_{32}) = d(q_3, \partial P)$ et $p_{31} \neq p_{32}$. En effet, on a que la boule centrée en q_2 (resp. q_3) de rayon $r = d(q_2, p_{21}) = d(q_2, p_{22})$ (resp. $r = d(q_3, p_{31})$) est justement la boule maximale. On ne peut pas trouver une boule similaire à q_1 .

Intuitivement, l'axe médian d'un objet est compris dans le diagramme de Voronoï généralisé de l'intérieur de l'objet. En effet les éléments du diagramme de Voronoï généralisé basé sur éléments sont composés des points de la frontière de la forme polygonale ∂S utilisés dans (7.1). Jusqu'à présent, on a défini toujours le diagramme de Voronoï comme la réunion de toutes les régions individuelles de Voronoï. En fait, la représentation du diagramme de Voronoï est basée sur une structure de graphe qui comprend des nœuds qui sont les sommets de Voronoï et des branches qui sont les arêtes de Voronoï. De plus, nous savons que (a) pour un sommet de Voronoï il existe un cercle centré au sommet passant par au moins trois germes et ne contenant aucun autre germe (voir les corollaires 2.2 et 2.3); (b) pour un point sur une arête de Voronoï, il existe deux éléments différents qui sont équidistants du point (la définition originale). Donc, nous pouvons définir le diagramme de Voronoï en terme des points du graphe associé.

Etant donné un ensemble d'objets polygonaux S , la distance entre un point q et S est toujours définie comme (2.1) :

$$d(q, \partial S) = \min_{p \in \partial S} d(q, p) \tag{7.2}$$

Alors, le graphe du diagramme de Voronoï basé sur les éléments de l'intérieur d'un ensemble d'objets polygonaux S peut être défini par l'ensemble des points suivants:

$$VOD(S) = \{q \in S \mid d(q, e_i) = d(q, e_j) = d(q, \partial S), e_i \neq e_j\} \tag{7.3}$$

Nous appelons les points de $VOD(S)$ les *points de graphe*. Illustrons cette caractérisation en terme d'égalité au cas du diagramme de Voronoï classique. Soit S un ensemble de points p_1, p_2, \dots, p_7 illustré en figure 7.6. Selon la définition de $VOD(S)$, on peut trouver qu'un point du graphe est toujours contraint par trois valeurs de distance. En figure 7.6, q_1 est un point de graphe et il vérifie : $d(q_1, p_1) = d(q_1, p_4) = d(q_1, S)$. Cependant, q_2 n'est pas un point de graphe car $d(q_2, p_1) = d(q_2, p_4) \neq d(q_2, S) = d(q_2, p_3)$.

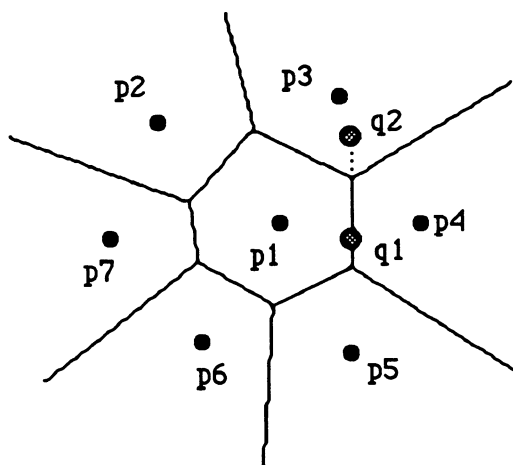


Figure 7.6 Un point sur le graphe du diagramme de Voronoï est contraint par trois quantités égales de distance

Dans le cas d'objets polygonaux, les germes sont les éléments issus de la décomposition de la frontière des polygones. Chaque élément e_i est formé d'un segment ouvert s_i ou d'un sommet concave p_i . Etant donné 2 éléments e_i et e_j alors le couple $e_i \times e_j$ appartient à l'ensemble $\{s_i s_j, s_i p_j, p_i s_j, p_i p_j\}$, on a alors :

$$VOD(S) = Q_{pp} \cup Q_{ss} \cup Q_{sp} \cup Q_{ps} \tag{7.4}$$

avec

$$Q_\alpha \cap Q_\beta = \emptyset \text{ si } \alpha \neq \beta$$

où

$$Q_{pp} = \{q \in S \mid d(q, p_i) = d(q, p_j) = d(q, \partial S), p_i \neq p_j\} \quad (7.5)$$

$$Q_{ss} = \{q \in S \mid d(q, s_i) = d(q, s_j) = d(q, \partial S), s_i \neq s_j\} \quad (7.6)$$

$$Q_{sp} = \{q \in S \mid d(q, s_i) = d(q, p_j) = d(q, \partial S), s_i \neq p_j\} \quad (7.7)$$

$$Q_{ps} = \{q \in S \mid d(q, p_i) = d(q, s_j) = d(q, \partial S), p_i \neq s_j\} \quad (7.8)$$

Car $s_i \neq s_j$, donc,

$$Q_{ss} = \{q \in S \mid d(q, p_i) = d(q, p_j) = d(q, \partial S), p_i \in s_i \neq p_j \in s_j\} \quad (7.9)$$

Remarquons que Q_{sp} peut être divisé en deux sous-ensembles de points selon les deux cas suivants : p_j est une extrémité de s_i ; et p_j n'est pas une extrémité de s_i . C'est-à-dire que :

$$\begin{aligned} Q_{sp} = & \{q \in S \mid d(q, s_{k1}) = d(q, p_{l1}) = d(q, \partial S); p_{l1} \text{ est extrémité de } s_{k1} \} \\ & \cup \{q \in S \mid d(q, s_{k2}) = d(q, p_{l2}) = d(q, \partial S); p_{l2} \text{ n'est pas d'extrémité de } s_{k2} \} \end{aligned} \quad (7.10)$$

De même :

$$\begin{aligned} Q_{ps} = & \{q \in S \mid d(q, p_{m1}) = d(q, s_{n1}) = d(q, \partial S); p_{m1} \text{ est extrémité de } s_{n1} \} \\ & \cup \{q \in S \mid d(q, p_{m2}) = d(q, s_{n2}) = d(q, \partial S); p_{m2} \text{ n'est pas extrémité de } s_{n2} \} \end{aligned} \quad (7.11)$$

En combinant respectivement les deuxièmes termes et les premiers termes de (7.10) et (7.11) on obtient :

$$\begin{aligned} Q_a = & \{q \in S \mid d(q, s_{k2}) = d(q, p_{l2}) = d(q, \partial S), p_{l2} \text{ n'est pas extrémité de } s_{k2}\} \\ & \cup \{q \in S \mid d(q, p_{m2}) = d(q, s_{n2}) = d(q, \partial S), p_{m2} \text{ n'est pas extrémité de } s_{n2}\} \\ = & \{q \in S \mid d(q, p_i) = d(q, p_j) = d(q, \partial S), p_i \neq p_j \in s_j \} \end{aligned} \quad (7.12)$$

$$\begin{aligned} Q_b = & \{q \in S \mid d(q, s_{k1}) = d(q, p_{l1}) = d(q, \partial S), p_{l1} \text{ est extrémité de } s_{k1}\} \\ & \cup \{q \in S \mid d(q, p_{m1}) = d(q, s_{n1}) = d(q, \partial S), p_{m1} \text{ est extrémité de } s_{n1}\} \\ = & \{q \in S \mid d(q, p_i) = d(q, s_j) = d(q, \partial S), p_i \text{ est extrémité de } s_j \} \end{aligned} \quad (7.13)$$

(7.4) devient alors :

$$VOD(S) = Q_{pp} \cup Q_{ss} \cup Q_a \cup Q_b \quad (7.14)$$

avec

$$Q_\alpha \cap Q_\beta = \emptyset \text{ si } \alpha \neq \beta.$$

Maintenant, nous déclarons un lemme important qui établit un rapport entre l'axe médian et le diagramme de Voronoï généralisé. Un lemme similaire a été donné par LEE [LEE 82a].

Lemme 7.1 L'axe médian d'un objet polygonal S , $AM(S)$, est un sous graphe du diagramme de Voronoï généralisé, $VOD(S)$, de l'intérieur de S basé sur les éléments. Les

éléments composent la frontière ∂S . Précisément, $AM(S)$ consiste en graphe issu de $VOD(S)$ excepté les arêtes connectées aux sommets concaves (mesurés depuis l'intérieur de S).

Preuve. Supposons que $q \in AM(S)$. Il est facile de démontrer $q \in VOD(S) = Q_{pp} \cup Q_{ss} \cup Q_a \cup Q_b$. Donc,

$$AM(S) \subseteq VOD(S). \quad (7.15)$$

D'autre part, on peut facilement prouver d'après (7.5) (7.9) et (7.12) que :

$$Q_{pp} \subseteq AM(S); \quad Q_{ss} \subseteq AM(S); \quad Q_a \subseteq AM(S) \quad (7.16)$$

Dans l'ensemble des éléments composant la frontière d'un objet polygonal S , un élément est de type "point" si et seulement si il est un sommet concave. Dans (7.13), le point p_i est une extrémité du segment s_j . Ceci implique que l'élément de type segment s_j est élément voisin du sommet concave p_i . Dans ce cas-ci, un point intérieur de Q_b , q , est situé sur une arête de Voronoï connectée à un sommet concave. Il est évident que q ne sera pas le centre d'une boule maximale puisque la boule centrée en q et de rayon $r = d(q, p_i)$ va certainement être incluse dans une autre boule (voir figure 7.7). Par conséquent, l'ouverture de Q_b , Q_b° , n'est pas une composante de l'axe médian, c'est-à-dire que,

$$Q_b^\circ \cap AM(S) = \emptyset \quad (7.17)$$

Selon (7.14), (7.15), (7.16) et (7.17), on a immédiatement :

$$AM(S) = VOD(S) - Q_b \quad (7.18)$$

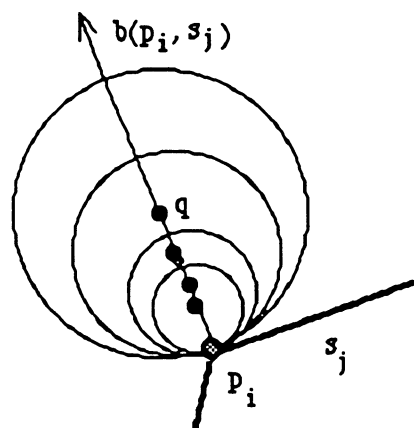


Figure 7.7 Les points intérieurs dans l'arête $b(p_i, s_j)$ ne sont pas centres de boules maximales où p_i est un sommet concave et s_j est un segment voisin de p_i

Ce dernier lemme nous permet de donner ci-dessous un algorithme pour extraire l'axe médian continu d'un objet plan S .

Algorithme AXE_MEDIAN;

entrée: un objet S ;

sortie: l'axe médian continu de S , $AM(S)$;

Début

1. Représenter l'objet S par un approximant polygonal, c'est-à-dire, la frontière de l'objet est représentée par la liste de sommets du polygone correspondant ;
2. Construire le diagramme de Voronoï généralisé pour l'intérieur de S en utilisant l'algorithme INVOR à partir de la représentation de la frontière du polygone, et noter ce diagramme par $INVOR(S)$;
3. Parcourir le graphe de $INVOR(S)$. Si une arête de Voronoï est connectée à un sommet concave (mesuré depuis l'intérieur) du polygone, on la supprime du graphe $INVOR(S)$;
4. $AM(S) \leftarrow INVOR(S)$;

Fin ;

Nous avons vu que l'extraction de l'axe médian n'est rien de plus que le calcul du diagramme de Voronoï de l'intérieur du polygone. Un autre algorithme selon le même principe a été proposé par Lee [LEE 82a]. Cependant, comme nous avons dit au chapitre 3, l'étape de calcul du diagramme de Voronoï dans son algorithme n'est pas complète. Précisément, les conditions d'arrêt du processus de fusion de deux sous-diagrammes de Voronoï ne sont pas complètes. Grâce à la modification apportée à cet algorithme, notre algorithme peut toujours donner un axe médian correct dans tous les cas.

7.4.2 Exosquelette

Le squelette du complémentaire à un ensemble d'objets donnés (ou le squelette de l'extérieur des objets) est un autre type de squelette. Il représente quelques caractéristiques externes des objets. On appelle ce type de squelettes *exosquelettes*. Soit S un ensemble d'objets. Nous notons le fond de S (ou le complémentaire de S) par S^c . Bien sûr, nous pouvons définir l'exosquelette comme les centres des boules maximales situées dans S . Mais, nous préférons ici la caractérisation par les trois valeurs égales de distances dont deux sont associées à deux points différents de l'ensemble des objets et l'une concerne la mesure de la distance minimale entre le point de l'exosquelette et S . Formellement, on a la définition suivante :

Définition 7.2 L'exosquelette d'un ensemble d'objets S , $ES(S)$, est défini par :

$$ES(S) = \{q \in S^c \mid \exists p_i, p_j \in \partial S (p_i \neq p_j) \text{ tels que } d(q, p_i) = d(q, p_j) = d(q, S)\} \quad (7.19)$$

Comme l'axe médian, un exosquelette est un sous-graphe du diagramme de Voronoï généralisé de l'extérieur des objets basé sur les éléments. En effet, le graphe du diagramme de Voronoï

généralisé de l'extérieur des objets peut être exprimé de manière similaire à (7.3):

$$VOD(S^c) = \{q \in S^c \mid d(q, e_i) = d(q, e_j) = d(q, S), e_i \neq e_j\} \quad (7.20)$$

De plus, nous pouvons aussi établir une relation entre l'exosquelette et le diagramme de Voronoï généralisé de l'extérieur. Ceci est:

Lemme 7.2 L'exosquelette d'un ensemble d'objets polygonaux S , $ES(S)$, est un sous-diagramme du diagramme de Voronoï généralisé de l'extérieur de S basé sur les éléments composant la frontière de S . Précisément, $ES(S)$ consiste en un graphe issu de $VOD(S^c)$ excepté les arêtes de Voronoï connectées aux sommets concaves de S (mesuré depuis l'extérieur de S).

Preuve. Similaire à celle du lemme 7.1.

Remarquons que la décomposition des éléments pour la construction du diagramme de Voronoï de l'extérieur est toujours faite sur la frontière d'un objet dans le sens inverse de celui de la décomposition des éléments pour la construction du diagramme de Voronoï de l'intérieur. Par conséquent, l'élimination d'arêtes de Voronoï pour obtenir l'axe médian est faite aux sommets concaves ; par contre, l'élimination d'arêtes de Voronoï pour extraire l'exosquelette s'effectue aux sommets convexes. Pour connaître un sommet concave (resp. convexe) il suffit de trouver un élément de type "point" dans la représentation du diagramme de Voronoï généralisé de l'intérieur des objets (resp. l'extérieur).

D'après le lemme 7.2 un simple algorithme peut être décrit pour extraire l'exosquelette.

Algorithme EXOSQUELETTE ;

entrée: l'ensemble d'objets S ;

sortie: l'exosquelette de S , $ES(S)$;

Début

1. Représenter chaque objet de S par une approximation polygonale, c'est-à-dire, la frontière de l'objet est représentée par la liste des sommets du polygone ;
2. Construire le diagramme de Voronoï de l'extérieur de S en utilisant l'algorithme EXVODMP (voir § 6.4) à partir de ∂S , et noter ce diagramme par $EXVOR(S)$;
3. Parcourir le graphe de $EXVOR(S)$. Si une arête de Voronoï est connectée à un élément du type POINT, on la supprime du graphe de $EXVOR(S)$;
4. $ES(S) \leftarrow EXVOR(S)$;

Fin ;

7.4.3 SKIZ

Considérons un ensemble de N ($N > 1$) objets disjoints dans le plan $S = \{S_1, S_2, \dots, S_N\}$, c'est-à-dire :

$$S = \bigcup_{i=1}^N S_i, \forall i, j \ i \neq j \ S_i \cap S_j = \emptyset \quad (7.21)$$

Afin de caractériser le domaine voisin de l'objet S_i , on utilise le terme "zone d'influence" associé à cet objet.

Définition 7.3 Pour chaque objet S_i de S , on définit sa zone d'influence $Z(S_i)$ par l'ensemble des points plus proches de S_i que de n'importe quel autre objet :

$$Z(S_i) = \{q \mid d(q, S_i) < d(q, S_j), i \neq j\} \quad (7.22)$$

Les points qui n'appartiennent à aucune zone d'influence constituent un type de squelette spécial appelé *SKIZ* (*SKELeton by Zones of Influence*) [SERRA 82]. Serra [SERRA 82] a utilisé les boules maximales pour décrire le *SKIZ*. Ici, nous nous intéressons encore à la caractérisation par trois quantités égales de distances pour formuler le *SKIZ*:

$$SKIZ(S) = \{q \in S^c \mid d(q, S_i) = d(q, S_j) = d(q, S), i \neq j\} \quad (7.23)$$

Cette définition souligne qu'un point de *SKIZ* est équidistant de deux objets différents en même temps qu'il est de distance minimale de l'ensemble des objets. Une telle définition basée sur les distances nous permet de rappeler immédiatement la définition du diagramme de Voronoï généralisé. En fait, le *SKIZ* d'un ensemble d'objets est précisément le diagramme de Voronoï généralisé basé sur les objets. On a formellement:

Lemme 7.3 Etant donné un ensemble d'objets polygonaux disjoints S , le *SKIZ* de S est identique au graphe du diagramme de Voronoï généralisé de l'extérieur de S basé sur les objets.

Preuve. Immédiatement d'après les définitions.

7.4.4 Résultats et discussions

Quelques sorties obtenues pour l'axe médian, l'exosquelette, et le *SKIZ* d'objets polygonaux sont respectivement donnés en figures 7.8, 7.9 et 7.10.

Comme nous l'avons défini, ces trois types de squelette sont tous déduits des diagrammes de

Voronoi généralisés. Cependant, les diagrammes de Voronoi généralisés sont tous construits sur la base d'un ensemble d'éléments composant la frontière des objets. De plus, les trois types de squelette sont composés de points contraints par (7.5), (7.9) et (7.12). Donc, trois types de branches de squelette sont possibles:

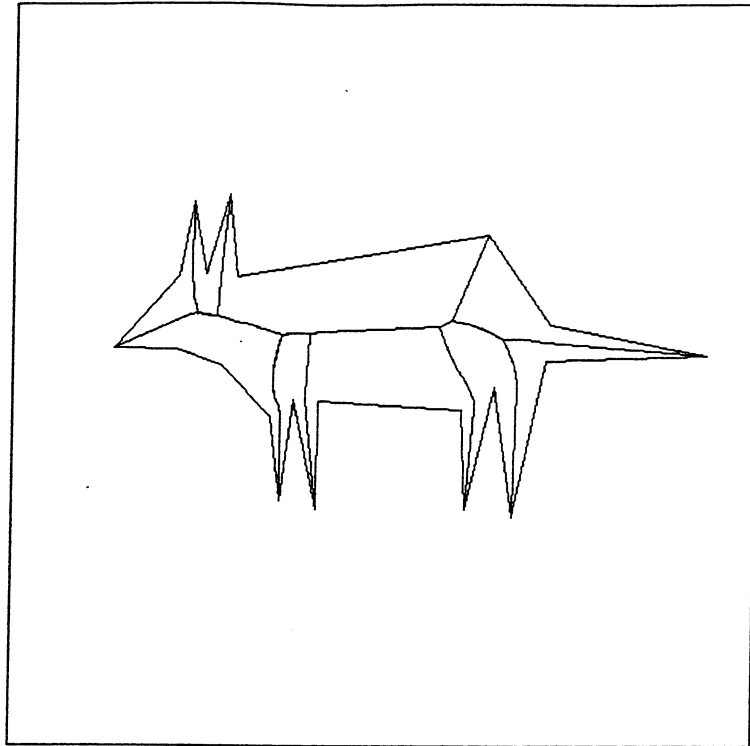
- (a) une portion de bissectrice d'angle entre deux éléments différents de type "segment";
- (b) une portion de bissectrice du type parabole déterminée par un élément de type "point" et un autre élément de type "segment";
- (c) Une portion de bissectrice entre deux points différents.

Donc, dans les cas de formes polygonales un squelette continu est composé de segments de droite et d'arcs paraboliques. En utilisant la terminologie du diagramme de Voronoi généralisé, nous pouvons caractériser les divers squelettes, c'est-à-dire que le concept original de Blum [BLUM 64] des squelettes peut être défini complètement selon le concept du diagramme de Voronoi généralisé.

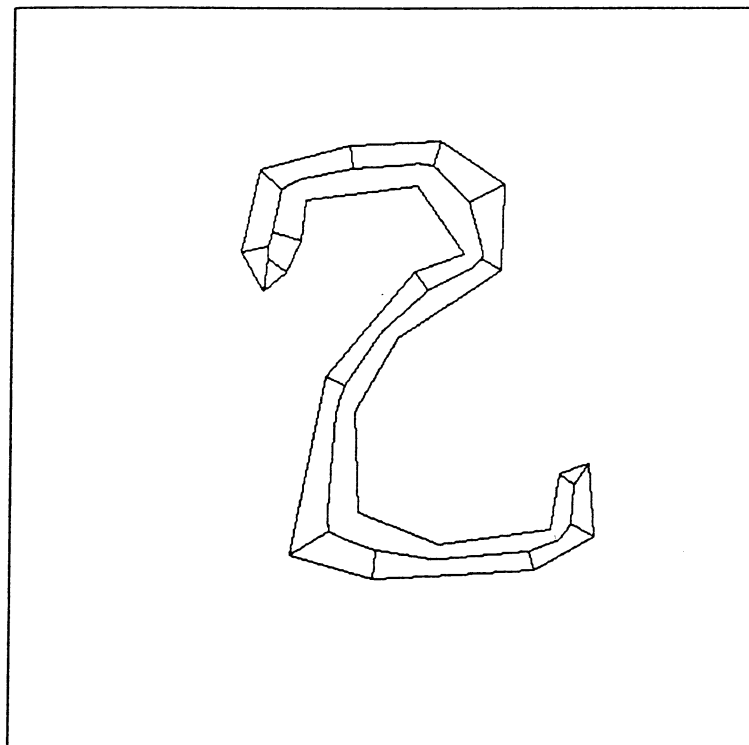
Quant aux différents types de points composant les squelettes, nous pouvons également les classer :

- (1) points normaux. Ce sont des points à l'intérieur d'une branche de squelette. Ces points sont situés dans les arêtes de Voronoi;
- (2) points multiples. Ce sont des sommets de Voronoi. Selon les corollaires 2.2 et 2.3, il existe un cercle centré en un sommet de Voronoi et tangent au moins à trois objets différents. Bien sûr, de tels cercles sont les boules maximales;
- (3) points terminaux. Ce sont des points appartenant à la fois au squelette et à la frontière d'objets. En fait, ce sont des sommets concaves des objets pour l'axe médian et des sommets convexes des objets pour l'exosquelette si la convexité des sommets des objets est mesurée depuis l'intérieur des objets. Il n'existe pas de tel point dans SKIZ. En raison de la représentation polygonale de la frontière des objets, l'explication du centre d'une boule maximale est non triviale.

Nous avons vu dans ce chapitre que la squelettisation d'objets polygonaux est un caractère inhérent au diagramme de Voronoi généralisé correspondant. Une telle squelettisation garde la propriété topologique désirée: invariabilité en translation, en rotation, et en changement d'échelle, elle vérifie l'idée originale de Blum.

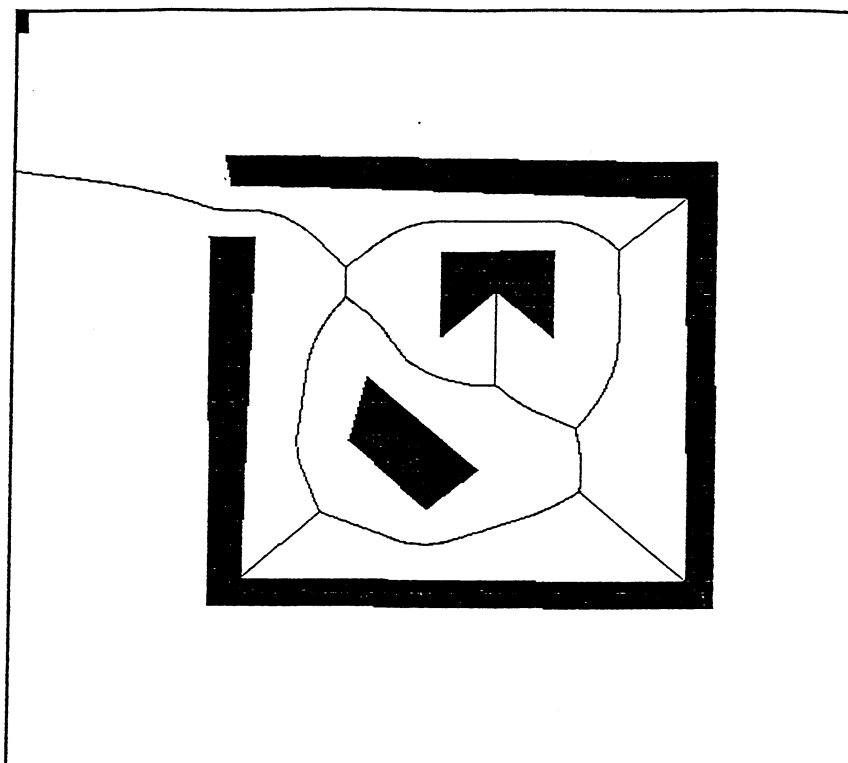


(a)

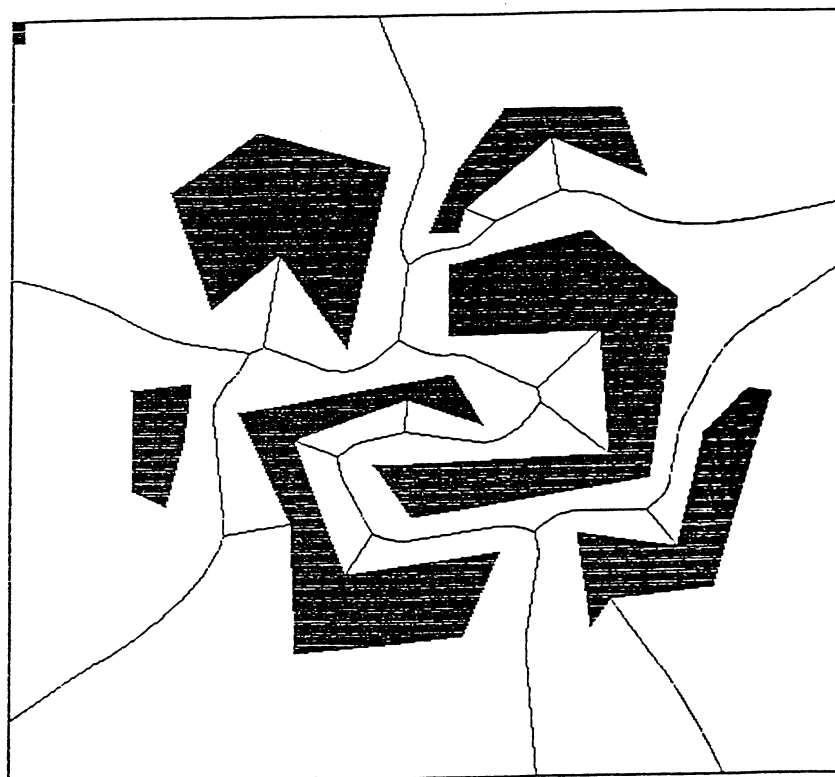


(b)

Figure 7.8 Exemples de l'axe médian

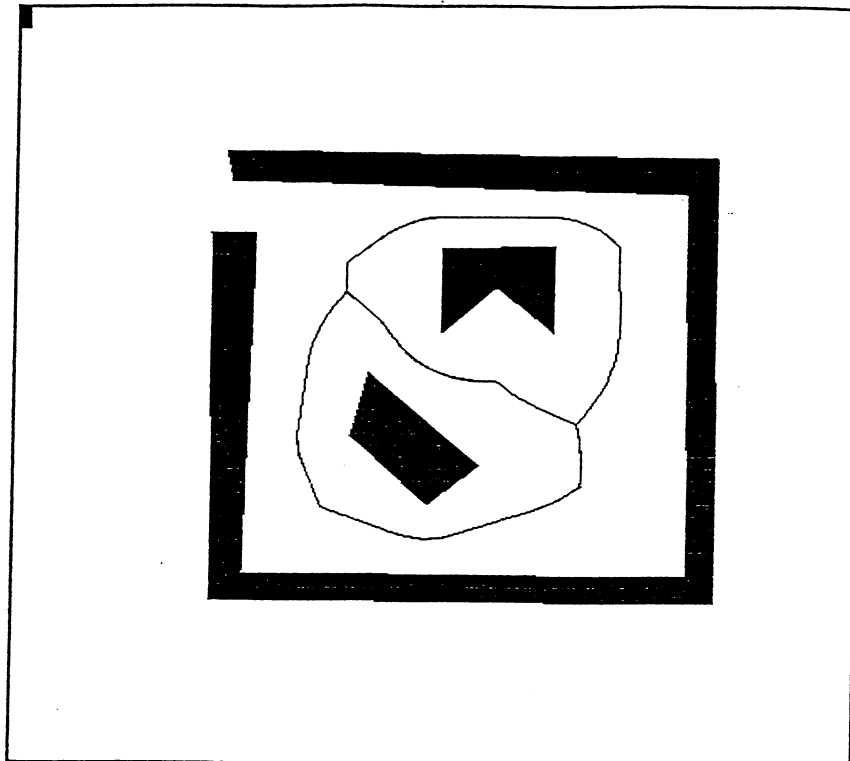


(a)

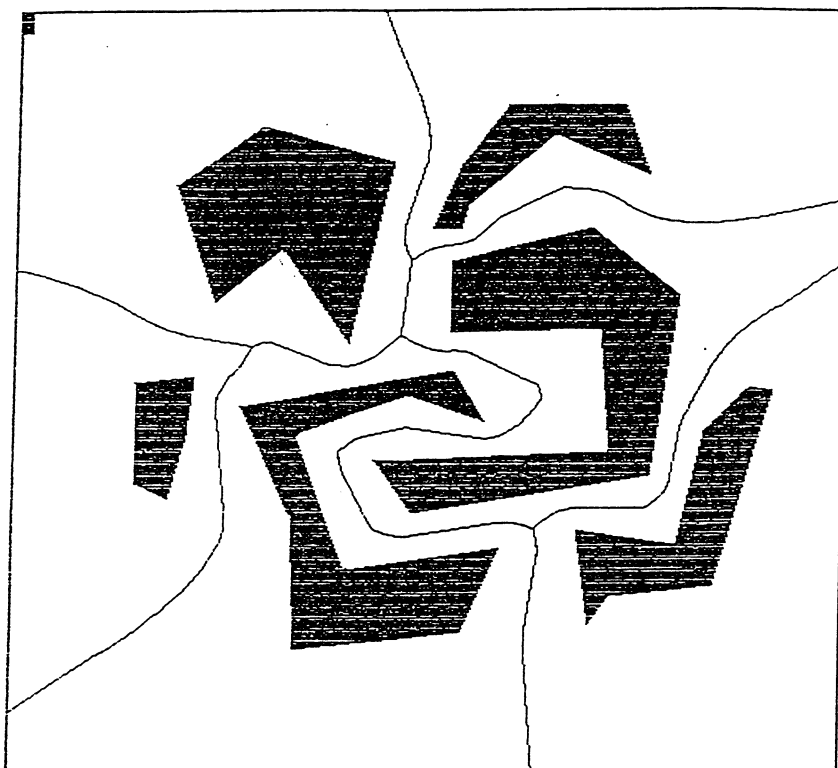


(b)

Figure 7.9 Exemples de l'exosquelette



(a)



(b)

Figure 7.10 Exemples de SKIZ

Conclusion et perspectives

Quelques problèmes théoriques et la réalisation du diagramme de Voronoï généralisé pour un ensemble de polygones ont été discutés dans le mémoire de cette thèse. Nous avons montré que le diagramme de Voronoï généralisé de K polygones quelconques peut être construit, par l'approche Divide-and-Conquer combinée avec l'approche incrémentale, dans le cas le plus défavorable en $O(\sum_{i=1}^K n_i \log n_i + \sum_{h=2}^K \sum_{i=1}^h n_i)$ où n_i est le nombre d'éléments composant la frontière du i -ème polygone de l'ensemble.

Une idée de calcul en *mode hiérarchique* ou *multi-échelles* a été appliquée. La construction en multi-échelles fournit un moyen pour partitionner et représenter un espace en différents niveaux. Pour calculer le diagramme de Voronoï généralisé d'un ensemble de polygones quelconques, nous construisons respectivement le diagramme de Voronoï de l'*intérieur* des polygones et celui de l'*extérieur* des polygones. Pour construire le diagramme de Voronoï de l'intérieur ou de l'extérieur d'un polygone, nous fusionnons les diagrammes de Voronoï de *chaînes polygonales* (voir la définition 3.2) issues de la frontière du polygone correspondant. Pour obtenir le diagramme de Voronoï d'une chaîne polygonale, nous décomposons la chaîne polygonale en *éléments* (voir la définition 2.2). Le diagramme de Voronoï généralisé à l'échelle de polygones est finalement construit à l'aide de celui à l'échelle d'éléments. Les algorithmes correspondants peuvent donc s'appliquer en pratique à un ensemble d'objets hybrides (points, segments, polygones).

Pour construire le diagramme de Voronoï généralisé de l'intérieur d'un polygone, nous avons modifié l'algorithme de Lee [LEE 82a] auquel des contre-exemples ont été rencontrés. Pour construire le diagramme de Voronoï généralisé de l'extérieur de polygones, nous avons proposé une approche de calcul d'un séparateur à chaque étape d'incrémental. Cette approche constitue la clé de nos algorithmes.

La construction en multi-échelles nous permet d'obtenir le diagramme de Voronoï généralisé au niveau de polygones, généralisant le diagramme de Voronoï généralisé de segments de droite comme ceux qui sont considérés dans [LEE 81] [YAP 87]. Comme nous manipulons des polygones, nous traitons un ensemble de segments comme un ensemble de polygones dégénérés.

En développant *a priori* la forme d'un séparateur, et en déterminant le nombre des branches (courbes connexes) du séparateur, le problème de la construction *semi-dynamique* du diagramme de Voronoï est aussi discuté. Plus précisément, à partir du diagramme de Voronoï d'un ensemble d'objets existants, quand un nouvel objet est inséré *aléatoirement* dans

l'ensemble, un nouveau diagramme de Voronoï correspondant doit être construit. Les trois types d'insertion sont envisagées : insertion d'un point, insertion d'un segment de droite et insertion d'un polygone convexe. Nous avons montré que l'insertions aléatoire d'un objet convexe peut être effectuée en temps linéaire dans le cas le plus défavorable.

L'un des exemples d'applications du diagramme de Voronoï généralisé se présente en analyse et compréhension de formes polygonales. Nous avons montré, au dernier chapitre, que les squelettes continus de formes polygonales sont des caractéristique inhérentes au diagramme de Voronoï généralisé associé. On peut obtenir directement, d'une part, les *axes médians* des formes à partir du diagramme de Voronoï généralisé à l'échelle d'éléments pour l'intérieur des formes. D'autre part, les exosquelettes et SKIZ des formes peuvent être directement déduits du diagramme de Voronoï généralisé de l'extérieur des formes polygonales.

Il existe de nombreux problèmes ouverts liés au diagramme de Voronoï généralisé. Nous en citons quelques-uns:

Existe t-il un algorithme en $O(K \log K)$ dans le cas le plus défavorable pour construire le diagramme de Voronoï généralisé de K polygones quelconques ? Une approche pourrait être l'utilisation du modèle "sweep-line" [FORTUNE 86] sur les polygones. Pour supporter la technique "sweep-line", une structure de données avancée associée directement aux polygones doit être développée pour décrire l'état de la ligne de balayage. Ainsi, une transformation géométrique appropriée devrait être introduite ou bien pourrait empruntée à la transformation de Fortune. Une autre approche pourrait consister à trouver une technique qui permette de calculer en temps linéaire le séparateur entre deux ensembles *arbitraires* de polygones afin que l'utilisation de la stratégie "divide-and-conquer" à l'ensemble de polygones puisse produire un algorithme en $O(K \log K)$. Cette dernière approche nécessite de manipuler des séparateurs non connexes. Il faut donc trouver au moins un point, par exemple un initialisateur défini au chapitre 5, pour chaque branche connexe du séparateur. Bien que le lemme 5.2 a montré qu'on peut déterminer *a priori* la forme d'un séparateur $Sep(L, R)$ selon la position relative de l'enveloppe convexe de R vis-à-vis de celle de L , L et R ne sont pas deux ensembles quelconques de polygones. Il semble qu'il existe des difficultés à déterminer la forme du séparateur entre deux ensembles arbitraires d'objets polygonaux.

Un autre problème important est de réaliser une insertion aléatoire d'un polygone quelconque ainsi que la construction complètement dynamique. Dans le contexte de la construction dynamique, il faut être capable d'*insérer*, de *façon aléatoire*, un objet polygonal dans un ensemble d'objets existants et d'*enlever* un polygone. Pour insérer un objet quelconque, il faut également trouver un algorithme efficace pour manipuler un séparateur plus compliqué que celui décrit dans le cas d'insertion d'un objet convexe (voir le chapitre 6). L'élimination d'un

objet polygonal autre qu'un point [GOWDA 83] demande une structure de données plus efficace pour supporter l'opération correspondante et vise un algorithme en temps linéaire pour mettre à jour le diagramme de Voronoï généralisé.

L'analyse de la complexité du cas moyen est généralement plus compliquée. Ceci demande des suppositions statistiques sur certaines données géométriques. Si nous pouvions démontrer que l'opération de modification locale, à chaque étape dans l'algorithme EXVODMP, pouvait être effectuée en moyenne en $O(k)$, où k est la taille du polygone introduit à cette étape ou bien le nombre *moyen* d'éléments par polygone de l'ensemble donné, nous pourrions aboutir à une complexité linéaire dans le cas moyen pour l'algorithme EXVODMP.

A partir de diagramme de Voronoï généralisé, de nombreux problèmes liés à la géométrie peuvent être résolus efficacement. Notamment, il a été montré ces dernières années que le problème de la planification du mouvement en robotique, où on doit trouver un chemin sans obstacles entre deux positions données d'un robot dans un environnement modélisé par un ensemble de polygones, peut être résolu à l'aide du diagramme de Voronoï généralisé (au sens de partition et de graphe). Comme le diagramme de Voronoï généralisé fournit une représentation et partition exacte d'un environnement, la solution au problème de la planification du mouvement issue du diagramme de Voronoï est donc précise. De plus, est-il envisageable que l'on puisse, à l'aide du diagramme de Voronoï généralisé, résoudre le problème de la planification du mouvement d'une façon *non heuristique* [SHARIR 89] ?

Enfin, comment généraliser le diagramme de Voronoï d'un ensemble de polygones à celui d'objets dans un espace de dimension supérieure est une grande provocation. Typiquement, nous voudrions construire le diagramme de Voronoï d'un ensemble de *polyèdres* dans l'espace 3D. Dans ce cas-ci, un séparateur entre deux objets serait composé de *hypersurfaces*. En utilisant l'idée de décomposition des surfaces des polyèdres en éléments (qui pourraient être : des sommets, des arêtes ouvertes et des plans ouverts dans l'espace) et en explorant l'approche incrémentale, on pourrait probablement aboutir à la construction du diagramme de Voronoï d'un ensemble des polyèdres dans l'espace.

Bibliographie

- [AGGARWAL 87] A. Aggarwal, L. Guibas, J. Saxe and P.W. Shor, "A linear time algorithm for computing the Voronoi diagram of a convex polygon," *Proc. 19th ACM Annual Symp. on Theory of Comput.* pp. 39-45, May, 1987.
- [AHUJA 82] N. Ahuja, "Dot pattern processing using Voronoi neighborhoods," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-4, No. 3, pp. 336-343, 1982.
- [AHUJA 85] N. Ahuja, B. An and B. Schachter, "Image representation using Voronoi tessellation," *Comput. Vision, Graphics and Image Processing*, vol. 29, pp. 286-295, 1985.
- [AHO 83] A. V. Aho, J. E. Hopcroft and J. D. Ullman, *Data Structures and algorithms*, Addison-Wesley, 1983.
- [ATALLAH 89] M. J. Atallah, R. Cole and M. T. Doodrich, "Cascading divide-and-conquer: a technique for designing parallel algorithms," *SIAM J. Comput.* Vol. 18. No. 3, pp. 499-532, June 1989.
- [AURENHAMMER 84] F. Aurenhammer and H. Edelsbrunner, "An optimal algorithm for constructing the weighted Voronoi diagram in the plane," *Pattern Recognition*, Vol. 17, No. 2, pp. 251-257, 1984.
- [AURENHAMMER 86] F. Aurenhammer, "The one-dimensional weighted Voronoi diagram," *Info. Process. Letters*, 22 (1986), 119-123.
- [AURENHAMMER 87] F. Aurenhammer, "Power diagrams: properties, algorithms and applications," *SIAM J. Comput.* vol. 16, No. 1, 1987, pp. 78-96.
- [AURENHAMMER 88] F. Aurenhammer, "Improved algorithms for discs and balls using power diagrams," *Journal of Algorithms*, Vol. 9, pp. 151-161, 1988.
- [BENTLEY 78] J. L. Bentley and M. I. Shamos, "Divide and conquer for linear expected time," *Inform. Proces. Lett.*, vol. 7, No. 2, Feb., 1978.
- [BENTLEY 80] J. L. Bentley, "Multidimensional divide and conquer," *Comm. ACM* 23(4), 214-229, 1980.
- [BLUM 64] H. Blum, "A transformation for extracting new descriptors for shape," *Symp. on Models for the perception of speech and visual form*. Boston, Nov. 1964, (MIT Press, Ed. by W. Wathen-Dunn, pp. 362-380, 1967).
- [BLUM 78] H. Blum and R. Nagel, "Shape description using weighed symmetric axis features," *Pattern Recognition*, vol.10, pp. 167-180, 1978.

- [BOISSONNAT 84] J. D. Boissonnat, "Geometric structures for three-dimensional shape representation," *ACM Trans. Graph.* Vol. 3, No. 4, Oct. 1984, 266-286.
- [BOISSONNAT 88] J. D. Boissonnat, "Shape reconstruction from planar cross section," *Comput. Vision, Graphics and Image Processing*, 44, pp. 1-29, 1988.
- [BOISSONNAT 90] J. D. Boissonnat, O. Devillers, R. Schott, M. Teillaud and M. Yvinec, "On-line geometric algorithms with good expected behaviours," *Journées de Géométrie algorithmique*, INRIA Sophia-Antipolis, France, 18-20 Juin, 1990.
- [BORGEFOIS 84] G. Borgefors, "Distance transformations in arbitrary dimensions," *Comput. Graphics and Image processing*, 27, pp. 321-345, 1984.
- [BORGEFORS 86] G. Borgefors, "Distance transformations in digital images," *Comput. Graphics and Image Processing*, 34, pp. 344-371, 1986.
- [BOWYER 81] A. Bowyer, "Computing Dirichlet tessellations," *The Computer J.* vol. 24, No. 2, pp. 162-166, 1981.
- [BROWN 79] K. Q. Brown, "Voronoi diagrams from convex hulls," *Inform. Processing Lett.*, vol. 18, No. 5, 1979, pp. 223-227.
- [BYKAT 78] A. Bykat, "Convex hull of a finite set of points in two deminsions," *Inf. Process. Lett.* 7, 296-298, 1978.
- [CHANG 90] M. S. Chang, N. F. Huang and C. Y. Tang, "An optimal algorithm for constructing oriented Voronoi diagrams and geographic neighborhood graphs," *Info. Proc. Letters*, 35, (1990), 255-260.
- [CHASSERY 79] J. M. Chassery, "Connectivity and consecutivity in digital picture," *Comput. Graphics and Image processing*, vol.9, pp. 294-300, 1979.
- [CHASSERY 83] J. M. Chassery, "Discrete Convexity: definition, parameterization, and compatibility with continous convexity," *Comput. Vision, Graphics and Image Process.* 21, 326-344, 1983.
- [CHASSERY 89] J. M. Chassery et M. Melkemi, "Segmentation d'images dans un environnement de diagramme de Voronoï," In *Proceedings PIXIM89*, Sept., 1989, Paris, pp.197-200.
- [CHASSERY 91] J. M. Chassery and A. Montanvert, *Géométrie Discrète en Analyse d'Images*, Hermès ed., 1991.
- [CHAZELLE 87] B. Chazelle and H. Edelsbrunner, "An improved algorithm for constructing k-th order Voronoi diagrams," *IEEE Trans. Comput.* vol. C-36, No. 11, pp. 1349-1354, 1987.
- [CHEN 89] C. L. Chen, "Computing the convex hull of a simple polygon," *Pattern Recognition* 22, 561-565, 1989.
- [DANIELSSON 80] P. E. Danielsson, "Euclidean distance mapping," *Comput. Graphics and Image processing*, 7, pp.227-248, 1980.
- [DAVIES 81] E. R. Davies and A. P. N. Plummer, "Thinning algorithms: a critique and a new methodology," *Pattern Recognition*, 14, pp.53-63, 1981.

- [DEHEIN 87] F. Dehein and R. Klein, "A sweepcircle algorithm for Voronoi diagrams (extended abstract)," *Lecture Notes in Comput. Sci.* vol. 314, WG'87, 1987, pp. 59-69.
- [DOBKIN 76] D. P. Dobkin and R. J. Lipton, "Multidimensional searching problems," *SIAM J. Computing*, vol. 5, no. 2. pp. 181-186, 1976.
- [DRYSDALE 79] R. L. Drysdale III, "Generalized Voronoi Diagrams and Geometric Searching," Ph. D. thesis, Stanford University, 1979.
- [DWYER 88] R. A. Dwyer, "Higher-dimensional Voronoi diagrams in linear expected time," *Technical Report CMU-CS-88-100*, Carnegie-Mellon University, Feb, 1988.
- [EDAHIRO 84] M. Edahiro, I. Kokubo and T. Asano, "A new point-location algorithm and its practical efficiency—comparison with existing algorithms," *ACM Trans. Graphics* 3(2), 86-109, 1984.
- [EDELBRUNNER 84] H. Edelsbrunner, "Key-problems and key-methods in computational geometry," *Lecture Notes in Comput. Sci.* vol. 166, STACS'84, 1984, pp. 1-13.
- [EDELBRUNNER 86] H. Edelsbrunner, L.J. Guibas and J. Stolfi, "Optimal point location in a monotone subdivision," *SIAM J. Comput.* vol. 15, No. 2, 1986, pp. 317-343.
- [FAUGERAS 90] O. D. Faugeras, E. Le Bras-Mehlman and J. D. Boissonnat, "Representating stereo data with the Delaunay triangulation," *Artificial Intelligence* 44 (1990) 41-87.
- [FENG 75] H. F. Feng and T. Pavlidis, "Decomposition of polygons into simpler components: feature generation for syntactic pattern recognition," *IEEE Trans. Comput.*, vol. C-24, pp. 636-650, 1975.
- [FORTUNE 86] S. Fortune, "A sweepline algorithm for Voronoi diagrams," *Proc. of 2nd Annu. symp. on Computational Geometry*, Yorktown Heights, NY, June 1986, pp. 313-322.
- [GHOSH 83] S. K. Ghosh and R. K. Shyamasundar, "A linear time algorithm for obtaining the convex hull of a simple polygon," *Pattern Recognition* 16, 587-592, 1983.
- [GOWDA 83] I. G. Gowda, D. G. Kirkpatrick, D. T. Lee and A. Naamad, "Dynamic Voronoi diagrams," *IEEE Trans. Inform. Theory*, vol. IT-29, No. 5, pp. 724-731, 1983.
- [GRAHAM 72] R. L. Graham, "An efficient algorithm for determining the convex hull of a planar set," *Inf. Process. Lett.* 1:132-133, 1972.
- [GRAHAM 83] R. L. Graham and F.F. Yao, "Finding the convex hull of a simple polygon," *J. algorithm* 4, 324-331, 1983.
- [GREEN 78] P. J. Green and R. Sibson, "Computing Dirichlet tessellations in the plane," *The Computer J.* vol. 21, No. 2, pp. 168-173, 1978.
- [GUIBAS 85] L. J. Guibas and J. Stolfi, "Primitives for the manipulation of general subdivisions and the computation of Voronoi diagrams," *ACM Trans. on Graphics*, Vol. 4, No. 2, pp. 74-123, April 1985.
- [HU 89] H. T. Hu and J. M. Chassery, "Diagrammes de Voronoï généralisés: premiers résultats", *Rapport de recherche* No. RR-799-M, IMAG, Grenoble, France, Dec., 1989.

- [HU 91] H. T. Hu and J. M. Chassery, "Squelette et diagramme de Voronoï généralisé en compréhension de formes," apparaîtra au 8^e congrès RFIA-AFCET, Lyon, Nov., 1991.
- [ILARI 90] J. Ilari and C. Torras, "2D path planning: a configuration space heuristic approach," *The Internat. Journal of Robotics Research* Vol. 9, No. 1, Feb. 1990.
- [IMAI 85] H. Imai, M. Iri and K. Murota, "Voronoi diagram in the Laguerre geometry and its applications," *SIAM J. Computing*, pp. 93-105, 1985.
- [KIRKPATRICK 79] D. G. Kirkpatrick, "Efficient computation of continuous skeletons," in *Proc. 20th Annu. Symp. Found. Computer Sci.*, Oct. 1979, pp. 18-27.
- [KIRKPATRICK 83] D. G. Kirkpatrick, "Optimal search in planar subdivisions," *SIAM J. Comput.* 12 (1), 1983, pp. 28-35.
- [KIRKPATRICK 86] D. G. Kirkpatrick and R. Seidel, "The ultimate planar convex hull algorithm?," *SIAM J. Comput.*, 15, 287-299, 1986.
- [KLEIN 88] R. Klein and D. Wood, "Voronoi diagrams based on general metrics in the plane (extended abstract)," *Lecture Notes in Comput. Sci.* Vol. 294, PP. 281-291, STACS 88, 1988.
- [KUBLER 89] O. Kübler, F. Klein, R. Ogniewicz and U. Kienholz, "Isolation and identification of abutting and overlapping objects in binary images," *Proc. of the 5th intern. Conf. on Image Analysis & Proc.*, 20-22, sept., 1989, Positano, Italy, pp. 340-347.
- [LACOLLE 88] B. Lacolle, "Géométrie algorithmique : Présentation, spécificité et complexité," *Journées de Géométrie Discrète, Géométrie Algorithmique, Passage du Discret au Continu*, Grenoble, France, 21-23 Nov., 1988.
- [LANTUEJOUL 80] C. Lantuejoul, "Skeletonization in quantitative metallography," in *Issues in Digital Image Processing*, edited by R. M. Haralick and J. C. Simon, Sijthoff and Noordhoff, 1980. pp. 107-135.
- [LEBRAS 89] E. Le Bras-Mehlman, "Représentation de l'environnement d'un robot mobile," Thèse de l'Université de Paris-sud, centre d'Orsay, Avril 1989.
- [LEE 77] D. T. Lee and F. P. Preparata, "Location of a point in a planar subdivision and its applications. *SIAM J. Comput.* 6(3), 1977, pp. 594-606.
- [LEE 78] D. T. Lee, "Proximity and reachability in the plane," *Tech. Rep.* No. R-831, Coordinated Sci. Lab., Unvi. of Illinois at Urbana, IL, 1978.
- [LEE 80a] D. T. Lee, "Two-dimensional Voronoi diagrams in the L_p -metric," *J. ACM*, vol. 27, No. 4, 1980, pp. 604-618.
- [LEE 80b] D. T. Lee and C. K. Wong, "Voronoi diagram in $L_1(L_\infty)$ metrics with 2-dimensional storage applications," *SIAM J. Comput.* vol. 9, No. 1, 1980.
- [LEE 81] D. T. Lee and R. L. Drysdale, "Generalization of Voronoi diagrams in the plan," *SIAM J. Comput.*, vol. 10, No. 1, 1981.
- [LEE 82a] D. T. Lee, "Medial axis transformation of a planar shape," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-4, pp. 363-369, 1982.

- [LEE 82b] D. T. Lee, "On k-nearest neighbor Voronoi diagrams in the plane," *IEEE Trans. Computers*, vol. C-31, pp. 478-487, 1982.
- [LEE 84] D. T. Lee and F. P. Preparata, "Computational geometry- A survey," *IEEE Trans. Comput.*, vol. C-33, No.12, pp. 1072-1101, 1984.
- [LEVEN 87a] D. Leven and M. Sharir, "An efficient and simple motion planning algorithm for a ladder amidst polygonal barriers," *J. of Algorithms* 8, 192-215 (1987)
- [LEVEN 87b] D. Leven and M. Sharir, "Planning a purely translational motion for a convex object in two-dimensional space using generalized Voronoi diagrams," *Discrete Comput. Geom.* 2: 9-31 (1987)
- [LU 89] Z. Y. Lu, "Application of Delaunay triangulation in image coding," *Proc. of the 5th Intern. conf. on image analysis and processing*, pp. 222-226, 20-22, Sept., 1989. Positano, Italy.
- [MATSUYAMA 83] T. Matsuyama and T. Y. Phillips, "Digital realization of labeled Voronoi diagram and its application to closed boundary detection," *Techn. Report CAR-TR-22*, Oct., 1983, University of Maryland.
- [McCALLUM 79] D. McCallum and D. Avis, "A linear algorithm for finding the convex hull of a simple polygon," *Inf. Process. Lett.* 9, 201-206, 1979.
- [MELKEMI 90] M. Melkemi, H.T. Hu, E. Bertin et J. M. Chassery, "Voronoi ponctuel, généralisé et 3D: approches algorithmiques et applications en analyse d'image," *Journées de Géométrie Algorithmique*, INRIA Sophia-Antipolis, 18-20 Juin 1990.
- [MESHKAT 87] S. N. Meshkat and C. M. Sakkas, "Voronoi diagram for multiply-connected polygonal domains II: implementation and application," *IBM J. Res. Develop.* vol. 31, No. 3, pp. 373-381, 1987.
- [MONTANARI 68] U. Montanari, "A methode for obtaining skeleton using a quai-euclidean distance," *J. ACM.*, 15, pp. 600-624, 1968.
- [MONTANARI 69] U. Montanari, "Continuous skeletons from digitized images," *J. ACM*, vol. 16, no.4, pp. 534-549, 1969.
- [MONTANVERT 87] A. Montanvert, "Contribution au traitement de formes discrètes: Squelettes et codage par graphe de la ligne médiane," Thèse de l'Université Grenoble I & de l'INPG, 1987.
- [O'DUNLAING 85] C. O'Dunlaing and C. K. Yap, "A 'retraction' method for planning the motion of a disc," *J. of algorithms*, vol. 6, pp.104-111, 1985.
- [O'DUNLAING 86] C. O'Dunlaing, M. Sharir and C. K. Yap, "Generalized Voronoi diagrams for moving a ladder. I: Topological analysis," *Comm. on Pure and Applied Mathe.*, vol. XXXIX, pp.423 -483, 1986.
- [OHYA 78] T. Ohya, M. Iri and K. Murota, "A fast Voronoi diagram algorithm with quaternary tree bucketing," *Infor. Process. Letters*, vol. 18, 1984, pp.227-231.
- [OHYA 84] T. Ohya, M. Iri and K. Murota, "Improvements of the incremental method for the Voronoi diagram with computational comparison of various algorithms," *J. Operations Res. Soci of Japan*, Vol. 27, No. 4, pp. 306-337, 1984.
- [ORLOWSKI 83] M. Orłowski, "On the conditions for success of Sklansky's convex hull algorithm," *Pattern Recognition* 16, 579-586, 1983.

- [ORLOWSKI 85] M. Orlowski, "A convex hull algorithm for planar simple polygons," *Pattern Recognition* 16, 361-366, 1985.
- [O'ROURKE 82] J. O'Rourke, C. B. Chien, T. Olson and D. Naddor, "A new linear algorithm for intersecting convex polygons," *Computer Graphics & Image Processing*, 19, 384-391, 1982.
- [PAVLIDIS 78] T. Pavlidis, "SURVEY: A review of algorithms for shape analysis," *Comput. Graphics Image Processing*, vol. 7. pp. 243-258, 1978.
- [PAVLIDIS 80] T. Pavlidis, "A thinning algorithm for discrete binary images," *Comput. Graphics and Image Processing*, 13, 1980, pp. 142-157.
- [PAVLIDIS 81] T. Pavlidis, "Shape description," In *Digital Image Processing*, Ed. by S.C. Simon and R. M. Haralick, pp. 289-310, D. Reidel Publishing Company, 1981.
- [PEROCHE 88] D. Péroche, "Modèles de partitionnement par diagrammes de Voronoï : aspects algorithmiques," *Journées de Géométrie Discrète, Géométrie Algorithmique, Passage du Discrète au Continu*, Grenoble, France, 21-23 Nov., 1988.
- [PERSON 77] E. Person and K. S. Fu, "Shape discrimination using Fourier descriptors," *IEEE Trans. Syst. Man. Cybern.* Vol. SMC-7, no. 3. pp. 170-179, 1977.
- [PFLATZ 67] J. Pflatz and A. Rosenfeld, "Computer representation of planar region by their skeleton," *Comm. ACM.*, Vol. 10. no. 2, 1967, pp. 119-125.
- [PHILLIPS 84] T. Y. Phillips and T. Matsuyama, "The labeled discrete Voronoi diagram," *Tech. Report*, TR-CAR-4, University of Maryland, 1983.
- [PREILOWSKI 88] W. Preilowski and W. Mumbeck, "A time-optimal parallel algorithm for computing of Voronoi diagrams," *Lecture Notes in Comput. Sci.* vol. 344, Proc. Intern. Workshop WG'88, June, 1988, pp. 424-434.
- [PREPARATA 77] F. P. Preparata and S. J. Hong, "Convex hull of finite set of points in two and three dimensions," *Commun. ACM.* 20, 87-93, 1977.
- [PREPARATA 81] F. P. Preparata, "A new approach to planar point location," *SIAM J. Comput.*, 10(3), 1981, pp.473-483.
- [PREPARATA 88] F. P. Preparata and M. I. Shamos, *Computational Geometry: An introduction*, (nouvelle version) Berlin: Springer-Verlag, 1988.
- [PREPARATA 89] F. P. Preparata and R. Tamassia, "Dynamic planar point location with optimal query time," *Lecture Notes in Comput. Sci.* Vol. 394, pp. 84-95, 1989.
- [ROM 88] H. Rom and S. Peleg, "Image representation using Voronoi tessellation: Adaptive and secure," In *Proceedings of Computer Vision & Pattern Recog.* June, 1988, Michigan.
- [RONSE 89] C. Ronse, "A bibliography on digital and computational convexity (1961-1988)," *IEEE Trans. on Patt. Analy. and Machine Intel.* Vol. 11, No.2, Feb. 1989, 181-190.
- [ROSENBERGER 88] H. Rosenberger, "Order-k Voronoi diagrams of sites with additive weights in the plane," *Technical Report* No. UIUCDCS-R-88-1431, Depart. of Comput. Sci., University of Illinois at Urbana-Champaign, 1988.

- [ROSENFELD 66] A. Rosenfeld and J. L. Pfaltz, "Sequential operations in digital picture processing," *J. ACM.* 13, 1966, 471-494.
- [ROSENFELD 68] A. Rosenfeld and J. L. Pfaltz, "Distance functions on digital picture," *Pattern Recognition*, 1, No. 1, pp. 33-61, 1968.
- [ROSENFELD 73] A. Rosenfeld and E. Johnston, "Angle detection on digital curves," *IEEE Trans. Comput.*, vol. C-22, pp. 875-878, 1973.
- [ROSENFELD 82] A. Rosenfeld and A. C. Kak, *Digital Picture Processing*, (2nd Edition), Academic Press, 1982.
- [SCHACHTER 78] B. Schachter, "Decomposition of polygons into convex sets," *IEEE Trans. on Comput.* 27(11), Nov, 1978, pp. 1078-1082.
- [SCHMITT 90] F. Schmitt and H. Borouchaki "Algorithme rapide de maillage de Delaunay dans R^d ," *Journée de Géométrie Algorithmique*, INRIA Sophia-Antipolis, 18-20 Juin 1990.
- [SCHWARZKOPF 89] O. Schwarzkopf, "Parallel computation of discrete Voronoi diagrams," *Lecture Notes in Comput. Sci.* vol. 349, STACS, 1989, pp.193-204.
- [SCHWARTZ 87] J. T. Schwartz, M. Sharir and J. E. Hopcroft (eds.), *Planning, Geometry and Complexity of Robot Motion*, Ablex Publishing, Norwood, N. J., 1987.
- [SCHWARTZ 89] J. T. Schwartz and M. Scharir, "A survey of motion planning and related geometric algorithms," in *Geometric Reasoning*, Ed. by D. Kapur & J. C. Munday, The MIT Press, 1989, pp. 157-169.
- [SCHWARTZ 90] J. T. Schwartz and M. Sharir, "Algorithmic motion planning in robotics," In *Handbook of Theoretical Computer Science*, Ed. by J. Van Leeuwen, Elsevier Science, B. V. 1990, pp. 392-430.
- [SERRA 82] J. Serra, *Image Analysis and Mathematical Morphology*, Volume 1, Academic Press, 1982.
- [SERRA 88] J. Serra, *Image Analysis and Mathematical Morphology*, Volume 2: *Theoretical Advances*, Ed. by J. Serra, Academic Press, 1988.
- [SHAMOS 75a] M. I. Shamos, "Geometric complexity," In *Proc. of the 7th ACM Symposium on Theory of Computing*, New York, 1975, pp. 224-233.
- [SHAMOS 75b] M. I. Shamos and D. Hoey, "Closest-point problems," *Sixteenth Annual IEEE Symposium on Foundations of Computer Science*, pp. 151-162, Oct. 1975.
- [SHAMOS 78] M. I. Shamos, "Computational geometry," Ph.D. Thesis, Dept. of Comput. Sci., Yale University, 1978.
- [SHARIR 89] M. Sharir, "Algorithmic motion planning in robotics," *IEEE Computer*, March 1989, pp. 9-19.
- [SHIN 86] S.Y. Shin and T. C. Woo, "Finding the convex hull of a simple polygon in linear time," *Pattern Recognition* 6, 453-458, 1986.
- [SKLANSKY 72] J. Sklansky, "Measuring concavity on a rectangular mosaic," *IEEE Trans. Comp.*, 21. 1355-1364, 1972.

- [SKLANSKY 82] J. Sklansky, "Finding the convex hull of a simple polygon," *Pattern Recognition Lett.* 1, 79-83, 1982.
- [SRINIVASAN 87] V. Srinivasan and L. R. Nackman, "Voronoi diagram for multiply-connected polygonal domains I: algorithm," *IBM J. Res. Develop.* vol.31, No.3, pp. 361-372, 1987.
- [SUGIHARA 89] K. Sugihara and M. Iri, "Construction of the Voronoi diagramme for one million generators in single-precision arithmetic," *First Canadian Conference on Computational Geometry*, August 21-25, 1989, Montreal, Canada.
- [TAKAHASHI 89] O. Takahashi and R.J. Schilling, "Motion planning in a plane using generalized Voronoi diagrams," *IEEE Trans. on Robotics & Automations*, Vol. 5. No. 2, april, 1989, pp. 143-150.
- [TOUSSAINT 80] G. T. Toussaint, "Pattern recognitiuon and geometric complexty," In *Proc. 5th Intern. Conf. Pattern Recogn.*, pp. 1324-1347, Dec., 1980.
- [TOUSSAINT 82] G. T. Toussaint and D. Avis, "On a convex hull algorithm for polygons and its application to triangulation problems," *Pattern Recognition* 15, 23-28, 1982.
- [TOUSSAINT 83] G. T. Toussaint and H. E. Gindy, "A counterexample to an algorithm for computing monotone hulls of simple polygons," *Pattern Recognition Lett.* 1, 219-222, 1983.
- [TOUSSAINT 85] G. T. Toussaint (ed.), *Computational Geometry*, North-holland, 1985.
- [TOUSSAINT 86] G. T. Toussaint, "Computational geometry and morphology," *Technical Report*, No. SOCS-86.3, Feb., 1986, School of Computer Science, McGill University, Montreal, Canada.
- [TÜCERYAN 90] M. Tüceryan and A. K. Jain, "Texture segmentation using Voronoi polygons," *IEEE Trans. on Patten Recogn. and Machine Intelligence*, vol.12, No.2, pp. 211-216, Feb, 1990.
- [VINCENT 90] L. Vincent, "Algorithmes morphologiques à base de files d'attente et de lacets. Extension aux graphes," Thèse d'Ecole des Mines, Paris, Mai, 1990.
- [VORONOI 08] G. Voronoï, "Nouvelles applications des parametres continus à la theorie des formes quadratiques," deuxième Mémoire: Recherches sur les paralléopèdre primitifs, *J. reine angew. Math.* 134, 198-287 (1908).
- [WATSON 81] D. F. Watson, "Computing the n-dimensional Delaunay tessellation with application to Voronoi polytopes," *The Computer J.* vol. 24, No. 2, pp.167-172, 1981.
- [XIA 89] Y. Xia, "Skeletonization via the realisation of the fire front's propagation and extinction in digital binary shapes," *IEEE Trans. on Pattern Anal. and Mach. Intell.*, vol. PAMI-11, no.10, pp. 1076-1086, Oct., 1989.
- [YAP 87] C. K. Yap, "An $O(n \log n)$ algorithm for the Voronoi diagram of a set of simple curve segments," *Discrete & Computational Geometry*, No.2, pp. 365-393, 1987.

AUTORISATION DE SOUTENANCE

DOCTORAT DE L'UNIVERSITE JOSEPH FOURIER - GRENOBLE 1

Vu les dispositions de l'Arrêté du 5 juillet 1984,

Vu les rapports de **Monsieur. Bernard .LACOLLE**.....

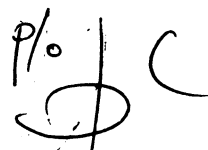
Monsieur. Francis .SCHMITT.....

Monsieur. Hai-Tao. .HU.....est autorisé(e)
à présenter une thèse en vue de l'obtention du **titre.de.Docteur.de.l'**.....
.Université. (nouveau régime)......

Grenoble, le **27 JUIN 1991**

Le Président de l'Université
Joseph Fourier - Grenoble 1

A. NEMOZ

P/° C

D. CORDARY
Vice-Président

Résumé

Ce mémoire présente les problèmes théoriques et pratiques en vue de la réalisation du diagramme de Voronoï généralisé (DVG) pour un ensemble de polygones en métrique euclidienne. Sur la base d'une idée de construction en multi-échelles, le DVG basé sur les objets (polygones) est obtenu d'après le DVG basé sur les éléments (segments, sommets) ; le DVG basé sur les éléments est composé du DVG pour l'intérieur et du DVG pour l'extérieur des polygones.

Pour construire le DVG de l'intérieur d'un polygone, nous avons modifié l'algorithme de Lee [LEE 82a] pour lequel des contre-exemples ont été rencontrés. A partir du concept de visibilité, nous proposons une condition correcte d'arrêt du processus de la fusion de deux DVGs de chaînes.

Pour construire le DVG de l'extérieur d'un ensemble de K polygones, nous avons implémenté un algorithme d'incrémental combinée avec "Divide-and-Conquer" dans le cas le plus défavorable en $O(\sum_{i=1}^K n_i \log n_i + \sum_{h=2}^K \sum_{i=1}^h n_i)$ où n_i est le nombre des éléments du i -ème polygone. Une approche de calcul d'un séparateur en temps linéaire à chaque étape d'incrémental est proposée. Cette approche constitue la clé de nos algorithmes. Afin de construire efficacement le séparateur, nous avons aussi proposé un algorithme pour trouver en même temps l'enveloppe convexe et l'information d'inclusion de deux polygones convexes d'*intersection simple*.

En développant *a priori* la forme d'un séparateur et en déterminant le nombre des branches (courbes connexes) du séparateur, nous proposons une solution en temps linéaire pour la construction semi-dynamique qui réalise une insertion aléatoire d'un polygone convexe dans un DVG en construction évolutive.

Un exemple des applications du DVG est ainsi présenté. Nous montrons que les squelettes continus de formes polygonales, en analyse de formes, sont des caractéristiques inhérentes au DVG. D'une part, les *axes médians* des formes sont un sous-diagramme du DVG à l'échelle d'éléments pour l'intérieur des formes. D'autre part, les exosquelettes des formes sont aussi un sous-diagramme du DVG de l'extérieur des formes à l'échelle d'éléments, et le SKIZ de formes est identique au graphe du DVG basé sur les objets des formes polygonales.

Mots clés : Géométrie Algorithmique, Diagramme de Voronoï généralisé, Divide-and-Conquer, Incrémental, Complexité, Enveloppe convexe, Axe médian, Exosquelette, SKIZ