

Eléments de Géométrie Discrète Vers une Etude des Structures Discrètes Bruitées

MÉMOIRE

Habilitation à diriger des recherches

présentée et soutenue publiquement le 13 décembre 2007

Université Henri Poincaré – Nancy I
(Spécialité Informatique)

par

Isabelle Debled-Rennesson

Composition du jury

Président : Karl Tombre, Professeur, Ecole des Mines de Nancy, Directeur du LORIA

Rapporteurs : Annick Montanvert, GIPSA-lab, Professeur, Université de Grenoble II
Eric Andres, SIC, Professeur, Université de Poitiers
Achille Braquelaire, LABRI, Professeur, Université de Bordeaux I

Examineurs : Jean-Pierre Reveillès, LAIC, Professeur, Université de Clermont-Ferrand
Philippe Even, LORIA, Professeur, Université Nancy I
Paul Zimmermann, LORIA, Directeur de recherche à l'INRIA

Mis en page avec la classe thloria.

Remerciements

Un grand merci à Annick Montanvert, Eric Andres et Achille Braquelaire qui ont accepté d'être rapporteurs de ce travail. Les conseils, les commentaires et les encouragements qu'ils m'ont apportés m'ont été d'une aide précieuse. Tous les trois ont contribué à l'essor de la géométrie discrète au niveau national et international et j'ai été très honorée de leur participation à mon jury d'HDR.

Je remercie Jean-Pierre Reveillès qui, plusieurs années après l'encadrement de ma thèse, a accepté de faire partie de mon jury d'HDR, il m'a donné le goût de la recherche et m'a initiée à la géométrie discrète. Ses travaux ont profondément influencé mon parcours de recherche.

Mes profonds remerciements à Paul Zimmermann pour m'avoir accueillie dans l'équipe POLKA à mon arrivée au LORIA, pour s'être intéressé à mon travail, pour m'avoir encouragée à poursuivre mes travaux en géométrie discrète et enfin pour avoir accepté d'être mon parrain scientifique et faire partie de mon jury d'HDR.

Je remercie Karl Tombre qui a accepté de présider mon jury d'habilitation malgré sa charge de travail importante en tant que directeur du LORIA.

Je remercie aussi Philippe Even pour sa participation à mon jury et pour l'intérêt qu'il a porté à mes travaux.

Le travail de recherche présenté dans ce mémoire est le fruit de nombreuses collaborations, d'échanges et de rencontres. Merci à tous ceux qui y ont contribué et qui sont cités dans ce mémoire. Des remerciements particuliers à Jocelyne Rouyer et Jean-Luc Rémy qui, à mon arrivée au LORIA, se sont intéressés à la géométrie discrète et sont devenus de précieux partenaires de travail.

L'encadrement d'étudiants en thèse est une expérience particulièrement enrichissante, je tiens à remercier les doctorants avec qui j'ai eu la chance de travailler et dont les travaux sont relatés dans ce mémoire : Hélène Dörksen-Reiter, Fabrice Touzain, Laurent Provot, Thanh Phuong Nguyen et Jean-Pierre Salmon.

Je remercie aussi bien sûr tous les membres de l'équipe ADAGIo pour leur soutien, pour leurs collaborations scientifiques, pour leur adhésion dans les différents projets cités dans ce mémoire ainsi que pour les échanges fructueux que nous avons, scientifiques ou non.

Merci encore à Eric Domenjoud, Jean-Luc Rémy et Laurent Wendling pour la relecture de ce mémoire.

Enfin, je tiens à remercier Jean-Loup pour son soutien constant et ses relectures attentives ainsi que nos enfants, Matthieu et Arthur, pour avoir accepté mon manque de disponibilité pendant l'écriture de ce mémoire ...

Table des matières

I	Résumé des activités	3
1	CURRICULUM VITÆ	5
2	Animation et rayonnement scientifique	7
2.1	Participation à la création de deux équipes du LORIA	7
2.2	Collaborations et projets scientifiques	8
2.2.1	Au niveau national	8
2.2.2	Au niveau local	8
2.3	Activités scientifiques diverses	9
2.4	Responsabilités collectives au LORIA	10
3	Bilan des encadrements	11
3.1	Participation à l'encadrement de thèses	11
3.2	Encadrement de stages de DEA/Master2	11
3.3	Autres	12
4	Activités liées à l'enseignement	15
4.1	Enseignements effectués	15
4.1.1	IUFM de Lorraine	15
4.1.2	Université Henri Poincaré, Nancy I	16
4.1.3	Université Nancy 2	17
4.1.4	Université Robert Schuman, Strasbourg III	17
4.1.5	Université Louis Pasteur, Strasbourg I	17
4.2	Responsabilités pédagogiques et administratives	17
4.2.1	Responsabilités à l'IUFM de Lorraine	17
4.2.2	Participation aux commissions de recrutement	18
4.2.3	Organisation d'une université d'été	18

II	Mémoire scientifique	19
	Introduction	21
1	Eléments pour l'analyse de courbes et surfaces régulières	25
1.1	Introduction	25
1.2	Décomposition polygonale de courbes discrètes en parties convexes et concaves	29
1.2.1	Convexité discrète	29
1.2.2	Décomposition polygonale	32
1.2.3	Perspectives	36
1.3	Estimation des normales à une surface discrète	38
1.3.1	Calcul des normales 3D en chaque point du bord d'un objet voxel . .	39
1.3.2	Calcul des normales 3D en chaque surfel du bord d'un objet voxel . .	42
1.3.3	Résultats	43
1.3.4	Perspectives	44
1.4	Reconnaissance de morceaux de plans discrets	46
1.4.1	Définitions et premières propriétés	46
1.4.2	Algorithme de reconnaissance	48
1.4.3	Perspectives	50
1.5	Paraboles discrètes	51
1.5.1	Définition	51
1.5.2	Etude de la structure des paraboles discrètes arithmétiques	52
1.5.3	Algorithme de construction de $P(f, \alpha, 1)$	54
1.5.4	Perspectives	55
1.6	Conclusion	55
2	Eléments de géométrie discrète pour l'étude des courbes bruitées	57
2.1	Introduction	57
2.1.1	Le bruit dans les images	58
2.1.2	Limites des outils existants de géométrie discrète	60
2.2	Segments flous arithmétiques	61
2.2.1	Approche arithmétique	62
2.2.2	Approche géométrique	68
2.2.3	Approximation polygonale multi-niveaux de courbes discrètes	74
2.2.4	Extension en dimension 3	78
2.3	Courbure discrète et autres paramètres géométriques	80
2.3.1	Tangente discrète	80
2.3.2	Courbure discrète	84

2.3.3	Segments flous maximaux et optimisation du calcul de la courbure . . .	87
2.3.4	Application à la détection de cercles et arcs de cercles dans des images techniques	90
2.3.5	Estimateur de la longueur d'une courbe discrète bruitée	94
2.3.6	Extension en dimension 3	95
2.4	Conclusion	97
3	Eléments de géométrie discrète pour l'étude des surfaces discrètes bruitées	99
3.1	Introduction	99
3.2	Morceaux flous de plans discrets	101
3.2.1	Définitions	101
3.2.2	Reconnaissance incrémentale	102
3.3	Extraction de paramètres géométriques	106
3.3.1	Patch d'épaisseur ν centré en un point	106
3.3.2	Normale d'épaisseur ν en un point	107
3.3.3	Aire d'un patch d'épaisseur ν	108
3.3.4	Estimateur de forme	109
3.4	Segmentation en morceaux flous de plans discrets	113
3.5	Conclusion	115
4	Applications en bioinformatique	117
4.1	Introduction	117
4.2	Courbure de l'ADN	117
4.3	Identification et classification des promoteurs	120
4.4	Conclusion	122
	Conclusion et perspectives de recherche	123
	Bibliographie	127
5	Liste des publications	137
III	Sélection d'articles de recherche	143

Liste des Algorithmes

1	Algorithme testant la convexité de la courbe inférieure associée au bord d'un polyomino dans le premier quadrant	31
2	Algorithme déterminant la représentation polygonale supérieure d'une courbe discrète concave du premier octant	35
3	Algorithme de recherche du plus haut triangle	49
4	Algorithme incrémental de recherche du plus haut triangle	50
5	Algorithme incrémental de construction de $P(f, \alpha, 1)$	55
6	Algorithme pour <i>bruiter</i> une courbe discrète	59
7	Reconnaissance arithmétique des segments flous	66
8	Reconnaissance incrémentale d'un segment flou d'épaisseur ν dans le premier octant	71
9	Algorithme de recherche de points multi-niveaux	76
10	Algorithme de reconnaissance d'un segment flou 3D d'épaisseur ν	79
11	Algorithme de segmentation d'une courbe discrète en segments flous 3D d'épaisseur ν	80
12	Algorithme de reconnaissance d'un segment flou centré du premier octant	82
13	Algorithme de recherche des segments flous maximaux d'épaisseur ν d'une courbe C	88
14	Estimateur de la courbure d'épaisseur ν en chaque point de C	89
15	Etape de la reconnaissance incrémentale d'un morceau flou de plan discret d'épaisseur ν	105

Première partie

Résumé des activités

Chapitre 1

CURRICULUM VITÆ

Nom : *DEBLED épouse RENNESSON* Prénom(s) : *Isabelle, Marie*
Date et lieu de naissance : *3 Décembre 1963 à Cherbourg (50)*
Situation de famille : *mariée, deux enfants* Nationalité : *française*
Adresse professionnelle : *LORIA, Campus Scientifique, BP 239,*
54506 Vandoeuvre-les-Nancy
Téléphone : *03 83 59 30 37* Email : *debled@loria.fr*

Titres universitaires français

1995 Doctorat en informatique,
“Etude et reconnaissance des droites et plans discrets” (Géométrie discrète et imagerie),
15 décembre 1995, Université Louis Pasteur (ULP), Strasbourg, Directeur : Professeur
Jean-Pierre Reveillès, mention très honorable avec félicitations du jury,
1992 Diplôme d'études approfondies d'informatique, avec options imagerie, modélisation et
synthèse d'images, ULP,
1991 Maîtrise d'informatique, ULP,
1990 Licence d'informatique, ULP,
1986 Licence de mathématiques, ULP.

Qualifications

1988 CAPES pratique de mathématiques,
1987 CAPES théorique de mathématiques.

Emplois

De septembre 2003 à août 2005 : *Détachement à l'INRIA Lorraine (Nancy)*
Depuis septembre 1997 : *Maître de conférences à l'IUFM de Lorraine (Nancy).*
De septembre 1996 à août 1997 : *PRCE à l'IUT de l'Université Louis Pasteur*
(Strasbourg I).
De septembre 1992 à août 1996 : *ATER au Département d'Informatique de l'IUT*
de l'Université Robert Schuman (Strasbourg III).
De septembre 1987 à août 1992 : *Professeure certifiée de mathématiques.*

Activités d'enseignement universitaire

Depuis septembre 1992, j'ai enseigné dans 5 établissements universitaires et je suis intervenue dans plusieurs cours sur l'algorithmique et les langages de programmation, ainsi que l'architecture des ordinateurs, les systèmes d'exploitation, les réseaux et les notions de base en informatique constituant le C2I. Plus particulièrement, depuis ma nomination comme maître de conférences à l'IUFM de Lorraine, j'y intervins en formation initiale et en formation continue ainsi qu'à l'Université Henri Poincaré en master première année Science de la Vie et de la Santé GM (Génétique Moléculaire) et en master Information Numérique en Entreprise CCI (Compétences Complémentaires en Informatique) et à l'Université Nancy 2 en première année (L1). Enfin, plus récemment, je suis aussi intervenue au département informatique de l'IUT de Saint Dié. Les détails des enseignements réalisés sont donnés dans le chapitre 4.

Activités de recherche

De 1992 à 1998, mes activités de recherche se sont déroulées au Laboratoire des Sciences de l'Image, de l'Informatique et de la Télédétection (LSIIT) à Strasbourg.

Depuis 1998, j'effectue mon travail de recherche au Laboratoire Lorrain de Recherche en Informatique et ses Applications (LORIA) d'abord dans l'équipe POLKA (responsable Paul Zimmermann) puis dans l'équipe ADAGE (responsable Grégory Kucherov) de 2001 à 2005 et enfin depuis janvier 2006 dans l'équipe ADAGIo dont je suis la responsable.

Dans ce cadre, j'ai développé plusieurs travaux qui ont donné lieu à des encadrements d'étudiants et des collaborations qui sont décrits dans les chapitres suivants.

Chapitre 2

Animation et rayonnement scientifique

Sommaire

2.1	Participation à la création de deux équipes du LORIA	7
2.2	Collaborations et projets scientifiques	8
2.2.1	Au niveau national	8
2.2.2	Au niveau local	8
2.3	Activités scientifiques diverses	9
2.4	Responsabilités collectives au LORIA	10

2.1 Participation à la création de deux équipes du LORIA

Depuis mon arrivée au LORIA, à la suite de mon recrutement comme maître de conférences à l'IUFM de Lorraine, j'ai été membre de 3 équipes :

- Equipe PolKA : octobre 1998 - décembre 2000
- Equipe ADAGE : janvier 2001 - décembre 2005
- Equipe ADAGIo : depuis janvier 2006

J'ai participé activement à la création des équipes ADAGE et ADAGIo.

L'équipe ADAGE (Algorithmique Discrète et ses Applications à la GENomique) du LORIA, dirigée par Gregory Kucherov, a été créée en janvier 2001. L'objectif de cette équipe était la mise au point d'algorithmes efficaces sur les structures discrètes. Ces algorithmes reposent souvent sur des études théoriques profondes de propriétés combinatoires de ces structures. Le champ d'applications privilégié de ces travaux est la bioinformatique, domaine dans lequel les modèles discrets apparaissent de façon naturelle et essentielle.

A la suite, d'une part, de la promotion de Grégory Kucherov comme directeur de recherche au LIFL à Lille, et d'autre part, du recrutement d'un collègue en géométrie discrète et du rapprochement scientifique avec plusieurs autres collègues du LORIA, nous avons décidé de créer une nouvelle équipe dans le même esprit qu'ADAGE mais avec l'imagerie comme champ

d'applications supplémentaire. L'équipe ADAGIo¹(Algorithmique Discrète et ses Applications à la Génomique et à l'Imagerie), a été créée en janvier 2006. J'ai piloté la rédaction du projet de l'équipe et j'en assure depuis la direction. L'équipe, à sa création, était composée de 5 permanents, deux doctorants et un post-doctorant.

2.2 Collaborations et projets scientifiques

2.2.1 Au niveau national

Dans le cadre de la vie scientifique nationale, j'ai participé à plusieurs actions qui ont donné lieu à différentes collaborations :

- Action Spécifique *Géométrie Discrète et Géométrie Algorithmique* pour l'année 2004.
- ACI IMPBio dans le cadre du projet FouDAnGA (Fouille de données pour l'annotation de génomes d'actinomycètes)², de 2004 à 2006.

Pilotage d'un projet blanc de l'ANR

Le projet (GEODIB), déposé en mars 2006 dans le cadre des appels à projets blancs de l'ANR, a été accepté et je me charge de sa coordination. Ce projet regroupe des membres de la communauté française de géométrie discrète, issus de 4 laboratoires (LABRI, LIRIS, LLAIC, LORIA), qui souhaitent développer une *Géométrie des Objets Discrets Bruités* en utilisant de nouvelles techniques issues de la géométrie discrète et en se détachant des méthodes classiques comme la logique floue ou encore les techniques de filtrage utilisées en traitement d'image. Plus spécifiquement, les membres de ce projet s'intéressent à l'étude des courbes et surfaces discrètes bruitées correspondant aux contours et bords d'objets discrets. Notre objectif est d'élaborer, dans le cadre de la géométrie discrète, un formalisme adapté à ces objets tenant compte du bruit inhérent aux outils et méthodes d'acquisition tout en contrôlant les approximations effectuées. Plusieurs façons d'aborder l'étude de la géométrie discrète ont été développées, nous utilisons dans ce projet le point de vue arithmétique définissant analytiquement les primitives discrètes.

Ce projet de recherche se déroule sur quatre années (de novembre 2006 à novembre 2010) pendant lesquelles des rencontres régulières auront lieu entre les quatre équipes impliquées (au moins 3 par an). Nous envisageons aussi d'organiser des journées nationales sur le thème Géométrie des objets discrets bruités où des participants extérieurs intéressés par ce sujet seront invités.

2.2.2 Au niveau local

Localement dans le cadre du Contrat de Plan Etat Région 2000-2006, l'Etat et la Région Lorraine ont créé 6 Pôles de Recherche Scientifique et Technologique et notre équipe a participé au pôle *Intelligence Logicielle* dans le thème de recherche *Bioinformatique et applications à la génomique*. Avec des membres de l'équipe ADAGE, j'ai participé à une opération s'intéressant à la détection des motifs de promoteurs dans le génome des Streptomyces. Ce travail a été mené en

¹<http://www.loria.fr/equipes/adagio>

²http://impbio.lirmm.fr/PROJETS_ACCEPTES_2004/76.html

collaboration avec le Laboratoire de Génétique et de Microbiologie de l'Université Henri Poincaré ainsi que des membres de l'équipe Orpailleur du Loria.

Dans le cadre d'un nouveau CPER débutant en 2007, je participe au projet *Modélisations, informations et systèmes numériques* dans une action du thème *Modélisation des Biomolécules et de leurs Interactions* sur l'étude de la modélisation des assemblages protéines-protéines. Ce sujet de recherche est mené en collaboration avec des membres de l'équipe Orpailleur du Loria.

2.3 Activités scientifiques diverses

Au niveau international, mon expertise a été demandée à plusieurs occasions en tant que :

- Relectrice d'articles pour un numéro spécial de LNCS sur "Digital and Image Geometry" ainsi que pour les revues Computer Graphics Forum, Theoretical Computer Science, Applied Soft Computing Journal et Journal of Discrete Algorithm.
- Membre du *comité de programme* des conférences :
 - 10 et 11ème International Workshop on Combinatorial Image Analysis (IWZIA'04 et IWZIA'06),
 - 12, 13 et 14ème International Conference on Discrete Geometry for Computer Imagery (DGCI'05, DGCI'06 et DGCI'08),
 - 2nd International Symposium on Visual Computing (ISVC'06).

De plus, j'ai participé ou participerai à quatre *jurys de thèse* :

- Membre du jury de thèse de Xavier Hilaire, de l'équipe QGAR du Loria, cette thèse a été soutenue le 15 janvier 2004.
- Membre du jury de thèse d'Hélène Dörksen-Reiter, de l'Université de Hambourg, cette thèse a été soutenue le 13 novembre 2004.
- Membre du jury de thèse de Damien Jamet, de l'Université de Montpellier, cette thèse a été soutenue le 13 décembre 2005.
- Membre du jury de thèse de Fabrice Touzain, de l'équipe ADAGio du Loria, cette thèse a été soutenue le 15 novembre 2007.

Au niveau national, d'octobre 2003 à octobre 2007, j'ai été membre élue du *conseil national des universités (CNU)* pour la section 27. De septembre 2004 à octobre 2007, j'ai aussi été *membre du bureau* et *seconde vice-présidente* du CNU de la section 27.

Je suis aussi membre du comité technique IAPR sur la géométrie discrète (TC18)³. Dans le cadre du GDR Informatique Mathématique, je participe au groupe de travail Géométrie discrète.

³<http://www.cb.uu.se/TC18/>

Enfin, j'ai participé à la rédaction de deux chapitres d'un ouvrage collectif sur la géométrie discrète paru en 2007 dont les éditeurs sont D. Coeurjolly, A. Montanvert, et JM. Chassery aux éditions Hermès.

2.4 Responsabilités collectives au LORIA

Je suis responsable de l'équipe ADAGIo depuis janvier 2006 et à ce titre, je suis membre de l'*assemblée des responsables d'équipe* du LORIA qui se réunit chaque mois. J'ai été ou suis toujours membre de diverses commissions du LORIA :

- Membre de 2000 à 2006 de la *commission documentation*,
- Membre depuis 2005 de la *commission hygiène et sécurité*,
- Membre en 2006 de la *commission personnel*, chargée d'examiner les candidatures des doctorants et post-doctorants,
- Membre en 2006 et 2007 du *comité des opérations du thème Bioinformatique et applications à la génomique* du PRST *Intelligence Logicielle*,
- Membre depuis 2007 des *comités des opérations du thème Modélisation des Biomolécules et de leurs Interactions ainsi que du thème Modélisation, Interaction et Simulation* du CPER débuté en 2007.

J'ai fait partie du comité local d'organisation du workshop *Discrete Tomography and related Problems*⁴ s'étant déroulé du 22 au 26 Mars 1999 au Château de Volkrange (à 90 km de Nancy).

⁴<http://dmtcs.loria.fr/tomo/>

Chapitre 3

Bilan des encadrements

Sommaire

3.1	Participation à l'encadrement de thèses	11
3.2	Encadrement de stages de DEA/Master2	11
3.3	Autres	12

3.1 Participation à l'encadrement de thèses

- De septembre 2001 à novembre 2004, participation à l'encadrement de la thèse (20%) de **Hélène Reiter**, étudiante à l'Université de Hambourg. Titre du travail : *Shape representations of digital sets based on convexity properties*. Directeur de thèse : Ulrich Eckhardt (Professeur, Université de Hambourg). Publications et interventions communes : 5, 12, 35 (cf. chapitre 5, partie II). Situation actuelle : post-doctorat à Hambourg.
- Depuis octobre 2003, participation à l'encadrement de la thèse (25%) de **Fabrice Touzain**. Titre du travail : *Analyse de signaux de transcription dans un génome bactérien*. Co-directeurs : Gregory Kucherov (DR CNRS, LIFL, 25%), Pierre Leblond (Professeur UHP, 25%), Bertrand Aigle (MC UHP, 25%). Soutenance prévue le 15 novembre 2007. Publications et interventions communes : 25, 32, 44 (cf. chapitre 5, partie II).
- Depuis octobre 2005, co-encadrement principal de la thèse de **Laurent Provot**. Titre du travail : *Polyédrisation d'objets discrets bruités*. Directeur de thèse : Philippe Even (Professeur UHP). Publication commune : 10, 46, 47 (cf. chapitre 5, partie II).
- Depuis décembre 2006, co-encadrement principal de la thèse de **Thanh Phuong Nguyen**. Titre du travail : *Courbure discrète et applications*. Directeur de thèse : Philippe Even (Professeur UHP). Publication commune : 8 (cf. chapitre 5, partie II).

3.2 Encadrement de stages de DEA/Master2

- Février 1997 - juin 1997 : co-encadrement de **Yannick Kirschhoffer** avec Pierre Tellier (LSIIT, Strasbourg). Titre du travail : *Estimation de normales dans un espace discret 3D*,

analyse comparative de méthodes. Situation actuelle : non connue.

- Février 2003 - juin 2003 : co-encadrement de **Fabrice Touzain** avec Gregory Kucherov. Titre du travail : *Recherche des sites de régulation de la transcription chez Streptomyces coelicolor A3(2)*. Situation actuelle : thèse en cours (Loria, Nancy).
- Février 2004 - août 2004 : encadrement de **Franck Rapaport**. Titre du travail : *Calcul du rayon de courbure en chaque nucléotide d'une séquence d'ADN*. Situation actuelle : thèse en cours (Institut Pasteur).
- Février 2005 - juin 2005 : encadrement de **Laurent Provot**. Titre du travail : *Reconnaissance de morceaux bruités de plans discrets*. Situation actuelle : thèse en cours (Loria, Nancy).
- Mars 2005 - août 2005 : encadrement de **Thanh Phuong Nguyen**. Titre du travail : *Optimisation du calcul de la courbure de l'ADN*. Situation actuelle : thèse en cours (Loria, Nancy).

3.3 Autres

- De juin 1999 à juillet 1999, co-encadrement d'**Isabelle Sivignon** avec Alexander Bookmayr. Diplôme : 1ère année ENS, Lyon. Titre du travail : *Reconnaissance de polyominos convexes par programmation par contraintes*. Situation actuelle : Chargée de Recherches CNRS.
- De mai 2001 à septembre 2001, co-encadrement de **Alexei Stanger** avec Grégory Kucherov et Gilles Schaeffer. Diplôme : magistère d'informatique de l'Université de Paris 6. Titre du travail : *Recherche des signaux de transcription dans un génome bactérien avec RMES*. Situation actuelle : non connue.
- De novembre 2001 à juillet 2003, co-encadrement de **Patricia Lavigne** avec Grégory Kucherov. Diplôme : DESS EGOIST (en alternance). Titre du travail : *Identification dans le génome de Streptomyces coelicolor A3(2) des sites de fixation aux facteurs de transcription*. Situation actuelle : dans l'industrie.
- De juin 2003 à septembre 2003, encadrement de **Sandra Luis**. Diplôme : DESS CCI. Titre du travail : *Recherche des signaux de transcription dans un génome bactérien*. Situation actuelle : dans l'industrie.
- De juin 2004 à août 2004, encadrement de **Nguyen Thanh-Trung**. Diplôme : magistère de Mathématiques Fondamentales & Appliquées et d'Informatique de la Région Parisienne. Titre du travail : *Polygonalisation de courbes bruitées*. Situation actuelle : thèse en cours (INRIA Sophia).
- De décembre 2006 à août 2006, encadrement de **Damien Jamet** en post-doctorat dans l'équipe ADAGIo. Titre du travail : *Courbure de l'ADN*. Situation actuelle : maître de

conférences (IUT St Dié des Vosges).

J'ai également encadré ou co-encadré plusieurs groupes d'étudiants de l'ESIAL (Ecole Supérieure d'Informatique et Applications de Lorraine) dans le cadre du module d'initiation à la recherche (en seconde année) sur des sujets de géométrie discrète.

En bioinformatique, j'ai aussi encadré des étudiants dans le cadre de stages de 3 à 6 mois en Master première année.

Chapitre 4

Activités liées à l'enseignement

Sommaire

4.1 Enseignements effectués	15
4.1.1 IUFM de Lorraine	15
4.1.2 Université Henri Poincaré, Nancy I	16
4.1.3 Université Nancy 2	17
4.1.4 Université Robert Schuman, Strasbourg III	17
4.1.5 Université Louis Pasteur, Strasbourg I	17
4.2 Responsabilités pédagogiques et administratives	17
4.2.1 Responsabilités à l'IUFM de Lorraine	17
4.2.2 Participation aux commissions de recrutement	18
4.2.3 Organisation d'une université d'été	18

4.1 Enseignements effectués

4.1.1 IUFM de Lorraine

Formation initiale

Maître de conférences à l'IUFM de Lorraine, je suis chargée de la formation des jeunes enseignants (professeurs stagiaires des lycées et collèges) à l'informatique. A mon arrivée à l'IUFM en septembre 1997, seules des formations au traitement de texte étaient organisées. J'ai donc mis en place un plan de formation à l'informatique, j'ai conçu les formations citées ci-dessous dont le volume horaire varie de 6 à 15 heures et j'ai élaboré les supports de cours correspondants. Chaque étudiant choisit son contenu de formation, en fonction de ses compétences et de ses objectifs. J'ai assuré les enseignements suivants pour des groupes de 10 à 20 étudiants :

- *Traitement des images*
- *Utilisation d'Internet (messagerie, moteurs de recherche)*
- *Création de pages HTML*
- *Initiation à JavaScript*
- *Initiation à Java*
- *Outils pour l'enseignant (traitement de texte, tableur, système d'exploitation,...)*
- *Perfectionnement tableur, initiation à Visual Basic*
- *Découverte de LINUX*

– *Initiation à \LaTeX*

D'autre part, dans chaque discipline, des formations aux logiciels spécifiques à ces disciplines sont proposées, je suis intervenue de 1997 à 2000 dans la formation des enseignants de mathématiques :

- Formation aux *logiciels de mathématiques* (30 heures) : les logiciels de construction géométrique comme Cabri II, GeoplanW, Calque, le logiciel de calcul formel Derive sont présentés ainsi que les utilisations possibles d'un tableur en classe de mathématiques.
- *Introduction au calcul formel avec le logiciel Maple* (12 heures).

Formation continue

Jusqu'en 2004, l'IUFM était chargé de l'organisation de la formation continue des enseignants et j'y suis intervenue dans 2 contextes différents :

- dans le cadre de la formation des animateurs TICE, enseignants déchargés pour suivre une formation lourde en informatique, j'ai assuré pendant 2 années les cours et TP du module de 48 heures *le langage de programmation JAVA* qui a abouti à la conception et à la réalisation, par chaque stagiaire, d'une application à usage pédagogique ;
- dans le cadre du plan académique de formation, pendant 3 années, j'ai pris en charge 3 formations de 24 heures de ce plan :
 - *Initiation au langage de programmation JAVA,*
 - *Compléments sur le langage de programmation JAVA,*
 - *Interactivité dans les pages web avec le langage JavaScript.*

Groupes de recherche action

J'ai piloté pendant 2 années (2001-2003) un groupe de 6 personnes chargé de réfléchir à la mise en place de dispositifs d'autoformation en TICE.

4.1.2 Université Henri Poincaré, Nancy I

Depuis 2000, j'enseigne chaque année en DESS Compétences Complémentaires en Informatique devenu ensuite master Information Numérique en Entreprise CCI dans l'UE *algorithmique et programmation java* (60 heures : cours, TD, TP). Depuis 2005, cet enseignement est réalisé dans le cadre du LMD en commun avec le master première année Science de la Vie et de la Santé option Génétique Moléculaire pour lequel je participe à l'enseignement d'un cours de programmation web (15 heures) : *langage HTML et JavaScript*. Je suis aussi membre du jury de master première année SVS GM.

En 2006/2007, j'ai pris en charge l'UE *Interfaces Graphiques en JAVA* à l'IUT de St Dié. Cet enseignement comporte 24 heures de cours, TD et TP.

4.1.3 Université Nancy 2

Le C2I niveau 2 *enseignant* est mis en œuvre à l'IUFM et les étudiants doivent acquérir le C2I niveau 1 à l'université. Afin de mieux adapter les enseignements en informatique proposés à l'IUFM de Lorraine, je participe depuis 2005 aux *enseignements d'informatique de L1* à l'université Nancy 2 (2 UE de 18 heures) à l'issue desquels les étudiants ont la possibilité de passer l'examen du *C2I*. Je participe à son organisation et je fais partie du jury. De plus, dans cet enseignement, prodigué dans toutes les sections de L1, il est nécessaire de coordonner les différents enseignants et, avec cet objectif, je participe au tutorat des nouveaux enseignants intervenant dans ce cadre.

4.1.4 Université Robert Schuman, Strasbourg III

Durant quatre années, de septembre 1992 à septembre 1996 j'ai exercé la fonction d'A.T.E.R. au Département d'Informatique de l'I.U.T. de l'Université Robert Schuman. Je me suis investie dans deux unités de valeur pour lesquelles j'ai assuré les travaux dirigés et les travaux pratiques :

- *Algorithmique fondamentale, langage C, C++* (80 heures)
- *Architecture et téléinformatique* (40 + 60 heures)

4.1.5 Université Louis Pasteur, Strasbourg I

Recrutée comme P.R.C.E. à l'I.U.T. de l'Université Louis Pasteur, pendant l'année 1996-1997, j'y ai enseigné l'informatique dans trois départements (Maintenance Industrielle, Mesure Physique, Génie Electrique et Informatique Industrielle) en adaptant cet enseignement au profil de chacune de ces formations professionnelles :

- *Systeme Unix et Langage C* (80 heures)
- *Algorithmique et Langage Pascal* (72 heures)
- *Bureautique* (96 heures)

4.2 Responsabilités pédagogiques et administratives

4.2.1 Responsabilités à l'IUFM de Lorraine

- De 1997 à 2003, *responsable des formations en informatique* des professeurs stagiaires des lycées et collèges de l'IUFM de Lorraine (environ 500 stagiaires sur les deux sites de Nancy et Metz). J'ai coordonné l'organisation des formations en début d'année scolaire, ce qui implique de recenser les choix des stagiaires parmi les modules proposés et ensuite d'organiser le planning des formations et recruter éventuellement des enseignants. Ce travail est réalisé en collaboration avec les formateurs informatiques (enseignants du second degré) des deux sites. De plus, chaque année, j'ai animé des réunions de concertation, dans lesquelles le plan de formation en informatique est réorganisé en fonction des besoins des enseignants stagiaires, qui évoluent très rapidement. J'ai écrit le chapitre TICE de deux projets d'établissement de l'IUFM en 1998 et 2002.

- **Chargée de mission** en informatique pédagogique pour la formation initiale à l'IUFM de Lorraine auprès du directeur adjoint J-C Roth de septembre 2000 à août 2003 (arrêt depuis détachement). En plus des tâches citées précédemment, j'ai assuré la responsabilité de la coordination des enseignements en TICE pour les professeurs des écoles stagiaires ainsi que les missions suivantes :
 - Représentante de l'IUFM dans des manifestations diverses où j'ai présenté les formations TICE de l'IUFM de Lorraine :
 - + Intervention aux journées Specif de Montpellier en septembre 2000
 - + Intervention à la manifestation "Vers l'école en réseau", Metz, novembre 2002.
 - Représentante de l'IUFM dans les réunions de coordination avec le rectorat et les universités de Nancy sur les actions liées aux nouvelles technologies de l'information et de la communication.
 - Participation aux réunions concernant les achats de matériels informatique de l'ensemble des sites de l'IUFM ainsi que la gestion des salles informatiques.
- Pilote de 1998 à 2001 de la mise en place du site internet de l'IUFM de Lorraine⁵, j'ai constitué un groupe de travail qui a défini ses orientations et ses contenus.
- Responsable de l'encadrement des jeunes docteurs recrutés pour le site de Nancy (4 entre 1998 et 2002).

4.2.2 Participation aux commissions de recrutement

- Membre titulaire de la **commission de spécialistes** scientifique de l'IUFM de Lorraine de septembre 2000 à août 2003 (arrêt en raison de mon détachement).
- Membre suppléante de la **commission de spécialistes** de la section 27 de l'université de Reims de septembre 2001 à août 2003 (arrêt en raison de mon détachement).
- Membre de la commission de recrutement des jeunes docteurs de l'IUFM de Lorraine de 1998 à 2000 (6 jeunes docteurs recrutés).
- Membre de la commission de recrutement des PRAG et PRCE en TICE de l'IUFM de Lorraine de 1998 à 2003 (arrêt en raison de mon détachement).

4.2.3 Organisation d'une université d'été

- Responsable pédagogique de l'université d'été "*Maîtriser des évolutions en informatique pour l'intégration des technologies de l'information et de la communication dans l'enseignement*" qui a eu lieu les 3, 4 et 5 novembre 1999 et a réuni 45 personnes. J'ai pris en charge toute l'organisation pratique du stage (repas, hébergement, réservation de salles, de machines, ...) et une grosse partie de l'organisation pédagogique (recrutement des intervenants, réunions de coordination, ...).

⁵<http://www.lorraine.iufm.fr>

Deuxième partie

Mémoire scientifique

Introduction

Depuis 1992, j'ai d'abord effectué mes activités de recherche au LSIIT (Laboratoire des Sciences de l'Image, de l'Informatique et de la Télédétection, UMR 7005) à Strasbourg dans le cadre de ma thèse puis au LORIA (Laboratoire Lorrain de Recherche en Informatique et ses Applications, UMR 7503) après mon recrutement comme maître de conférences à l'IUFM de Lorraine, Nancy.

Les travaux que j'ai entrepris pendant cette période se situent, pour la majorité, dans le cadre de la *géométrie discrète* et, pour quelques-uns dans le cadre de la bioinformatique. J'expose dans ce mémoire les résultats obtenus et les orientations de recherche choisies depuis mon intégration au LORIA. Ceux-ci sont liés aux équipes dans lesquelles j'ai effectué ces travaux ainsi qu'aux collaborations internes engagées avec des membres du Loria et aux collaborations externes avec des chercheurs du domaine de la géométrie discrète.

À mon intégration au Loria, j'ai été accueillie dans l'équipe PolKA (Polynômes, combinatoire, arithmétique), dirigée par Paul Zimmermann. Ce projet avait pour objectif l'étude d'algorithmes sur les nombres, les polynômes et les structures combinatoires, leur implantation efficace, ainsi que leurs applications. Dans ce cadre, j'ai poursuivi le travail engagé durant ma thèse sur les droites et les plans discrets avec des collaborations extérieures. J'ai aussi piloté un groupe de travail sur la géométrie discrète auquel participaient Jocelyne Rouyer et Jean-Luc Rémy, membres de l'équipe PolKA.

En janvier 2001, l'équipe POLKA s'est scindée en deux équipes : SPACES (Systèmes Polynomiaux, Arithmétiques, Calculs Efficaces et Sûrs) et ADAGE (Algorithmique Discrète et ses Applications à la GÉnomique). De 2001 à 2006, j'ai fait partie de l'équipe ADAGE, dirigée par Grégory Kucherov, qui s'intéressait à la mise au point d'algorithmes efficaces sur les structures discrètes. Un des champs d'application de l'équipe étant la bioinformatique, domaine dans lequel les modèles discrets apparaissent de façon naturelle, je me suis investie dans ce domaine et j'ai engagé des travaux dans cette voie. De septembre 2003 à août 2005, j'ai bénéficié d'un détachement à l'INRIA qui m'a permis d'accentuer mes activités de recherche et de nouer des collaborations qui m'ont donné la possibilité d'utiliser certains résultats de géométrie discrète dans des applications en traitement d'image et en bioinformatique. Ceci m'a permis de prendre conscience de l'intérêt de l'élaboration d'une géométrie discrète des objets bruités mieux adaptée aux applications et j'ai alors commencé à travailler dans cette voie.

En janvier 2006, l'équipe ADAGIo (Algorithmique Discrète et ses Applications à la Génomique et à l'Imagerie) a été créée à la suite des départs d'Alexander Bockmayr et Grégory Kucherov, respectivement responsables des équipes MODBIO et ADAGE. Depuis sa création, je suis la responsable de cette nouvelle équipe, ADAGIo, dont le projet se situe dans la continuité

de l'équipe ADAGE avec l'imagerie comme champ d'application supplémentaire et un fort intérêt de tous ses membres pour l'élaboration d'une géométrie discrète des objets discrets bruités.

La rédaction du projet de l'équipe ADAGIo fin 2005 a été l'occasion de mener une réflexion approfondie sur mon projet personnel de recherche. De plus, le pilotage de la rédaction d'un projet blanc de l'ANR, GEODIB (Géométrie Discrète des objets bruités), début 2006, a étayé ce projet dont je donne les lignes directrices dans la conclusion de ce mémoire.

Depuis mon intégration au Loria, la grande majorité de mes travaux a été effectuée en *géo-métrie discrète* dite *analytique* initiée par Jean-Pierre Reveillès avec l'introduction de la définition arithmétique d'une droite discrète en 1991. Les définitions analytiques d'objets discrets permettent de les représenter de manière compacte, d'étudier des objets intrinsèquement discrets (pas uniquement des approximations d'objets continus), et de définir des objets discrets infinis. Cette approche vise à résoudre rigoureusement des problèmes pratiques posés par la non validité de la plupart des théorèmes euclidiens dans un espace discret (par exemple l'intersection de deux droites discrètes). Elle permet également d'éviter les erreurs engendrées par l'utilisation de nombres réels codés en virgule flottante.

Ces quinze dernières années, de nombreux résultats fondés sur cette approche ont vu le jour, ils peuvent être classés dans les catégories énoncées ci-dessous :

- définition et étude de nouvelles classes d'objets discrets (droites 3D, hyperplans, cercles, sphères, simplexes, ...),
- reconnaissance analytique de structures discrètes permettant, non seulement de dire si un ensemble de points est ou non un objet discret donné (segment de droite, morceau de plan, cercle, ...), mais aussi de fournir les coefficients des inéquations analytiques correspondantes,
- extraction de paramètres géométriques sur des structures discrètes, des paramètres globaux comme l'aire, le périmètre ou des paramètres locaux comme les normales, les tangentes ou la courbure,
- reconstruction analytique visant à passer d'une représentation en compréhension du discret à une représentation en compréhension du continu,
- transformations discrètes : applications quasi-affines, rotations, ...
- visualisation utilisant des propriétés des objets discrets telles que l'épaisseur optimale des objets ou encore les normales.

L'application première de ces travaux se trouve en informatique géométrique et graphique, en particulier en synthèse d'image, traitement d'image, analyse de documents et imagerie médicale. Un ouvrage en français sur la géométrie discrète en analyse d'image est paru cette année [CMC07], il est le fruit de la rédaction de plusieurs chercheurs de ce domaine (j'ai participé à l'écriture de deux chapitres) et relate les progrès du domaine effectués depuis 1991, date de la parution d'un premier ouvrage sur ce sujet [CM91], écrit par J.-M. Chassery et A. Montanvert. Des livres en langue anglaise sur ce sujet ont aussi été publiés pendant cette période [KR96, Her98, KR04]. Cette géométrie à l'usage des ordinateurs est aujourd'hui en plein essor comme en témoignent aussi les colloques internationaux du domaine : DGCI (Discrete Geometry for Computer Imagery) et IWCIA (International Workshop on Combinatorial Image Analysis) dont respectivement les 14^{ième} et 12^{ième} éditions auront lieu en 2008.

Les travaux que j'ai réalisés depuis 1997, année de mon recrutement comme maître de conférences à l'IUFM de Lorraine, se situent dans le cadre décrit ci-dessus et plus spécifiquement dans les 4 premiers axes cités : définition d'objets discrets, reconnaissance de structures discrètes, extraction de paramètres géométriques, reconstruction. Je reprends ces axes dans les chapitres

de ce mémoire pour décrire les résultats obtenus ainsi que les collaborations, les projets et les encadrements auxquels j'ai participé.

Le *chapitre* 1 résume les résultats obtenus dans le cadre de l'analyse de courbes et surfaces *régulières*. Mes travaux sur la convexité discrète y sont exposés et sont ensuite utilisés dans le cadre de la recherche d'une représentation polygonale *la plus significative* d'une courbe discrète. Puis une méthode pour calculer les normales en tout point du bord d'un objet discret est proposée, elle repose sur l'étude des coupes de cet objet. Un travail sur la reconnaissance des morceaux de plans discrets naïfs est ensuite présenté, il utilise une approche originale reposant sur l'étude de l'enveloppe convexe de l'ensemble des cordes de l'ensemble testé. Enfin dans la dernière section de ce chapitre, un nouvel objet discret est proposé et étudié : les paraboles discrètes arithmétiques.

Un de mes objectifs est d'élaborer, dans le cadre de la géométrie discrète, un formalisme adapté aux objets discrets bruités qui tienne compte du bruit inhérent aux outils et méthodes d'acquisition. Pour étudier les courbes et les surfaces discrètes bruitées, j'ai proposé dans les *chapitres 2 et 3* les notions de segments flous arithmétiques et de morceaux de plans flous qui sont des généralisations des notions de segments de droites et de morceaux de plans discrets arithmétiques. L'étude des propriétés arithmétiques et géométriques de ces objets a permis d'obtenir des algorithmes de *reconnaissance* et d'extraire des *paramètres géométriques* (périmètre, courbure, normale, aire, ...) sur les courbes et surfaces bruitées. Ces méthodes sont présentées dans ces deux chapitres. Des algorithmes de *reconstruction* d'objets continus correspondant aux structures discrètes étudiées sont aussi proposés.

Dans le *chapitre* 4 une application en bioinformatique sur le calcul de la courbure de l'ADN est présentée ainsi qu'un travail indépendant en génomique. Enfin je termine ce mémoire par mon projet de recherche.

Chapitre 1

Eléments pour l'analyse de courbes et surfaces régulières

Sommaire

1.1	Introduction	25
1.2	Décomposition polygonale de courbes discrètes en parties convexes et concaves	29
1.2.1	Convexité discrète	29
1.2.2	Décomposition polygonale	32
1.2.3	Perspectives	36
1.3	Estimation des normales à une surface discrète	38
1.3.1	Calcul des normales 3D en chaque point du bord d'un objet voxel	39
1.3.2	Calcul des normales 3D en chaque surfel du bord d'un objet voxel	42
1.3.3	Résultats	43
1.3.4	Perspectives	44
1.4	Reconnaissance de morceaux de plans discrets	46
1.4.1	Définitions et premières propriétés	46
1.4.2	Algorithme de reconnaissance	48
1.4.3	Perspectives	50
1.5	Paraboles discrètes	51
1.5.1	Définition	51
1.5.2	Etude de la structure des paraboles discrètes arithmétiques	52
1.5.3	Algorithme de construction de $P(f, \alpha, 1)$	54
1.5.4	Perspectives	55
1.6	Conclusion	55

1.1 Introduction

Les travaux présentés dans ce chapitre concernent l'étude des courbes et surfaces discrètes non bruitées, ils se situent dans la continuité des recherches entreprises pendant ma thèse. Dans la section 1.2 mes travaux sur la convexité discrète sont exposés et sont ensuite utilisés dans le cadre de la recherche d'une représentation polygonale la plus significative d'une courbe discrète. Une partie de ces travaux [6,18] a été réalisée avec Jean-Luc Rémy et Jocelyne Rouyer, deux collègues de l'équipe ADAGE/ADAGIo. La seconde partie de ce travail [5,12,35] a été effectuée avec

Hélène Dörksen-Reiter, doctorante sous la direction du professeur Eckhardt (Université de Hambourg). Les méthodes proposées utilisent l'algorithme de reconnaissance de segments de droites discrètes naïves développé pendant ma thèse (cf. [7] et rappel ci-après), c'est le cas aussi dans la section 1.3 où l'algorithme est utilisé dans le cadre de la recherche de tangentes discrètes 2D pour l'évaluation des normales à la surface d'un objet discret. Cet estimateur de normales [19] a été conçu avec Pierre Tellier (LSIIT, Strasbourg) et Yannick Kirschoffer (LSIIT, Strasbourg, 1997) dans le cadre de son stage de DEA. Mon intérêt pour la reconnaissance des objets discrets n'a jamais faibli et je présente dans la section 1.4 un algorithme de reconnaissance de morceaux de plans discrets naïfs [3] effectué en collaboration avec Yan Gérard (LLAIC, Clermont-Ferrand) et Paul Zimmermann (équipe CACAO, Loria). Il utilise une approche originale reposant sur l'étude de l'enveloppe convexe de l'ensemble des cordes de l'ensemble testé. Enfin dans la dernière section de ce chapitre, l'étude d'un nouvel objet discret, les paraboles discrètes arithmétiques [9], est exposée et a été réalisée avec Eric Domenjoud et Damien Jamet, deux collègues de l'équipe ADAGIo.

Les travaux décrits dans ce chapitre ouvrent des perspectives de recherche qui seront décrites à la fin de chaque section.

Je rappelle rapidement ci-dessous certaines définitions, notations et propriétés élémentaires de géométrie discrète qui sont utilisées dans plusieurs sections de ce mémoire.

Un *point discret* est un élément de \mathbb{Z}^n , pour $n = 2$, on parle aussi de *pixel* et pour $n = 3$ de *voxel*. Un *objet discret* est un ensemble de points discrets.

Il existe deux définitions du *voisinage* entre deux points discrets de \mathbb{Z}^n . La première (cf. [And00]) est générale et s'applique en dimension n :

Définition 1.1 *Deux points discrets distincts $M = (m_1, \dots, m_n)$ et $N = (n_1, \dots, n_n)$ sont dits k -voisins pour $0 \leq k \leq n$, si $|m_i - n_i| \leq 1$ pour $1 \leq i \leq n$ et si $\sum_{i=1}^n |m_i - n_i| \leq n - k$.*

La seconde définition propose une terminologie en dimension 2 et 3. En dimension 2, on parle de *8- et de 4-voisinages* qui correspondent respectivement aux 0- et 1-voisinages de la définition précédente. En dimension 3, on parle de *26-, 18- et de 6-voisinages* qui correspondent respectivement aux 0-, 1- et 2-voisinages.

Définition 1.2 *Un k -chemin dans un objet discret \mathcal{O} est une suite de points discrets de \mathcal{O} telle que deux points consécutifs de cette suite soient k -voisins.*

Un objet discret \mathcal{O} est k -connexe s'il existe un k -chemin entre deux points quelconques de \mathcal{O} .

En 1970, H. Freeman [Fre70] a donné une méthode générale pour coder une courbe discrète 8- ou 4-connexe dans \mathbb{Z}^2 . A chacune des directions principales, on associe un entier α variant de 0 à 7 selon le schéma de la Figure 1.1.

Définition 1.3 *On appelle **code de Freeman** d'une courbe discrète, un point P_0 , premier point de la courbe et une suite (α_i) pour i variant de 0 à n avec α_i la direction permettant le passage de P_i à P_{i+1} . Une **courbe discrète du premier octant** (resp. **premier quadrant**) est une courbe dont le code est constitué uniquement des directions 0 et 1 (resp. 0,1 et 2).*

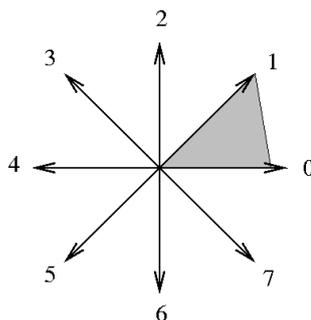


FIG. 1.1 – Directions de Freeman, en gris le premier octant.

Des primitives discrètes ont été étudiées ces vingt dernières années et plus spécifiquement la définition arithmétique d'une droite discrète a été introduite par J.P. Reveillès [Rev91] :

Définition 1.4 Une **droite discrète arithmétique** de borne inférieure μ , d'épaisseur arithmétique ω et de vecteur directeur (b, a) , est l'ensemble des points (x, y) de \mathbb{Z}^2 vérifiant

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

avec a, b, μ, ω entiers, $\text{pgcd}(a, b) = 1$, elle est notée $\mathcal{D}(a, b, \mu, \omega)$.

Les droites **naïves**, notées $\mathcal{D}(a, b, \mu)$, pour lesquelles $\omega = \sup(|a|, |b|)$ et les droites **standards** telles que $\omega = |a| + |b|$ sont les droites discrètes les plus étudiées dans la littérature car elles correspondent aux bords 8- et 4-connexes d'objets discrets.

Définition 1.5 Les droites réelles $ax - by = \mu$ et $ax - by = \mu + \omega - 1$ sont appelées les **droites d'appui** de la droite naïve $\mathcal{D}(a, b, \mu)$. Un point entier de ces droites est nommé **point d'appui**.

La droite d'appui localisée au-dessus (resp. en-dessous) de \mathcal{D} dans le premier quadrant ($0 \leq a$ et $0 \leq b$) vérifie l'équation $ax - by = \mu$ (resp. $ax - by = \mu + \omega - 1$), et est appelée **droite d'appui supérieure** (resp. **droite d'appui inférieure**) de \mathcal{D} .

Définition 1.6 Le **reste au point** $M(x_M, y_M)$ en fonction de $\mathcal{D}(a, b, \mu, \omega)$, noté $r(M)$, est défini par

$$r(M) = ax_M - by_M.$$

La **reconnaissance des segments de droites discrètes arithmétiques naïves et standards** est un sujet que j'ai étudié durant ma thèse [7,43], certains des résultats obtenus sont utilisés dans les travaux présentés dans ce mémoire et je les rappelle brièvement.

Soit Σ un segment de $\mathcal{D}(a, b, \mu)$, avec $0 \leq a \leq b$, et M un point ajouté à Σ . Le théorème ci-dessous permet de dire si $\Sigma' = \Sigma \cup \{M\}$ est ou non un segment de droite discrète naïve et, si c'est le cas, il fournit les caractéristiques a', b', μ' de la droite correspondant à $\Sigma \cup \{M\}$.

Théorème 1.1 (7,43) Soit $\Sigma(M_0, M_1)$ un segment de $\mathcal{D}(a, b, \mu)$, avec $0 \leq a \leq b$. Le point $M(x_M, y_M)$ (avec $M = M_1 + (1, 0)$ ou $M = M_1 + (1, 1)$) est ajouté à Σ . U_F (resp. L_F) est le point d'appui supérieur (resp. inférieur) de Σ dont la coordonnée en x est minimale et U_L (resp. L_L) est le point d'appui supérieur (resp. inférieur) de Σ dont la coordonnée en x est maximale. On considère alors la valeur du reste au point M en fonction de \mathcal{D} ($r(M) = ax_M - by_M$). Alors, :

- (i) Si $\mu \leq r(M) < \mu + \omega$, $M \in \mathcal{D}(a, b, \mu)$, $\Sigma \cup \{M\}$ est un segment de \mathcal{D} .
- (ii) Si $r(M) = \mu - 1$, $\Sigma \cup \{M\}$ est un segment de la droite naïve dont la pente est donnée par le vecteur $\overrightarrow{U_F M}$.
- (iii) Si $r(M) = \mu + \omega$, $\Sigma \cup \{M\}$ est un segment de la droite naïve dont la pente est donnée par le vecteur $\overrightarrow{L_F M}$.
- (iv) Si $r(M) < \mu - 1$ ou $r(M) > \mu + \omega$, $\Sigma \cup \{M\}$ n'est pas un segment de droite discrète naïve.

Remarque : Dans le cas (iv) M est dit **fortement extérieur** à \mathcal{D} et :

- Si $r(M) < \mu - 1$ (M est au-dessus de Σ), il existe un point entier entre $\overrightarrow{U_F M}$ et Σ .
- Si $r(M) > \mu + \omega$ (M est en-dessous de Σ), il existe un point entier entre $\overrightarrow{L_F M}$ et Σ .

Un algorithme incrémental et linéaire de reconnaissance de segments de droites discrètes naïves a été élaboré à partir du théorème 1.1 (cf. [7,43]), la procédure de reconnaissance est nommée par la suite **RecoSegNaïf**, selon son utilisation, différents paramètres issus de la reconnaissance seront fournis comme résultat. Cet algorithme est rappelé dans l'article [6] figurant dans la partie III de ce mémoire.

La notion d'**hyperplan discret** généralisant les droites discrètes arithmétiques a aussi été proposée [Rev91, AAS97, And00] :

Définition 1.7 Un **hyperplan discret** de paramètres entiers $(a_1, \dots, a_n, \mu, \omega)$ est l'ensemble des points x de \mathbb{Z}^n qui vérifient

$$\mu \leq \sum_{i=1}^n a_i x_i < \mu + \omega$$

Un tel hyperplan est noté $\mathcal{P}(a_1, \dots, a_n, \mu, \omega)$. En fonction du paramètre ω , appelé **épaisseur arithmétique**, on distingue différentes classes d'hyperplans [AAS97, And00] :

- **les hyperplans naïfs** pour lesquels $\omega = \max(|a_i|)$; ces hyperplans sont fonctionnels selon un des axes de coordonnées dit **axe principal**. C'est la classe des hyperplans les plus minces sans $(n - 1)$ -tunnels [And00].
- **les hyperplans standards** pour lesquels $\omega = \sum_i (|a_i|)$; ces hyperplans ont des propriétés topologiques intéressantes.
- **les hyperplans épais** pour lesquels $\omega > \sum_i (|a_i|)$.

D'autres définitions et notations sont rappelées au cours de ce mémoire en fonction des besoins.

1.2 Décomposition polygonale de courbes discrètes en parties convexes et concaves

Les travaux présentés dans cette section ont été effectués sur plusieurs années de 1999 à 2006. Ils ont débuté par l'étude de la convexité discrète en collaboration avec Jean-Luc Rémy et Jocelyne Degli-Rouyer. Plus particulièrement, une propriété des segments de droites discrètes a été démontrée et utilisée dans un algorithme détectant la convexité des polyominos [6,18], elle est présentée dans le premier paragraphe. Des contacts sur ce sujet ont ensuite été établis avec le Professeur Eckhardt de l'Université d'Hambourg à l'occasion de mon invitation à un workshop en novembre 2002 [28]. Une de ses étudiantes, Hélène Dörksen-Reiter, doctorante, a fait plusieurs séjours au LORIA et nous avons alors travaillé ensemble sur des problèmes théoriques liés à la décomposition polygonale du bord d'un objet en relation avec la convexité discrète [5,12,35]. Ces travaux sont résumés dans le second paragraphe de cette section et font partie de son mémoire de thèse [Dör04].

1.2.1 Convexité discrète

La convexité discrète intervient dans de nombreux domaines de la géométrie et plus particulièrement en traitement d'image [CM91]. C'est une importante propriété des figures du plan qui permet, par exemple, de contrôler la régularité de formes géométriques.

La convexité est bien définie dans le cas continu mais, dans le cas discret, plusieurs définitions existent. Les figures discrètes étudiées sont des sous-ensembles finis 8-connexes du plan et sont appelées *régions discrètes*. En 1970, Slansky [Sla70] propose une première définition dépendant du processus de discrétisation utilisé :

Définition 1.8 Une *région discrète* est dite *convexe* s'il existe une région convexe (euclidienne) dont l'image (après discrétisation) est une région discrète.

D'autre part, Minsky et Papert [MP69] ont donné la définition suivante de la convexité d'une région discrète R :

Définition 1.9 R est *convexe* s'il n'existe pas de triplet de points discrets colinéaires (c_1, c_2, c_3) tel que c_1 et c_3 appartiennent à R et c_2 appartienne au complémentaire de R .

Depuis, Kim et Rosenfeld [KR80, KR82, KIM82] ont donné plusieurs définitions équivalentes :

Proposition 1.2 Une région discrète R est convexe

- si et seulement si son enveloppe convexe (euclidienne) ne contient aucun point discret du complémentaire de R .
- ou si et seulement si elle possède la propriété de l'aire, i.e. si et seulement si, pour tous points p_1 et p_2 de R , $P(R, p_1, p_2)$ ne possède pas de point du complémentaire de R , où $P(R, a, b)$ représente le polygone dont les arêtes sont formées par le segment ab et les arêtes de R (voir Fig. 1.2).

Nous avons utilisé ces deux définitions dans notre étude de la convexité des polyominos. Kim et Rosenfeld [KR80, KR82, KIM82] ont comparé ces définitions avec les précédentes de Minsky et Papert.

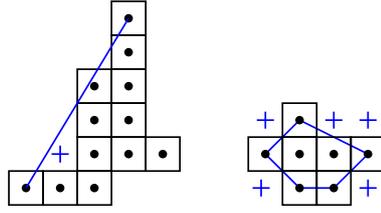


FIG. 1.2 – Sur la gauche, une région discrète non convexe. Sur la droite, une région discrète convexe.

Dans le travail [6,18], nous avons démontré une propriété sur l'enveloppe convexe d'un segment de droite discrète arithmétique naïve $\mathcal{D}(a, b, \mu)$. Pour cela, nous utilisons la notion de **reste au point** $M(x_M, y_M)$ **en fonction de** \mathcal{D} (cf. section 1.1).

Proposition 1.3 *Soit M_0M_1 un segment de la droite $\mathcal{D}(a, b, \mu)$ du premier quadrant, l'enveloppe convexe inférieure des points de M_0M_1 est la courbe polygonale passant par les points suivants (cf. Fig. 1.3 pour un exemple) :*

- *Le premier et le dernier point d'appui inférieur du segment, appelés L_F et L_L .*
- *Entre M_0 et L_F : la suite maximale (N_i) avec $x_{M_0} \leq x_{N_i} \leq x_{L_F}$, $N_0 = M_0$ tels que $x_{N_i} < x_{N_{i+1}}$ et $r(N_i) < r(N_{i+1}) \leq \mu + \omega - 1$.*
- *Entre M_1 et L_L : la suite maximale (P_i) avec $x_{M_1} \geq x_{P_i} \geq x_{L_L}$, $P_0 = M_1$ tels que $x_{P_i} > x_{P_{i+1}}$ et $r(P_i) < r(P_{i+1}) \leq \mu + \omega - 1$.*

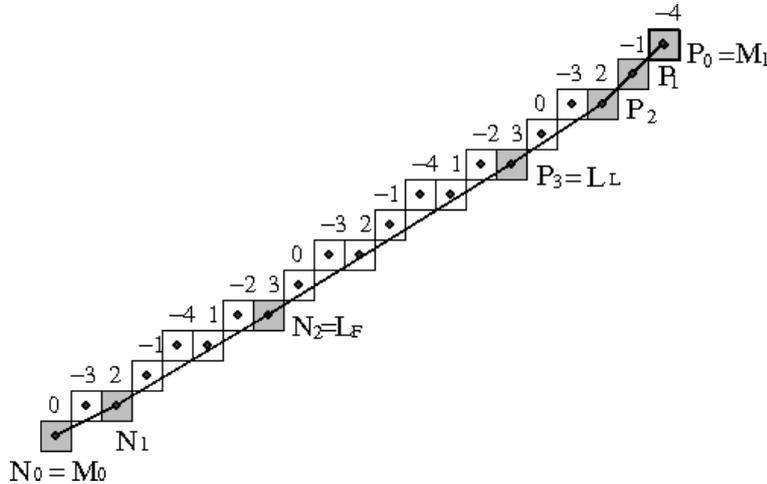


FIG. 1.3 – Segment de la droite naïve $\mathcal{D}(5, 8, -4)$, l'enveloppe convexe inférieure du segment est dessinée, la valeur du reste est indiquée en chaque point.

L'enveloppe convexe supérieure d'une droite discrète possède bien sûr la propriété symétrique de celle de la proposition 1.3 (remplacer $\mu + \omega - 1$ par μ et inverser les signes $<$ et $>$).

Un polyomino hv-convexe est un polyomino dont les points de chaque colonne sont consécutifs (v-convexité) ainsi que ceux des lignes (h-convexité). Par conséquent, définir sa convexité revient à étudier la convexité de son bord. En utilisant la propriété sur l'enveloppe convexe d'un segment de droite discrète naïve énoncée précédemment et l'algorithme de reconnaissance des segments de droites discrètes, nous avons obtenu un algorithme (cf. Algorithme 1) de **détection de la**

convexité du bord d'un polyomino hv-convexe de complexité linéaire. De plus, dans le cas où le polyomino est convexe, l'Algorithme 1 fournit les sommets de l'enveloppe convexe du bord du polyomino. Son principe dans le premier quadrant consiste à appliquer l'algorithme de reconnaissance des segments de droites naïves sur la courbe discrète correspondant au bord de l'objet ; quand un segment Σ de $\mathcal{D}(a, b, \mu)$ est reconnu, cela signifie que le dernier point M ajouté n'a pas pu former un segment avec Σ , deux cas sont possibles :

- le point M allonge un palier de \mathcal{D} et est tel que $ax_M - by_M > \mu + \max(a, b)$, dans ce cas, selon la remarque suivant le théorème 1.1 de la section 1.1, il existe un point entier entre Σ et $\overrightarrow{L_F M}$, par conséquent la courbe n'est pas convexe.
- le point M raccourcit un palier de \mathcal{D} et dans ce cas, la courbe reste convexe. Pour obtenir une suite de segments de pentes croissantes, la proposition 1.3 est utilisée et le segment courant est raccourci et s'arrête sur le dernier point d'appui inférieur qui devient aussi le premier point du nouveau segment.

Dans cet algorithme, **RecoSegNaïf** est l'algorithme de reconnaissance de segments de droites naïves (cf. section 1.1) et retourne ici le point M , dernier point testé ainsi que les caractéristiques a, b, μ de la droite naïve correspondant au segment reconnu, L_L , dernier point d'appui inférieur du segment et un booléen *fin* indiquant si la courbe a été entièrement parcourue.

Définition 1.10 Une courbe discrète \mathcal{C} est dite **convexe inférieure** (resp. **convexe supérieure**) s'il n'existe pas de point entier entre \mathcal{C} et la frontière inférieure (resp. supérieure) de l'enveloppe convexe de \mathcal{C} .

Théorème 1.4 Une courbe \mathcal{C} du premier quadrant est convexe inférieure si et seulement si l'Algorithme 1 parcourt entièrement \mathcal{C} (convexe = vrai à la fin).

Dans l'article [6], en annexe de ce mémoire (partie III), le théorème 1.4 a été démontré ainsi que la linéarité de l'Algorithme 1. Ce résultat et cet algorithme ont été utilisés dans le paragraphe suivant et ont permis d'obtenir un algorithme de décomposition d'une courbe discrète en parties convexes et concaves ainsi qu'une représentation polygonale d'une courbe discrète.

Algorithme 1 : Algorithme testant la convexité de la courbe inférieure associée au bord d'un polyomino dans le premier quadrant

Entrée : \mathcal{C} suite 8-connexe de points de \mathbb{Z}^2 du premier quadrant.

Résultat : la variable booléenne *convexe* à vrai si \mathcal{C} est convexe

```

convexe = vrai ;
fin = faux ;
premierPoint = premier point de  $\mathcal{C}$  ;
tant que convexe et !fin faire
    RecoSegNaïf(premierPoint)  $\rightarrow$  ( $M, a, b, \mu, L_L, fin$ ) ;
    si  $ax_M - by_M > \mu + \max(a, b)$  alors
        | convexe = faux ;
    sinon
        | premierPoint =  $L_L$  ;

```

1.2.2 Décomposition polygonale

Avec Hélène Dörksen-Reiter, nous utilisons la notion de *segment fondamental d'une courbe 8-connexe* correspondant à la notion de *segment maximal d'une courbe 8-connexe* introduite par Fabien Feschet et Laure Tougne en 1999 [FT99].

Définition 1.11 Soit \mathcal{C} une courbe discrète 8-connexe, un segment d'une droite discrète naïve $\mathcal{D}(a, b, \mu)$ appartenant à \mathcal{C} est **fondamental** ou **maximal**, noté $\mathcal{F}(a, b, \mu)$, s'il ne peut pas être étendu à l'une de ses extrémités sur \mathcal{C} en restant un segment de droite discrète naïve.

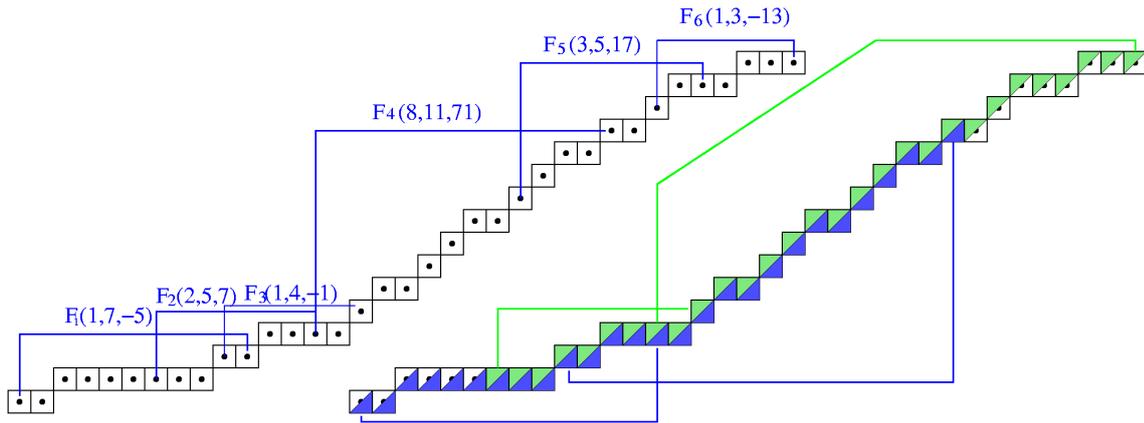


FIG. 1.4 – À gauche, segments fondamentaux $\mathcal{F}_i(a_i, b_i, \mu_i)$, $i = 1, \dots, 6$ de la courbe discrète. Le premier point de la courbe a pour coordonnées $(0, 0)$. Les bornes inférieures μ_i , $i = 1, \dots, 6$ sont calculées par rapport à $(0, 0)$. À droite, parties maximales convexes (en bleu) et concaves (en vert) de la courbe.

Il est clair que la suite des segments discrets fondamentaux d'une courbe discrète est *unique* et la succession des segments discrets fondamentaux fournit des informations sur la structure de la courbe. Nous avons alors introduit deux définitions reposant sur les segments fondamentaux d'une courbe discrète :

Définition 1.12 Une courbe discrète \mathcal{C} du premier octant est dite *convexe (resp. concave)* si la suite des pentes de ses segments fondamentaux est croissante (resp. décroissante).

Définition 1.13 Les *parties maximales convexes (resp. concaves)* d'une courbe discrète \mathcal{C} du premier octant sont les suites maximales des segments fondamentaux de \mathcal{C} ayant des pentes croissantes (resp. décroissantes).

Une illustration de cette définition se trouve à la Figure 1.4 : les pentes des 6 segments fondamentaux successifs sont $\frac{a_i}{b_i} = 0.1429, 0.4, 0.25, 0.7273, 0.6, 0.3333$. Nous en déduisons que la courbe de la Figure 1.4 possède 4 séquences maximales : $\mathcal{F}_1, \mathcal{F}_2$ et $\mathcal{F}_3, \mathcal{F}_4$ sont des séquences maximales convexes et, les séquences $\mathcal{F}_2, \mathcal{F}_3$ et $\mathcal{F}_4, \mathcal{F}_5, \mathcal{F}_6$ sont des séquences maximales concaves.

Il a été montré le résultat suivant dans l'article [5], en annexe de ce mémoire (partie III), qui fait le lien entre l'enveloppe convexe de la courbe discrète et la notion de courbe discrète convexe :

Théorème 1.5 Une courbe discrète \mathcal{C} du premier octant est convexe si et seulement si elle est convexe inférieure (i.e. si il n'y a pas de point entier entre \mathcal{C} et son enveloppe convexe inférieure).

Nous nous sommes ensuite intéressées à la décomposition polygonale d'une courbe discrète et nous avons défini la notion de **représentation polygonale fondamentale** :

Définition 1.14 Soit $(\mathcal{F}_i(a_i, b_i, \mu_i))_{i=1, \dots, m}$, la suite des segments fondamentaux d'une courbe discrète \mathcal{C} . La courbe polygonale dont les côtés sont les segments issus des droites d'appui inférieures (resp. supérieures) des $\mathcal{F}_i(a_i, b_i, \mu_i)$ et les sommets sont leurs points d'intersection successifs est appelée **représentation polygonale fondamentale inférieure (resp. supérieure)** (cf. Figure 1.5).

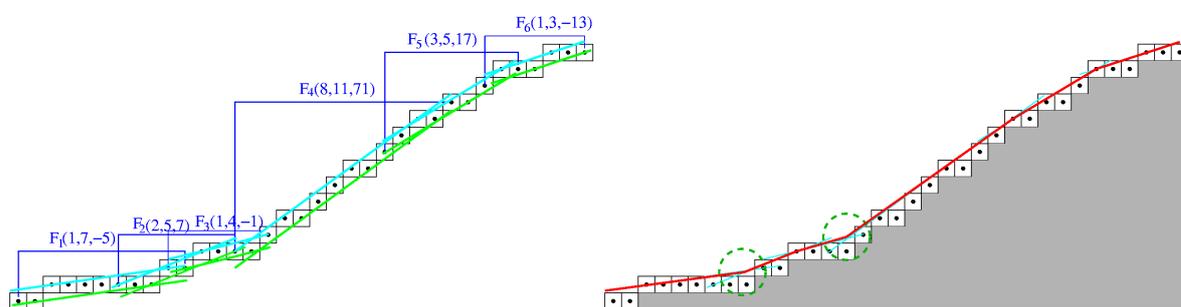


FIG. 1.5 – A gauche, droites d'appui des segments fondamentaux de la courbe discrète. A droite, représentation polygonale fondamentale supérieure.

La représentation polygonale fondamentale inférieure ou supérieure d'une courbe discrète possède de nombreuses propriétés :

- *il n'y a pas de point entier entre la représentation polygonale et la courbe discrète,*
- *la représentation polygonale a les mêmes propriétés de convexité et concavité que la courbe discrète,*
- Si les points d'appui inférieurs (resp. supérieurs) des segments fondamentaux $\mathcal{F}_i(a_i, b_i, \mu_i)$, $i = 1, \dots, m$ d'une courbe convexe (resp. concave) sont tels que $L_{L_j} = L_{F_{j+1}}$ (resp. $U_{L_j} = U_{F_{j+1}}$)⁶ pour $1 \leq j \leq m - 1$ alors les sommets de la représentation polygonale inférieure (resp. supérieure) entre L_{F_1} et L_{L_m} (resp. U_{F_1} et U_{L_m}) sont les sommets de l'enveloppe convexe inférieure (resp. supérieure) de la courbe discrète.

Cette représentation s'obtient très facilement. En effet, la suite des segments fondamentaux d'une courbe discrète est calculée par l'algorithme linéaire proposé par F. Feschet et L. Tougne dans [FT99], puis les points d'intersection des droites d'appui supérieures (resp. inférieures) des couples de segments fondamentaux successifs sont calculés et la suite des sommets de la représentation polygonale fondamentale supérieure (resp. inférieure) est ainsi obtenue.

Cependant, en général, les sommets de la représentation polygonale fondamentale inférieure ou supérieure ne sont pas des points entiers ; par exemple on peut voir sur la Figure 1.5 à droite que 2 sommets de la représentation polygonale fondamentale de la courbe discrète ne sont pas des

⁶ L_{F_i} et U_{F_i} sont respectivement les premiers points d'appui inférieur et supérieur du segment fondamental \mathcal{F}_i . L_{L_i} et U_{L_i} sont respectivement les derniers points d'appui inférieur et supérieur du segment fondamental \mathcal{F}_i .

points entiers (sommets entourés par un cercle en pointillés). Nous avons donc étudié précisément dans l'article [12] les conditions requises pour qu'une courbe discrète possède une représentation polygonale :

- respectant les convexités et concavités de la courbe discrète,
- dont les sommets sont entiers.

Une étude théorique des différentes situations a été réalisée et nous avons élaboré un algorithme linéaire permettant d'obtenir, quand cela est possible, une représentation polygonale à sommets entiers et respectant les convexités et concavités de la courbe discrète [12]. Cet algorithme utilise la propriété sur l'enveloppe convexe d'un segment discret (cf. paragraphe 1.2.1) ainsi que l'algorithme de reconnaissance de segments de droites naïves [7]. Afin de le décrire succinctement, nous distinguons la **représentation polygonale supérieure** des courbes discrètes concaves et convexes.

Pour les **courbes discrètes concaves**, l'Algorithme 2, qui est une variante de l'Algorithme 1 du paragraphe précédent et qui utilise la proposition 1.3, fournit toujours une représentation polygonale supérieure dont les sommets sont entiers et qui respecte les convexités et concavités de la courbe discrète. Dans cet algorithme, **RecoSegNaïf** est l'algorithme de reconnaissance de segments de droites naïves (cf. section 1.1), il retourne ici les caractéristiques a , b , μ de la droite naïve correspondant au segment reconnu ainsi que U_F et U_L , premier et dernier point d'appui inférieur du segment, et un booléen *fin* indiquant si la courbe a été entièrement parcourue.

Dans le cas d'une **courbe discrète convexe**, il n'existe pas toujours de telle représentation polygonale supérieure et un complément à cet algorithme est proposé dans [12] et [Dör04] qui repose sur les propositions 1.6 et 1.7.

Proposition 1.6 *Soit \mathcal{C} une courbe discrète du premier octant constituée de 2 segments fondamentaux \mathcal{F}_1 et \mathcal{F}_2 . Si $x_{U_{L_1}} < x_{U_{F_2}}$, il n'existe pas de représentation polygonale supérieure de \mathcal{C} dont les sommets soient entiers et qui respecte les convexités et concavités de la courbe discrète. De plus, il existe au moins un point entre U_{L_1} et U_{F_2} qui implique une concavité dans la représentation polygonale supérieure (cf. Figure 1.6).*

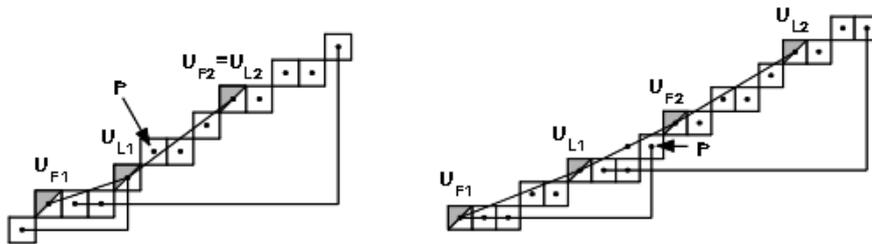


FIG. 1.6 – Exemples de courbes convexes ne possédant pas de représentation polygonale supérieure dont les sommets sont entiers et qui respecte les convexités et concavités de la courbe discrète. Sur la courbe de gauche, l'Algorithme 2 a trouvé le sommet P qui implique une concavité. Sur la courbe de droite, il calcule le sommet P tel que U_{L_1} et U_{F_2} impliquent des concavités avec P .

Proposition 1.7 *Soit \mathcal{C} une courbe discrète du premier octant constituée de 2 segments fondamentaux \mathcal{F}_1 et \mathcal{F}_2 . Si une des conditions ci-dessous est respectée alors il existe une représentation*

Algorithme 2 : Algorithme déterminant la représentation polygonale supérieure d'une courbe discrète concave du premier octant

Entrées : \mathcal{C} suite 8-connexe de n points $(M_i)_{i=1\dots n}$ de \mathbb{Z}^2 du premier octant.

Résultat : L La liste des sommets de la représentation polygonale supérieure de \mathcal{C}

$L = \emptyset$;

$fin = \text{faux}$; $Deb = M_1$;

répéter

$\text{RecoSegNaif}(Deb) \rightarrow (a, b, \mu, U_F, U_L, fin)$;

$L \leftarrow L \cup$ suite des $\{P_i\}_{1\dots k}$ entre Deb et U_F tels que

$r(Deb) > r(P_i) > r(P_{i+1}) > r(U_F)$;

$L \leftarrow L \cup U_F \cup U_L$;

$Deb \leftarrow U_L$;

jusqu'à fin

$L \leftarrow L \cup$ suite des $\{P_i\}_{1\dots k}$ entre M_n et U_L tels que $r(M_n) > r(P_i) > r(P_{i+1}) > r(U_L)$;

polygonale supérieure entre U_{F_1} et U_{L_2} dont les sommets sont entiers et qui respecte les convexités et concavités de la courbe discrète :

1. $x_{U_{F_2}} = x_{U_{L_1}}$,
2. $x_{U_{F_2}} < x_{U_{L_1}}$ et il n'existe pas de point M de \mathcal{C} tel que M se trouve au dessus du segment $U_{L_1}U_{L_2}$ ($x_{U_{L_1}} < x_{U_{L_2}}$),
3. $x_{U_{F_2}} < x_{U_{L_1}}$ et il n'existe pas de point M de \mathcal{C} tel que M se trouve au dessus du segment $U_{F_1}U_{F_2}$ ($x_{U_{F_1}} < x_{U_{F_2}}$).

Un lemme généralise ces propositions à des courbes discrètes constituées de $m \geq 2$ segments fondamentaux et un exemple de résultat obtenu par l'Algorithme 2 étendu est donné à la Figure 1.7.

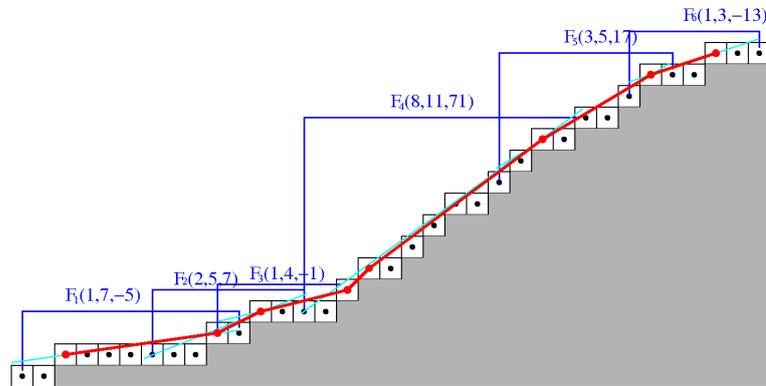


FIG. 1.7 – Représentation polygonale avec sommets entiers et respect des convexités/concavités de la courbe discrète.

1.2.3 Perspectives

Nous avons proposé dans cette section une propriété sur l'enveloppe convexe d'un segment de droite discrète qui nous a permis d'élaborer un algorithme de détection de la convexité du bord inférieur d'une courbe discrète. Cette propriété a aussi été utilisée dans le cadre de la recherche de la représentation polygonale d'une courbe discrète reposant sur ses segments fondamentaux, appelée représentation polygonale fondamentale qui garantit d'être à une distance inférieure à 1 de la courbe discrète et respecte les concavités et convexités de la courbe. De plus une étude a été menée identifiant les cas où une courbe discrète ne possède pas de représentation polygonale à sommets entiers qui respecte les concavités et convexités de la courbe discrète.

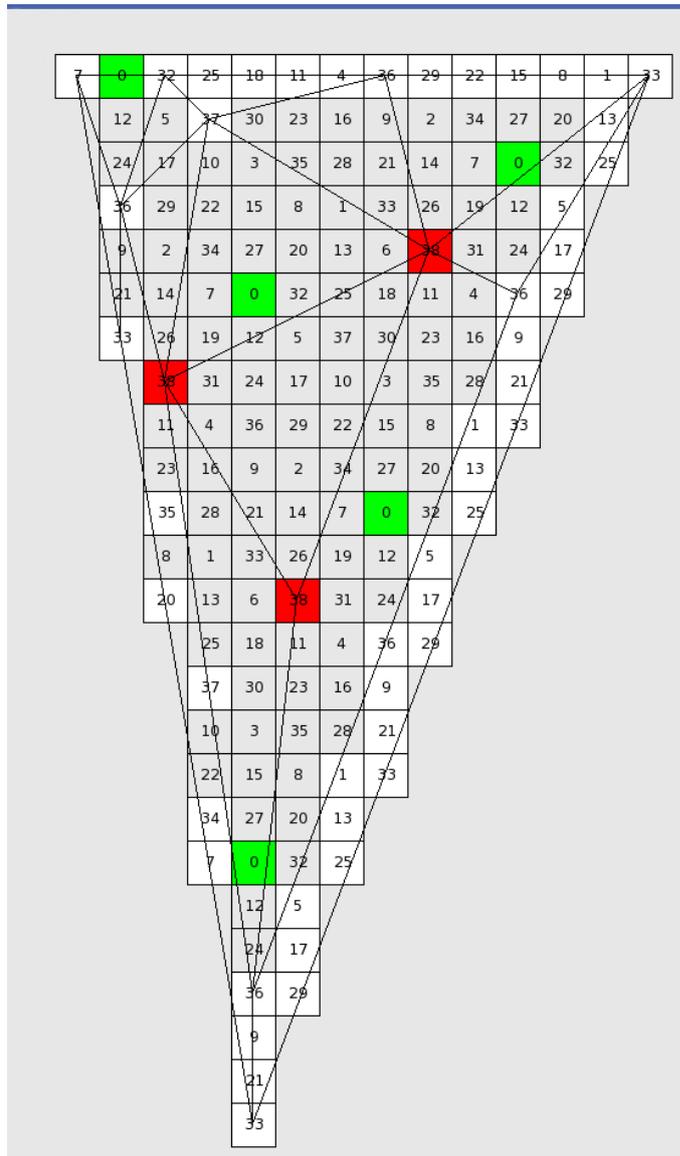


FIG. 1.8 – Représentation par restes d'un morceau du plan discret $P(12, 32, 39, 0, 39)$, son enveloppe convexe est tracée sur le morceau.

Nous avons montré que l'utilisation des segments fondamentaux d'une courbe discrète est

une démarche pertinente pour obtenir une représentation polygonale significative de la courbe discrète étudiée et nous souhaitons utiliser ce type d'approche en dimension 3 dans le cadre de la recherche de la représentation polyédrique d'objets discrets. Cela implique bien sûr de nombreux travaux préliminaires et la notion de *patch* développée au chapitre 3 pour les objets discrets bruités peut très bien être utilisée avec des morceaux de plans discrets. De plus, nous avons remarqué que la propriété 1.3 sur l'enveloppe convexe d'un segment de droite discrète naïve reposant sur les séquences des restes peut être étendue aux morceaux de plans discrets naïfs (cf. Figure 1.8) mais doit être démontrée. Cette étape franchie, nous étudierons les enveloppes convexes d'un objet discret composé de deux morceaux de plans naïfs.

D'autres perspectives sont aussi envisagées du côté de la représentation polygonale d'une courbe discrète bruitée, liée à la décomposition en segments flous arithmétiques (cf. chapitre 2 section 2.2), elles seront précisées dans le chapitre 2.

1.3 Estimation des normales à une surface discrète

Une bonne estimation des vecteurs normaux à un objet discret est primordiale pour de nombreuses opérations comme par exemple la simulation de l'éclairage d'un objet discret. Plusieurs auteurs se sont intéressés à ce problème en utilisant différentes approches. Une synthèse des travaux entrepris sur ce sujet jusqu'en 1992 est présentée dans un article de Cohen [YCK92]. Depuis cette date, d'autres approches considérant le voisinage 3D des points sont apparues, l'une d'entre elles, proposée par G. Thurmer et A. Wuthrich, [TW97, Thu01] utilise le volume autour du point où le vecteur normal est calculé. Une autre [PF98], introduite par L. Papier et J. Françon, repose sur une généralisation de la notion d'ombrelle [Lef75, Fran95] en un sommet d'un volume.

En 1996 [LMR96], A. Lenoir et al. proposent un estimateur de normales reposant sur le calcul des tangentes des contours des coupes parallèles aux plans de coordonnées de la surface de l'objet. Ils utilisent pour cela un calcul de dérivées partielles avec un paramètre permettant de fixer la taille du voisinage. En 1999, en utilisant aussi les coupes des objets voxels mais avec un calcul différent des tangentes sur un voisinage de taille non fixée, nous avons proposé une autre méthode [19] avec Pierre Tellier (LSIIT, Strasbourg) qui fut initiée dans le cadre du stage de DEA de Yannick Kirschhoffer (LSIIT, Strasbourg, 1997) [Kir97]. Celle-ci considère qu'un point entier est représenté par un voxel ; un cube dont les côtés sont des unités et les faces des surfels. Le voisinage de chaque point est considéré sans lissage.

Cette nouvelle estimation des vecteurs normaux à une surface discrète 3D repose sur la notion de tangente discrète 2D définie par A. Vialard [Via96a, BV99] ; pour calculer le vecteur normal en un point du bord de l'objet discret, nous utilisons les coupes de l'objet dans les 3 plans canoniques passant par ce point (cf. Figure 1.9). Le principe utilisé est le suivant : sur le contour de chaque coupe, nous calculons la tangente discrète au point considéré puis en combinant les tangentes discrètes 2D obtenues (1, 2 ou 3) et en utilisant un produit vectoriel nous obtenons le vecteur normal 3D au point de l'objet voxel. Nous avons aussi proposé une approche surfel en calculant selon le même principe un vecteur normal en chaque surfel du bord d'un objet voxel.

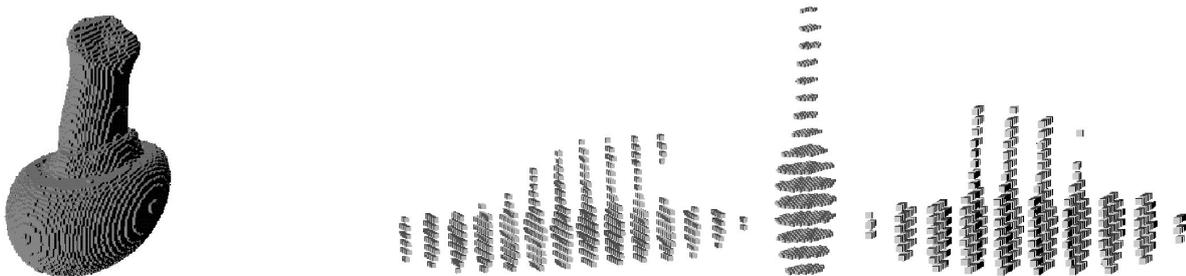


FIG. 1.9 – Champignon discrétisé et ses coupes.

Nous utilisons dans cette section la notion de *tangente discrète* introduite par A. Vialard [Via96a] :

Définition 1.15 *La tangente discrète en un point M d'une suite C de pixels 8- ou 4-connexe est la plus longue portion de C centrée en M qui soit un segment de droite discrète naïve ou standard.*

A. Vialard a proposé avec A. Braquelaire dans [Via96a, BV99] un algorithme de reconnaissance incrémental d'un segment de droite discrète centré en un point permettant de déterminer le plus long segment discret centré en M en ajoutant à chaque étape un point à chacune des

extrémités. Si le couple de points ajouté ne permet pas de prolonger le segment, aucun des deux points n'est ajouté. Cet algorithme est une extension de l'algorithme de reconnaissance de segments de droites naïves proposé pendant ma thèse [7] (cf. section 1.1). Nous utilisons dans la suite de cette section cet algorithme permettant de déterminer la tangente discrète 2D pour un point d'une suite de pixels 8- ou 4-connexe.

1.3.1 Calcul des normales 3D en chaque point du bord d'un objet voxel

Dans toute cette étude, nous notons \mathcal{O} un objet voxel 6-connexe. Le **bord** de \mathcal{O} est l'ensemble des points de \mathcal{O} ayant un 6-voisin n'appartenant pas à \mathcal{O} , cet ensemble est noté \mathcal{B} . Dans cette section le bord de l'objet discret est assimilé à sa surface.

Le calcul des normales en tout point du bord \mathcal{B} d'un objet voxel 6-connexe \mathcal{O} repose sur l'étude des contours associés à chaque point du bord de l'objet et situés sur les coupes de l'objet passant par le point considéré :

Définition 1.16 *Soit M un point du bord \mathcal{B} d'un objet discret tridimensionnel \mathcal{O} , une **coupe en x (resp. y, z) au point M** , avec $M(x_M, y_M, z_M)$, est l'ensemble des points de \mathcal{O} ayant même coordonnée x_M (resp. y_M, z_M) (cf. Figure 1.9).*

*Les **pixels associés à M** sont les points 2D qui correspondent aux projetés de M sur les plans de coordonnées.*

*Les **contours** associés à une coupe de l'objet sont formés par les suites 8-connexes fermées des pixels associés aux points des bords de la coupe (cf. Figure 1.11).*

Les normales en chaque point de \mathcal{B} sont calculées en fonction des contours auxquels ces points appartiennent. Et selon la forme de ces contours, associer une seule normale par voxel peut poser des problèmes, c'est pourquoi nous avons adopté la stratégie proposée dans le paragraphe suivant.

Position des voxels sur le bord de l'objet

Nous avons volontairement adopté un point de vue réaliste en voulant calculer la meilleure estimation de normale pour obtenir un rendu efficace. Or, l'approche voxel qui consiste à considérer un point entier comme un cube à 6 faces, engendre des problèmes dus à la position du point sur le bord de l'objet.

Dans le cas général, correspondant aux cas dit **réguliers**, les voxels possèdent 1, 2 ou 3 faces, appelées **surfels**, non opposées 2 à 2, sur le bord de l'objet.

Dans les autres cas, dits **non réguliers**, nous trouvons deux types de configuration de voxels :

- les voxels possédant au moins deux faces opposées adjacentes au bord, c'est-à-dire appartenant à des endroits **fins** de l'objet ayant une épaisseur de 1 voxel (cf. Figure 1.10 à droite),
- les voxels de coin ou d'arête (cf. Figure 1.10 à gauche) correspondant à des **discontinuités** de l'objet.

Ces différentes configurations sont simplement détectables en dimension 2 sur les contours calculés au point M considéré. Supposons que chaque contour soit codé selon le code de Freeman [Fre70], pour chaque contour passant par M , on note d_{ar} (resp d_{av}) la direction représentant le déplacement vers le pixel suivant (resp. précédant) le pixel associé à M sur le contour dans le sens trigonométrique. Nous en déduisons la classification suivante pour un point M de \mathcal{B} selon les valeurs de d_{av} et d_{ar} .



FIG. 1.10 – A gauche, en gris, les voxels réguliers et en blanc les voxels non réguliers. A droite, un exemple d'objet fin d'épaisseur un voxel.

Le voxel M est dans la configuration générale (cas régulier), si dans **chaque coupe** passant par M , **au plus un contour passe par un pixel associé à M** et si :

$$d_{ar} \equiv d_{av} + 3[8] \text{ ou } d_{ar} \equiv d_{av} + 4[8] \text{ ou } d_{ar} \equiv d_{av} + 5[8]$$

Dans le cas contraire (cas non régulier),

- si, dans une même coupe, plusieurs contours passent par un pixel associé à M , ou bien si un contour passe plusieurs fois par ce point, le point M est situé à un endroit *fin* de l'objet,
- si sur un contour passant par M les directions d_{ar} et d_{av} sont incompatibles, le bord de l'objet voxel présente une *discontinuité* à cet endroit.

Des exemples de ces 3 types de points apparaissent sur un des contours associés à une coupe de champignon numérisé de la figure 1.11. Par exemple, au pixel R , $d_{av} = 0$ et $d_{ar} = 7$ sont deux directions incompatibles, il y a donc une discontinuité au point correspondant.

Il apparaît clairement qu'une normale unique ne peut pas être affectée à un voxel dans une

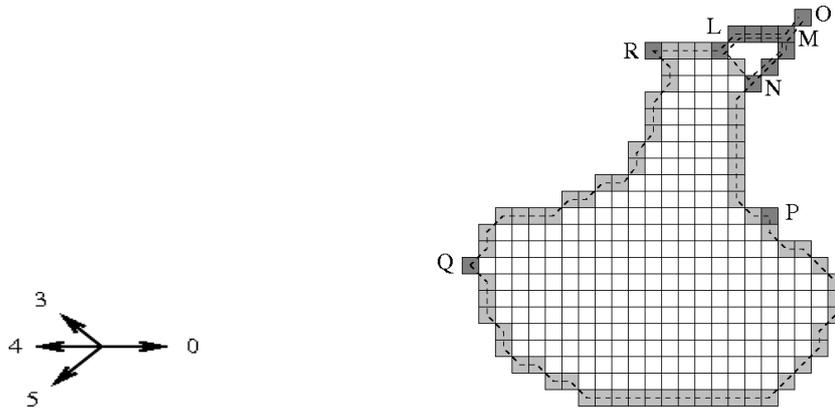


FIG. 1.11 – A gauche, $d_{av} = 0$, les seules directions arrières possibles pour être dans le cas général sont $d_{ar} = 3, 4$ ou 5 . A droite, les contours de cette coupe de champignon sont marqués en pointillés. Les pixels de L à M et de M à N sont des points non réguliers car deux contours passent par ces points; les pixels O , P , Q et R aussi car les directions d_{av} et d_{ar} associées à ces pixels sont incompatibles. Les autres pixels grisés des contours sont réguliers.

configuration non régulière, la décomposition en surfels de ce voxel doit alors être considérée. Les calculs des normales pour ces différentes configurations sont décrits dans le paragraphe suivant.

Principe de l'estimation des normales

Dans *le cas général*, nous définissons la *normale* en un point M du bord d'un objet voxel comme la somme des vecteurs (un, deux ou trois) obtenus après les étapes suivantes :

- chercher dans chaque coupe associée au point M , sur le contour passant par M , le vecteur tangent 2D au pixel associé à M , l'*étendre* à la dimension 3.
- si 2 ou 3 vecteurs tangents ont été obtenus, effectuer les produits vectoriels entre ces vecteurs (dans le cas de 3 vecteurs tangents, 3 produits vectoriels sont effectués et la normale considérée est la moyenne des 3 vecteurs obtenus), sinon prendre le vecteur perpendiculaire au seul vecteur tangent obtenu.

La recherche des vecteurs tangents 2D nécessite plusieurs phases :

- (i) Déterminer les 3 coupes auxquelles le point M appartient.
- (ii) Rechercher dans la liste des contours associés à chacune des 3 coupes, les contours passant par le point M .
- (iii) Pour chaque contour passant par M , considérer ce contour comme un chemin discret et calculer la tangente au pixel associé à M en utilisant l'algorithme en 8-connexité [Via96a, BV99].

La normale obtenue est alors associée à chaque surfel du voxel appartenant au bord de l'objet.

Dans les *cas non réguliers*, la recherche des vecteurs tangents et l'affectation des normales aux surfels est différente ; les surfels du bord d'un même voxel ne vont pas avoir la même normale. Par conséquent, pour chaque surfel d'un voxel, trois tangentes (une par coupe) pourront lui être associées, l'algorithme de recherche des normales aux surfels est le suivant :

Dans les trois coupes auxquelles le point M appartient,

- Si le pixel associé à M est isolé, les tangentes standards sont affectées à chacun de ces côtés puis répercutées sur les surfels correspondants (cf. l'étude du point B dans le plan Oxz de la Figure 1.12).
- Sinon, pour chaque contour passant par un pixel associé à M ,
 - * Si le contour présente en ce point une discontinuité (d_{av} et d_{ar} incompatibles), calculer la tangente à droite et à gauche au pixel considéré en utilisant les algorithmes en 8-connexité.

L'affectation des tangentes aux côtés des pixels puis aux surfels correspondants se fait de la manière suivante :

- Le surfel correspondant au côté adjacent à la direction (d_{av}) dans le sens trigonométrique hérite de la demi-tangente correspondant au demi-contour avant.
- De façon symétrique, le surfel correspondant au côté adjacent à la direction d_{ar} dans le sens trigonométrique inverse prendra la valeur de la tangente associée au demi-contour arrière.
- Enfin, si en balayant l'ensemble des directions dans le sens trigonométrique de d_{av} jusqu'à d_{ar} , un des côtés du pixel courant est sur le bord de l'objet et ne possède pas de tangente dans cette coupe, lui affecter la somme des deux demi-tangentes orientées dans le même sens.
- * Sinon, le contour ne présente pas de discontinuité en ce point, calculer la tangente au pixel associé à M en utilisant l'algorithme décrit dans la section 2. Cette tangente est associée à chaque surfel (représenté par une arête du pixel) inscrit dans la portion de plan associée à ce contour.

En fonction des tangentes affectées aux surfels, les normales aux surfels sont calculées par

produits vectoriels.

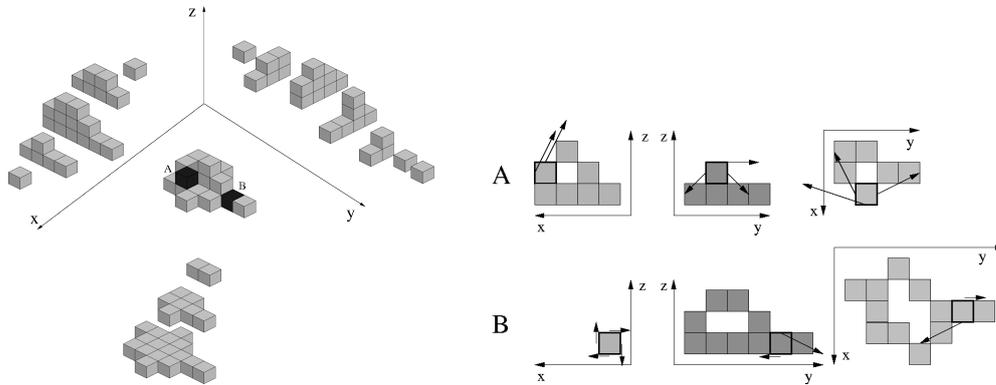


FIG. 1.12 – Étude de deux voxels A et B non réguliers sur un objet voxel. A gauche, la représentation voxel de cet objet avec ses coupes. A droite, les tangentes obtenues sur chaque contour pour les points A et B.

Remarques :

1. Les cas non réguliers ne forment qu'une partie infime des objets voxels synthétiques ou réels ; des résultats sont donnés dans le paragraphe suivant. Mais tenir compte de ces cas particuliers permet d'obtenir un rendu très réaliste sur les extrémités des objets et cela même en cas d'aspérité (cf. Figures 1.14 à 1.21).
2. Des tests ont été réalisés en utilisant des variantes [Via96a] de la définition de tangente discrète : tangente réduite (les points des extrémités du segment sont supprimés), tangente asymétrique (prolongement de la tangente à une extrémité), tangente asymétrique et réduite. Ces différentes définitions n'ont pas donné de résultats satisfaisants et n'apportent pas d'amélioration à la première définition.

1.3.2 Calcul des normales 3D en chaque surfel du bord d'un objet voxel

Après la définition utilisée précédemment, il a semblé naturel de s'intéresser à une approche surfel, en calculant la normale à chaque surfel du bord d'un objet voxel. L'idée générale reste identique mais quelques définitions supplémentaires sont nécessaires :

Définition 1.17 *Les côtés d'un pixel sont appelés **lignels** et ses sommets des **pointels**. Les **contours pointels** associés à une coupe de l'objet sont formés par les suites 4-connexes des pointels des pixels du bord de la coupe (cf. Figure 1.13).*

La **normale à un surfel** du bord d'un objet voxel est le vecteur obtenu après les étapes suivantes :

- Chercher les deux coupes auxquelles il appartient. Par exemple si le surfel est parallèle au plan Oxy , les deux coupes se feront parallèlement aux plans Oxz et Oyz .
- Déterminer dans chaque coupe le contour pointel passant par le lignel associé au surfel, puis calculer les tangentes discrètes 4-connexes en chacun des pointels délimitant le lignel associé au surfel considéré. La tangente au lignel est obtenue en calculant la moyenne des tangentes aux pointels.
- Le produit vectoriel résultant des tangentes calculées (puis étendues à la dimension 3) dans chaque coupe est défini comme la normale au surfel.

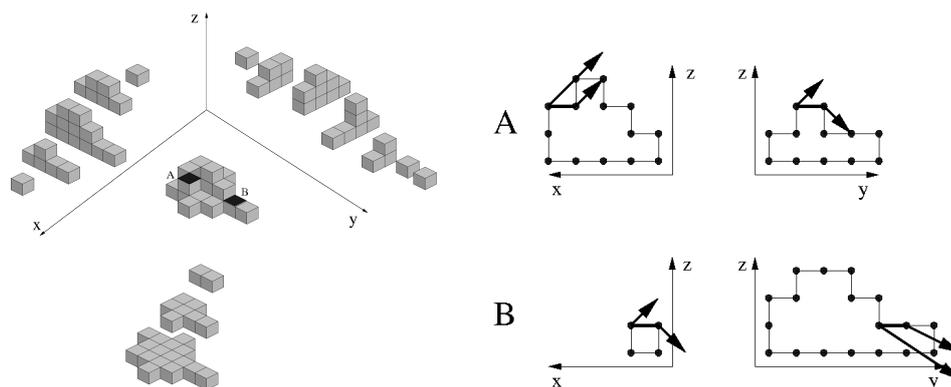


FIG. 1.13 – Étude de deux surfels A et B sur un objet voxel. A gauche, la représentation voxel de cet objet avec ses coupes. A droite, les tangentes aux pointels obtenues sur chaque contour pointel pour les surfels A et B.

Remarques :

1. Pour chaque surfel du bord, quatre calculs de tangentes discrètes sont nécessaires (un seul est nécessaire dans l'approche voxel pour les cas réguliers).
2. La méthode proposée est toujours applicable, il n'y a pas de cas particulier.
3. Deux tangentes sont affectées par surfel alors que dans la méthode précédente cela pouvait aller jusqu'à trois.

1.3.3 Résultats

L'évaluation des normales est réalisée de deux manières ; sur des objets synthétiques en comparant les normales obtenues aux normales théoriques et visuellement sur des objets numérisés.

Les Figures 1.14 et 1.16 à 1.21 présentent dans la partie gauche l'objet voxel et dans la partie droite le rendu obtenu avec les normales calculées en chaque voxel de l'objet.

Nous avons comparé les normales obtenues aux normales théoriques calculées comme suit :

- Un plan $P(a, b, c, \mu, \omega)$ est défini par l'ensemble des points entiers (x, y, z) vérifiant $\mu \leq ax + by + cz < \mu + \omega$ où tous les paramètres sont entiers. La normale théorique est le vecteur (a, b, c) .
- Une sphère de rayon r entier est définie par les points entiers (x, y, z) vérifiant $x^2 + y^2 + z^2 < r(r + 1)$ [And92]. La normale théorique est le vecteur qui part du centre de la sphère et passe par le centre des voxels (resp. surfels) dans le cas de l'approche voxel (resp. surfel).
- La normale théorique à un voxel (resp. surfel) d'un cube est la normale à la face à laquelle il appartient.

L'angle entre la normale théorique et la normale estimée est calculé en chaque point des objets testés. La moyenne et l'écart type de ces valeurs sont présentés dans le tableau ci-après selon les méthodes exposées dans les paragraphes précédents.

Objets	Méthode du paragraphe 1.3.1		Méthode du paragraphe 1.3.2	
	Erreur moyenne	Ecart type	Erreur moyenne	Ecart type
Sphère de rayon 5	4.83	3.63	6.95	5.23
Sphère de rayon 11	3.59	2.71	3.65	1.91
Sphère de rayon 15	2.87	2.20	3.01	1.86
Sphère de rayon 25	2.84	2.24	3.22	1.92
Cube d'arête 15	0.00	0.00	8.18	4.82
Plan (1,2,-5,-10)	1.78	0.00	1.78	0.00
Plan (3,4,-5,-5)	1.02	0.00	1.02	0.00
Plan (1,7,-25,54)	2.28	0.90	2.26	0.92

On constate des erreurs plus importantes avec la deuxième méthode sur des objets comme le cube (cf Figure 1.15) ; les coins sont lissés.

Afin d'estimer le coût des deux méthodes, le tableau ci-dessous précise pour les objets présentés la résolution, le nombre de voxels et de surfels du bord, le nombre de voxels non réguliers ainsi que les temps de calcul des codes de Freeman des coupes des objets et le temps de calcul des normales. Ces travaux ont été réalisés sur Silicon Graphics R10000 195 Mhz.

Scène	Résolution	Voxels du bord	Surfels	Non réguliers	Freeman	Normales
Veines	266x285x241	40912	89238	11232	438 s	40 s
Cerveau	128x128x128	32259	60052	16907	47 s	96 s
Tête	128x128x128	36108	62006	1910	15 s	53 s
Champignon	86x100x56	15253	26392	714	2 s	14 s
Cavalier	39x25x25	2727	5526	1531	0.4 s	1.1 s
Cavalier	249x157x156	121540	246714	69026	47 s	251 s
Sphère	10x10x10	210	414	24	0.01 s	0.1 s
Sphère	50x50x50	6342	11646	0	0.6 s	4 s
Sphère	100x100x100	25746	46950	72	5 s	28 s
Sphère	200x200x200	103734	188382	0	50 s	198 s

1.3.4 Perspectives

La méthode présentée dans cette section est simple et efficace pour calculer les normales d'un objet voxel. De nombreux exemples ont été testés en utilisant des données réelles issues de l'imagerie médicale ; les algorithmes utilisés se sont révélés fiables et rapides. Cependant en 2001, G. Thürmer [Thu01] a proposé une autre technique de calcul, basée sur le volume présent au voisinage du point dont la normale est calculée, qui améliore son estimateur présenté en 1997 [TW97] (des détails sur cette méthode sont donnés au chapitre 3, section 3.3.2, page 107). Celle-ci s'est révélée plus rapide que celle présentée dans cette section (voir [TW97] pour des tests comparatifs) mais deux paramètres doivent être fixés qui induisent des résultats et des temps de calcul différents alors que notre méthode ne nécessite aucun paramètre. Cependant la technique proposée par G. Thürmer ne semble pas s'adapter aux objets fortement bruités alors que nous envisageons d'adapter la méthode présentée dans cette section aux objets bruités en utilisant les segments flous 2D et la notion de tangente discrète associée présentée dans le chapitre 2 de ce mémoire.

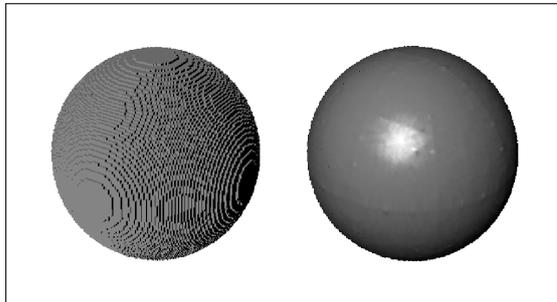


FIG. 1.14 – Sphère 50x50x50

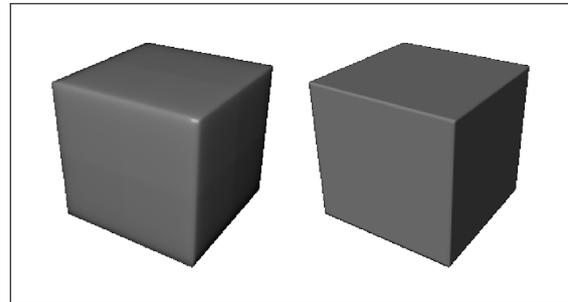


FIG. 1.15 – Cube rendu en 4 et 8-connextité

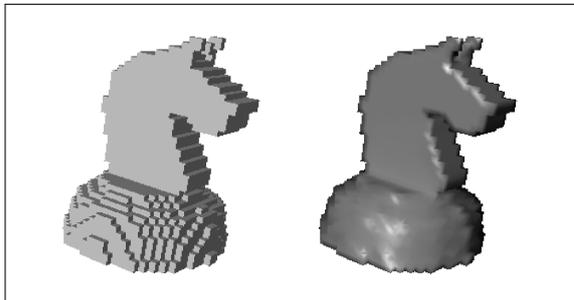


FIG. 1.16 – Cavalier 39x25x25

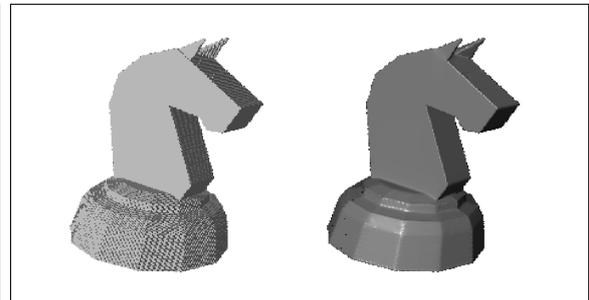


FIG. 1.17 – Cavalier 23.759x157x156

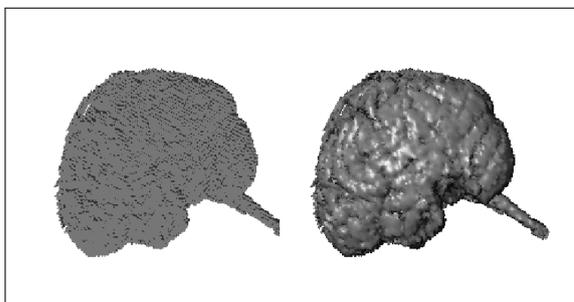


FIG. 1.18 – Cerveau 128x128x128

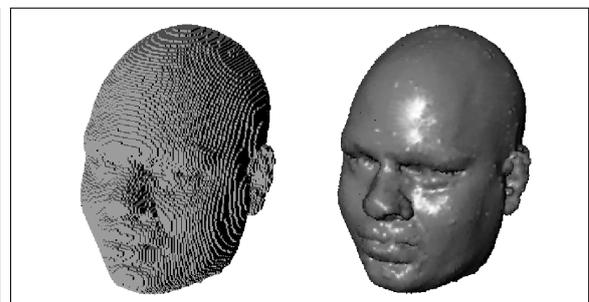


FIG. 1.19 – Tête 128x128x128

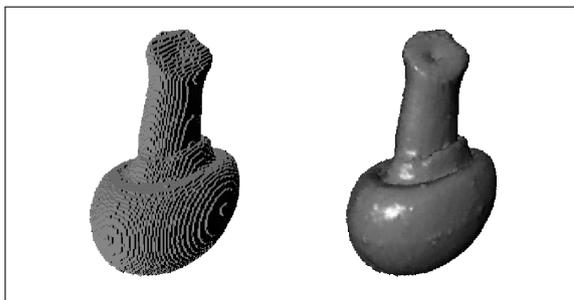


FIG. 1.20 – Champignon 86x100x56

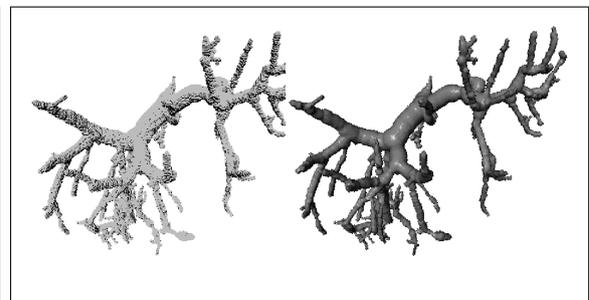


FIG. 1.21 – Veine porte du foie humain 266x285x241

1.4 Reconnaissance de morceaux de plans discrets

Je m'intéresse au problème de la reconnaissance des morceaux de plans discrets depuis ma thèse pendant laquelle j'avais proposé un algorithme incrémental de reconnaissance de morceaux rectangulaires de plans naïfs [20,43]. Rappelons la définition d'un plan discret :

Définition 1.18 *Un plan discret naïf est le sous-ensemble de points (x, y, z) de \mathbb{Z}^3 vérifiant la double inégalité $h \leq ax + by + cz < h + \max\{|a|, |b|, |c|\}$, avec $(a, b, c, h) \in \mathbb{Z}^4$.*

Le problème de la reconnaissance étudié dans cette section est le suivant : étant donné un sous-ensemble fini de \mathbb{Z}^3 , le problème est de déterminer s'il existe ou non un plan discret naïf le contenant.

Nous pouvons distinguer trois catégories de méthodes pour résoudre le problème de la reconnaissance de morceaux de plans discrets naïfs. Elles appartiennent aux domaines de l'arithmétique, de la programmation linéaire ou de la géométrie algorithmique. Un état de l'art assez complet de ces différentes méthodes est donné dans [BCK07] ainsi que dans ma thèse.

Avec Yan Gérard (LLAIC, Clermont Ferrand) et Paul Zimmermann (équipe SPACES, LORIA), nous avons proposé un algorithme [3] utilisant l'ensemble des cordes de l'ensemble étudié ainsi qu'une stratégie d'optimisation dans un ensemble de facettes triangulaires sur l'axe Ox .

L'algorithme de reconnaissance proposé utilise différentes définitions et propriétés qui sont énoncées dans le paragraphe suivant.

1.4.1 Définitions et premières propriétés

Les différentes notions définies dans ce paragraphe ont été développées dans le mémoire de thèse de Yan Gérard [Ger99].

Définition 1.19 *L'ensemble des cordes de $S \subset \mathbb{R}^n$ est l'ensemble des différences entre les points de S :*

$$\text{chord}(S) = S + (-S) = \{m' - m \mid m, m' \in S\}$$

Nous noterons $\text{chord}(S)$ cet ensemble (cf. Figure 1.22).

L'ensemble des cordes de S est la somme de Minkowski de S et $-S$. Soit $|S| = n$ alors le cardinal de $\text{chord}(S)$ est au plus $n(n-1) + 1$.

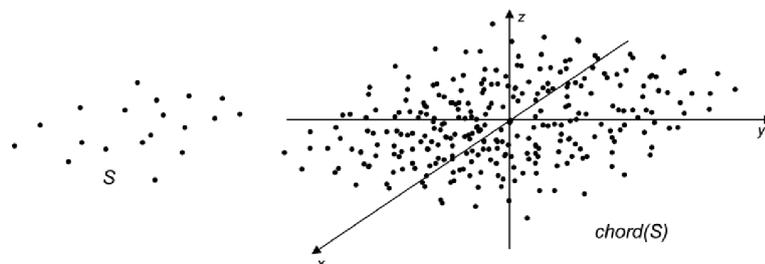


FIG. 1.22 – Un ensemble fini S , $|S| = 19$, et $\text{chord}(S)$.

L'enveloppe convexe d'un ensemble S de \mathbb{R}^n est notée par $\text{conv}(S)$. Nous pouvons remarquer la propriété suivante [Ger99] :

Proposition 1.8 $\forall S \subset \mathbb{R}^n, \text{conv}(\text{chord}(S)) = \text{chord}(\text{conv}(S))$.

L'ensemble convexe $\text{conv}(\text{chord}(S))$ est utilisé pour définir *l'épaisseur géométrique de S dans une direction*.

Définition 1.20 *L'épaisseur géométrique d'un ensemble fini $S \subset \mathbb{R}^n$ dans la direction Oz est la coordonnée en z du point du demi-axe Oz (avec $z \geq 0$) appartenant à la surface de l'enveloppe convexe de $\text{chord}(S)$. Elle sera notée $\tau_{Oz}(S)$ (cf. Figure 1.23).*

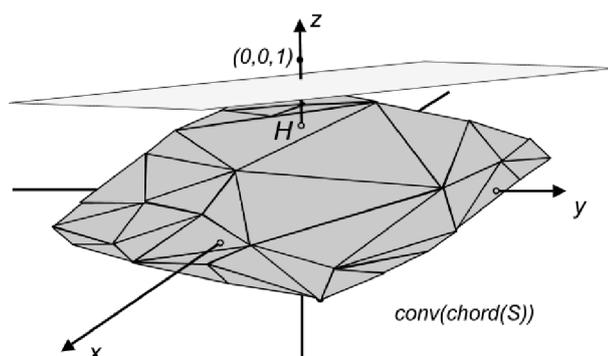


FIG. 1.23 – Enveloppe convexe de $\text{chord}(S)$: l'épaisseur de S dans la direction Oz est la coordonnée en z du point H .

Par la suite, nous appellerons *épaisseur de S* , l'épaisseur géométrique de S dans la direction Oz . Un résultat a ensuite été élaboré qui a permis d'élaborer une nouvelle stratégie pour la reconnaissance des morceaux de plans discrets naïfs, nous supposons par la suite que $c = \max\{|a|, |b|, |c|\}$:

Théorème 1.9 *Soit $S \subset \mathbb{Z}^3$ un ensemble fini, il existe $(a, b, c, h) \in \mathbb{Z}^4$ vérifiant $\forall (x, y, z) \in S, h \leq ax + by + cz < h + c$ si et seulement si $\tau_{Oz}(S) < 1$.*

La preuve de ce théorème figure dans l'article [3]. Ce théorème nous a permis de résoudre le problème de reconnaissance, noté *Pb1*, énoncé ci-dessous : Soit S un ensemble de points entiers, existe-t-il $(a, b, c, h) \in \mathbb{Z}^4$ tel que

$$\forall (x, y, z) \in S, h \leq ax + by + cz < h + c ?$$

Si l'épaisseur de S dans la direction Oz est supérieure ou égale à 1, alors le problème *Pb1* n'a pas de solution. La preuve du théorème 1.9 utilise la normale de la face de $\text{conv}(\text{chord}(S))$ qui coupe le demi-axe positif Oz pour trouver une solution car elle fournit un plan euclidien séparant strictement le point $(0, 0, 1)$ de $\text{conv}(\text{chord}(S))$. Tout plan respectant cette condition peut fournir une solution possible (cf. Figure 1.23), cependant, l'algorithme proposé ci-après recherche la solution obtenue à partir d'une face de $\text{conv}(\text{chord}(S))$ qui coupe le demi-axe positif Oz .

1.4.2 Algorithme de reconnaissance

Un algorithme naïf utilisant directement le théorème 1.9 permet de résoudre le problème *Pb1* en calculant l'enveloppe convexe de $\text{chord}(S)$ puis en recherchant les faces candidates. Cependant cette structure n'est pas nécessaire, nous proposons un algorithme qui parcourt des facettes triangulaires coupant le demi-axe positif Oz et formées de points de $\text{chord}(S)$.

Définition 1.21 Les 2-simplexes dont les sommets sont dans $\text{chord}(S)$ et qui coupent l'axe Oz sont appelés *triangles*.

La *hauteur* d'un triangle T est la coordonnée en z du point d'intersection entre T et Oz (si l'intersection n'est pas réduite à un point alors la hauteur est la coordonnée maximale en z).

Nous recherchons la face de $\text{conv}(\text{chord}(S))$ qui coupe le demi-axe Oz et contient le triangle le plus haut (un des triangles les plus hauts dans les cas dégénérés). La stratégie choisie est la suivante : considérons le triangle courant T , son vecteur normal \vec{n} est calculé (avec une coordonnée en z positive). Nous cherchons 2 points de S , m_{\min} et m_{\max} , tels que $\vec{n} \cdot m_{\min}$ et $\vec{n} \cdot m_{\max}$ soient respectivement minimal et maximal. $M = m_{\max} - m_{\min}$ est un point de $\text{chord}(S)$ qui maximise la forme linéaire $(\vec{n} \cdot)$ sur $\text{chord}(S)$. Deux cas sont possibles :

- M se situe sur le plan de T (cf. Figure 1.24). Dans ce cas, T est sur une face de $\text{conv}(\text{chord}(S))$. Si la hauteur de T est inférieure à 1, alors nous avons une solution : a , b et c sont les coordonnées du vecteur normal \vec{n} et h est donné par $\vec{n} \cdot m_{\min}$.

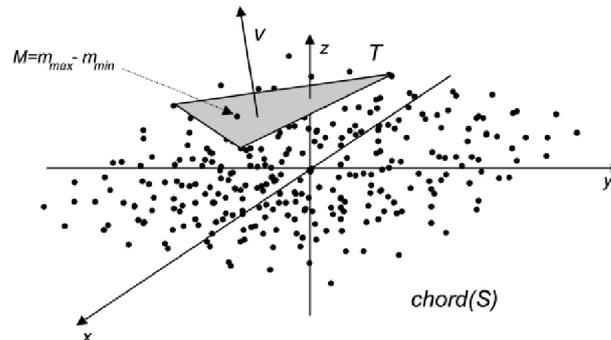
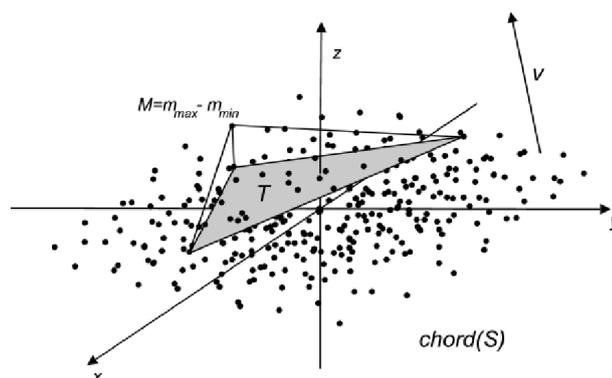


FIG. 1.24 – Cas où le point M appartient au plan du triangle courant T .

- sinon, les sommets de T définissent avec M un tétraèdre (cf. Figure 1.25). Par construction, une des 4 faces de ce tétraèdre est un triangle plus haut que T . Si sa hauteur est supérieure ou égale à 1, alors le problème n'a pas de solution. Sinon, le triangle le plus haut de ce tétraèdre est choisi comme nouveau triangle courant T .

Ce principe est utilisé dans l'Algorithme 3 qui permet de trouver le triangle le plus élevé. Quatre procédures y sont employées :

- **ChercherPremierTriangle** retourne une face de départ. Pour cela, un triangle T , tel que $T = (M_1, M_2, M_3) \in \text{chord}(S)$, est recherché dont la projection dans le plan Oxy contient l'origine. Une solution est de choisir $(M_1, M_2, M_3) = (m_1 - m_2, m_2 - m_3, m_3 - m_1)$ où m_1 , m_2 et m_3 sont des points de S non alignés. Cette opération est réalisée en $O(n)$ opérations.
- **Normal** calcule la normale de T orientée dans le sens des z croissants.
- **TrouverMinMax** retourne le point M de $\text{chord}(S)$ ayant un produit scalaire maximal avec la normale de T . $M = m_{\max} - m_{\min}$ avec m_{\min} et m_{\max} de S , tels que $\vec{n} \cdot m_{\min}$ et $\vec{n} \cdot m_{\max}$

FIG. 1.25 – Cas où le point M forment un tétraèdre avec T .

soient respectivement minimal et maximal. Cette recherche est aussi effectuée en $O(n)$ opérations.

- `TrouverTriangle` retourne le triangle du tétraèdre formé par T et M plus haut que T et coupant l'axe Oz .

Algorithme 3 : Algorithme de recherche du plus haut triangle

Entrée : un sous-ensemble fini S de \mathbb{Z}^3

Résultat : le plus haut triangle de $\text{chord}(S)$ coupant le demi-axe Oz

$\text{now}T = \text{ChercherPremierTriangle}(S);$

répéter

$T = \text{now}T;$

$\vec{n} = \text{Normal}(T);$

$M = \text{TrouverMinMax}(S, \vec{n});$

$\text{now}T = \text{TrouverTriangle}(T, M);$

jusqu'à $M \cdot \vec{n} \leq T \cdot \vec{n}$

Retourner T ;

Dans l'article [3], il est prouvé que l'Algorithme 3 est correct et se termine. Cet algorithme n'a pas une complexité théorique meilleure que celle en $O(|S|^6)$ de l'algorithme naïf calculant la hauteur de tous les triangles possibles. En effet, l'Algorithme 3 a une complexité en $O(\text{loop} * |S|)$ opérations où la variable *loop* contient le nombre de triangles T utilisés. Ce nombre est borné par $|S|^6$, ce qui conduit à une complexité en $O(|S|^7)$. Cependant, le comportement expérimental de cet algorithme est bien meilleur avec $\text{loop} = |S| + \text{constante}$ ($\text{constante} = 10$) et il semble linéaire en pratique. De plus le code du programme est court et élémentaire, il est téléchargeable aux adresses <http://www.loria.fr/~debled/plane> et <https://svn.liris.cnrs.fr/dcoeurjo/DigitalPlaneRecognition/trunk/YanGerardRecognition/> pour une version plus récente.

Une version incrémentale de cette méthode de reconnaissance a aussi été proposée et figure dans l'Algorithme 4. Cette version peut donc être utilisée dans le cadre de la segmentation d'un objet discret en morceaux de plans discrets naïfs.

Algorithme 4 : Algorithme incrémental de recherche du plus haut triangle**Entrée** : un sous-ensemble fini S de \mathbb{Z}^3 **Résultat** : le plus haut triangle de $\text{chord}(S)$ coupant le demi-axe Oz $T := (m_1 - m_2, m_2 - m_3, m_3 - m_1) = \text{ChercherPremierTriangle}(S);$ $\vec{n} = \text{Normal}(T);$ $S_0 = \{m_1, m_2, m_3\}; m_{\min} = m_1; m_{\max} = m_1;$ **pour** $m \in (S - S_0)$ **faire** $(T, \vec{n}, m_{\min}, m_{\max}) = \text{AjouterPoint}(T, \vec{n}, m_{\min}, m_{\max}, S_0, m);$ $S_0 = S_0 \cup \{m\};$ Retourner T ;**Fonction** $\text{AjouterPoint}(T, \vec{n}, m_{\min}, m_{\max}, S_0, m)$ **si** $m \cdot \vec{n} > m_{\max} \cdot \vec{n}$ **alors** $m_{\max} = m;$ **sinon** **si** $m \cdot \vec{n} < m_{\min} \cdot \vec{n}$ **alors** $m_{\min} = m;$ **tant que** $(m_{\max} - m_{\min}) \cdot v > T \cdot v$ **faire** $T = \text{TrouverTriangle}(T, m_{\max} - m_{\min});$ $\vec{n} = \text{Normal}(T);$ $(m_{\max}, m_{\min}) = \text{TrouverMinMax}'(S_0 \cup \{m\}, \vec{n});$ Retourner $T, \vec{n}, m_{\min}, m_{\max};$

La fonction $\text{TrouverMinMax}'$ de l'Algorithme 4 retourne deux points de S dont le produit scalaire avec \vec{n} est maximal et minimal. La fonction AjouterPoint teste si le point $m \in S$ modifie le plus haut triangle et, si c'est le cas, met à jour m_{\max} et m_{\min} puis cherche le triangle le plus haut dans $S_0 \cup \{m\}$.

Nous nous sommes aussi intéressés aux caractéristiques des plans naïfs obtenus après le processus de reconnaissance que nous avons proposé et nous avons exhibé une condition pour que les caractéristiques (a, b, c, h) obtenues soient optimales dans le sens $|c|$ minimal :

Proposition 1.10 *Soient S un sous-ensemble de points entiers de \mathbb{Z}^3 et T_{\max} le triangle le plus haut obtenu par les Algorithmes 3 ou 4, et (a, b, c, h) les caractéristiques du plan naïf déduit, la valeur c est minimale si $\left\lceil \frac{1}{1-\tau_z(S)} \right\rceil = c$.*

1.4.3 Perspectives

Dans cette section nous avons proposé un algorithme de reconnaissance de morceaux de plans discrets naïfs de complexité théorique élevée mais rapide en pratique.

Une des perspectives envisagées est l'adaptation de l'algorithme présenté dans cette section à la reconnaissance de morceaux de plans discrets épais, avec des points manquants. Ces objets discrets sont appelés morceaux flous de plans discrets et ils sont définis au chapitre 3, section 3.2 de ce mémoire.

1.5 Paraboles discrètes

Avec Damien Jamet et Eric Domenjoud, membres de l'équipe ADAGIo, nous avons proposé en 2006 une nouvelle définition de paraboles discrètes, appelées **paraboles discrètes arithmétiques** [9]. Les motivations de ce travail sont, d'une part, la recherche de nouvelles primitives discrètes permettant de définir une géométrie discrète non linéaire et, d'autre part, une approche de la différentiabilité discrète. Dans ce travail, nous avons transposé aux paraboles discrètes l'idée proposée par Damien Jamet, Jean-Luc Toutant et Christophe Fiorio dans [FJT06] pour définir les cercles discrets avec une épaisseur reposant sur une norme du gradient de son équation implicite.

Ce travail s'inscrit dans la continuité des travaux engagés pour décrire de manière analytique des classes de primitives discrètes. Différents modèles ont été proposés et étudiés [CM91, ANF97, TWR99, And00, BAB00, LW00b, LW00a] : le modèle supercouverture, le modèle standard, le modèle naïf fermé, le modèle pythagoricien fermé, les modèles basés sur la distance de Hausdorff.

Dans [FJT06], une extension aux cercles discrets de la définition de la droite discrète arithmétique proposée par Jean-Pierre Reveillès est donnée et nous la rappelons ci-dessous.

Définition 1.22 Soient $X \in \mathbb{Z}^2$, $R \in \mathbb{N}$, $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ une fonction définie par $f(x) = \|x - X\|_2^2 - R^2$ et $w : \mathbb{R}^2 \rightarrow \mathbb{R}$. Le **cercle discret arithmétique** de centre $X \in \mathbb{Z}^2$, de rayon $R \in \mathbb{N}$ et d'épaisseur w est le sous-ensemble $C(X, R, w)$ défini par :

$$C(X, R, w) = \left\{ x \in \mathbb{Z}^2, -\frac{w(x)}{2} \leq f(x) < \frac{w(x)}{2} \right\}$$

1.5.1 Définition

Dans le même esprit, nous nous sommes intéressés aux paraboles discrètes et nous avons proposé des résultats concernant les paraboles discrètes droites, c'est-à-dire celles dont l'axe est vertical. Les **paraboles discrètes arithmétiques** sont définies de la manière suivante :

Définition 1.23 Soit $f : \mathbb{R} \rightarrow \mathbb{R}$ une fonction polynomiale de degré 2 à une inconnue et soit $\tilde{f} : \mathbb{R}^2 \rightarrow \mathbb{R}$ définie par $\tilde{f}(\mathbf{x}) = f(x_1) - x_2$. On considère la fonction $w : \mathbb{R}^2 \rightarrow \mathbb{R}$, une **parabole discrète arithmétique** d'équation $\tilde{f}(\mathbf{x}) = 0$ et d'épaisseur w est le sous-ensemble $P(\tilde{f}, w)$ de \mathbb{Z}^2 satisfaisant :

$$P(\tilde{f}, w) = \left\{ \mathbf{x} \in \mathbb{Z}^2, -\frac{w(\mathbf{x})}{2} \leq \tilde{f}(\mathbf{x}) < \frac{w(\mathbf{x})}{2} \right\}$$

Cette définition permet de tester très simplement si un point appartient à une parabole discrète arithmétique donnée.

Toutes les fonctions d'épaisseur w peuvent être utilisées mais nous avons considéré plus spécifiquement certaines fonctions ; celles homothétiques à $\|\nabla \tilde{f}\|_\alpha$, avec $\alpha \in [1, +\infty]$ et :

$$\begin{aligned} \nabla \tilde{f} &: \mathbb{R}^2 \longrightarrow \mathbb{R}^2 \\ \mathbf{x} &\longmapsto (f'(x_1), -1) \end{aligned}$$

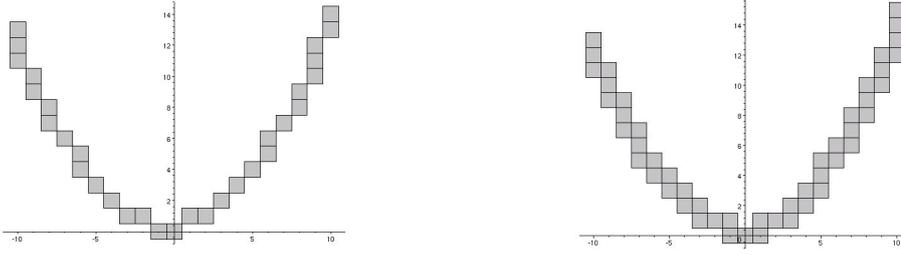


FIG. 1.26 – Une parabole discrète naïve — Une parabole discrète standard pour $\tilde{f}(\mathbf{x}) = 7x_1^2 + 4x_1 + 21 - 56x_2$.

Nous rappelons que pour $\alpha \in [1, +\infty[$,

$$\begin{aligned} \|\cdot\|_\alpha : \mathbb{R}^2 &\longrightarrow \mathbb{R}_+ & \text{et} & \quad \|\cdot\|_\infty : \mathbb{R}^2 &\longrightarrow \mathbb{R}_+ \\ \mathbf{x} &\longmapsto (|x_1|^\alpha + |x_2|^\alpha)^{\frac{1}{\alpha}}, & & \quad \mathbf{x} &\longmapsto \max\{|x_1|, |x_2|\}. \end{aligned}$$

Pour plus de clarté, nous introduisons les notations suivantes :

Soit $f : \mathbb{R} \rightarrow \mathbb{R}$ une fonction polynomiale de degré 2 dans $\mathbb{R}[x]$: il existe $(a, b, c) \in \mathbb{R}^3$ tel que, pour tout $x \in \mathbb{R}$, $f(x) = ax^2 + bx + c$. Soit $\alpha \in [1, +\infty]$ et $\lambda \in \mathbb{R}_+$. La **parabole discrète arithmétique** d'équation f d'épaisseur $\|\cdot\|_\alpha$ et de facteur d'épaississement λ est l'ensemble $P(f, \lambda \| \nabla f \|_\alpha)$. Nous noterons $P(f, \alpha, \lambda)$ la parabole $P(f, \lambda \| \nabla f \|_\alpha)$.

Nous avons nommé les paraboles discrètes arithmétiques $P(f, \infty, 1)$ (resp. $P(f, 1, 1)$) **paraboles discrètes naïves** (resp. **paraboles discrètes standards**). Plus précisément,

$$\begin{aligned} P(f, \infty, 1) &= \left\{ (x_1, x_2) \in \mathbb{Z}^2, -\frac{\max(|2ax_1 + b|, 1)}{2} \leq ax_1^2 + bx_1 + c - x_2 < \frac{\max(|2ax_1 + b|, 1)}{2} \right\} \\ P(f, 1, 1) &= \left\{ (x_1, x_2) \in \mathbb{Z}^2, -\frac{|2ax_1 + b| + 1}{2} \leq ax_1^2 + bx_1 + c - x_2 < \frac{|2ax_1 + b| + 1}{2} \right\} \end{aligned}$$

La figure 1.26 illustre ces deux définitions.

1.5.2 Etude de la structure des paraboles discrètes arithmétiques

Nous avons ensuite étudié les propriétés structurelles et topologiques ainsi que le comportement de l'épaississement par le facteur λ des paraboles discrètes arithmétiques, ces résultats sont décrits succinctement ci-après.

Comme pour les droites discrètes, nous pouvons définir des **courbes d'appui** pour les paraboles discrètes. Considérons le cas de $P(f, \infty, \lambda)$. Alors, pour tout $\mathbf{x} \in \mathbb{Z}^2$,

$$\mathbf{x} \in P(f, \infty, \lambda) \iff -\lambda \frac{\max\{|2ax_1 + b|, 1\}}{2} \leq f(x_1) - x_2 < \lambda \frac{\max\{|2ax_1 + b|, 1\}}{2},$$

et l'ensemble $P(f, \infty, \lambda)$ est borné par les 4 courbes d'équation : $f(x_1) - x_2 = \pm \lambda \frac{2ax_1 + b}{2}$ et $f(x_1) - x_2 = \pm \frac{\lambda}{2}$. Plus précisément, les **courbes d'appui supérieure et inférieure** sont

constituées respectivement de 3 morceaux de paraboles (cf. Figure 1.27) :

$$l_{\infty}^U(x_1) = \begin{cases} f(x_1) - \frac{\lambda}{2}(2ax_1 + b), & x_1 \in]-\infty, \frac{-1-b}{2a}[\\ f(x_1) + \frac{\lambda}{2}, & x_1 \in [\frac{-1-b}{2a}, \frac{1-b}{2a}] \\ f(x_1) + \frac{\lambda}{2}(2ax_1 + b), & x_1 \in [\frac{1-b}{2a}, \infty[\end{cases}$$

$$l_{\infty}^L(x_1) = \begin{cases} f(x_1) + \frac{\lambda}{2}(2ax_1 + b), & x_1 \in]-\infty, \frac{-1-b}{2a}[\\ f(x_1) - \frac{\lambda}{2}, & x_1 \in [\frac{-1-b}{2a}, \frac{1-b}{2a}] \\ f(x_1) - \frac{\lambda}{2}(2ax_1 + b), & x_1 \in [\frac{1-b}{2a}, \infty[\end{cases}$$

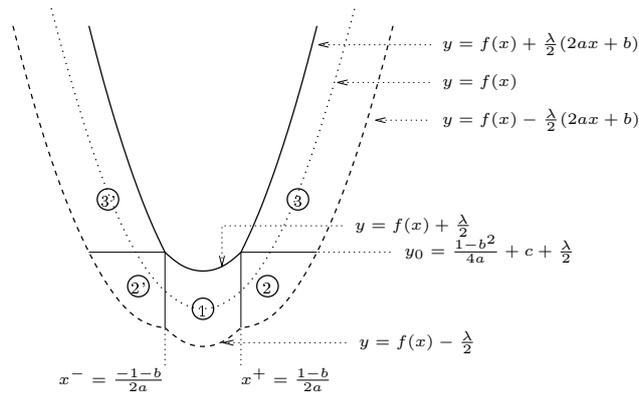


FIG. 1.27 – Partitionnement de $P(f, \infty, \lambda)$ en 5 domaines.

De même, on obtient que l'ensemble $P(f, 1, \lambda)$ est borné par les courbes d'équation $f(x_1) - x_2 = \pm \lambda \frac{(2ax_1+b+1)}{2}$ (cf. Figure 1.28) et que $P(f, 2, \lambda)$ est borné par les courbes d'équation $f(x_1) - x_2 = \pm \lambda \frac{\sqrt{(2ax_1+b)^2+1}}{2}$.

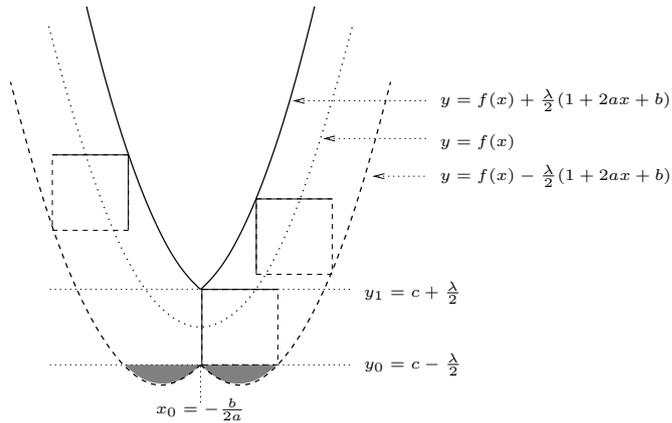


FIG. 1.28 – Partitionnement de $P(f, 1, \lambda)$.

Le théorème suivant sur la **connexité des paraboles discrètes** a été démontré dans [7] :

Théorème 1.11 Soient $f : \mathbb{R} \rightarrow \mathbb{R}$ une fonction polynomiale de degré 2 , $\alpha \in [1, +\infty]$ et $\lambda \in [1, +\infty[$.

- (i) La parabole discrète arithmétique $P(f, \alpha, \lambda)$ est 0-connectée.
(ii) La parabole discrète arithmétique $P(f, 1, \lambda)$ est 1-connectée.

Nous avons remarqué que pour une parabole discrète donnée $P(f, \alpha, \lambda)$, **le paramètre d'épaisseur λ pouvait engendrer une déformation de la représentation graphique de P** (cf. Figure 1.27 et Figure 1.28). Nous avons donc fait une étude plus précise et nous proposons des bornes sur les valeurs de λ en fonction de la valeur de α :

- Si $\alpha = \infty$, une condition suffisante pour obtenir une *bonne* représentation graphique est que l'abscisse du sommet de la première parabole définissant la courbe d'appui inférieure de P soit plus grande que $\frac{-1-b}{2a}$. Cette condition correspond à $\lambda \leq \frac{1}{a}$ (cf. Figure 1.29).
- Si $\alpha = 1$, une condition suffisante pour obtenir une *bonne* représentation graphique est que la différence entre l'ordonnée du sommet d'une parabole définissant la courbe d'appui inférieure de P et l'ordonnée du point d'intersection des deux paraboles définissant les courbes d'appuis inférieure est inférieure ou égale à 0.5. Cette condition correspond à $\lambda^2 \leq \frac{2a-b^2}{a^2}$ (cf. Figure 1.29).

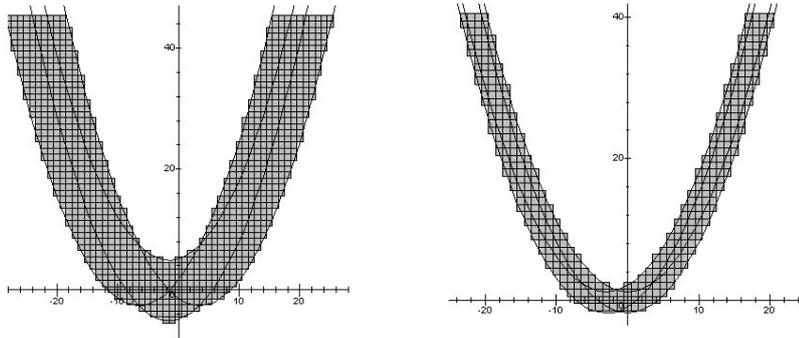


FIG. 1.29 – A gauche $P(f, \infty, 9)$ et à droite $P(f, 1, 3)$ avec $f(x) = 0.1x^2 + 0.3x + 0.1$.

1.5.3 Algorithme de construction de $P(f, \alpha, 1)$

L'algorithme de construction des paraboles discrètes $P(f, \alpha, 1)$ proposé utilise les propriétés de connexité des paraboles discrètes arithmétiques (cf. théorème 1.11). Les paraboles $P(f, \alpha, 1)$ telles que $\tilde{f}(\mathbf{x}) = ax_1^2 + bx_1 + c + dx_2$ avec $(a, b, c, d) \in \mathbb{Z}^4$, $a > 0$ et $d < 0$ sont considérées. Nous remarquons que :

$$\begin{aligned}\tilde{f}(x_1 + 1, x_2) &= \tilde{f}(x_1, x_2) + 2ax_1 + a + b \\ \tilde{f}(x_1, x_2 \pm 1) &= \tilde{f}(x_1, x_2) \pm d\end{aligned}$$

L'Algorithme 5 est déduit de cette remarque.

Algorithme 5 : Algorithme incrémental de construction de $P(f, \alpha, 1)$

Entrées : a, b, c, d les caractéristiques de $P(f, \alpha, 1)$, N le nombre de points, (p_x, p_y) le premier point

Initialisation : $x = p_x$; $y = p_y$; $F = ax^2 + bx + c + dy$; $n = 0$;

Afficher le pixel (x, y) ;

tant que $n \leq N$ **faire**

si $x < -\frac{b}{2a}$ **alors** $s_y = -1$; **sinon** $s_y = 1$;

si $F + 2ax + a + b \geq -\frac{\|\nabla \tilde{f}\|_{\alpha}(x+1, y)}{2}$ **et** $F + 2ax + a + b < \frac{\|\nabla \tilde{f}\|_{\alpha}(x+1, y)}{2}$ **alors**

$F = F + 2ax + a + b$;

$x = x + 1$;

sinon

si $F + s_y d \geq -\frac{\|\nabla \tilde{f}\|_{\alpha}(x, y+1)}{2}$ **et** $F + s_y d < \frac{\|\nabla \tilde{f}\|_{\alpha}(x, y+1)}{2}$ **alors**

$F = F + s_y d$;

$y = y + s_y$;

sinon

$F = F + 2ax + a + b + s_y d$;

$x = x + 1$; $y = y + s_y$;

fin

fin

 Afficher le pixel (x, y) ;

$n = n + 1$;

fin

Nous nous sommes aussi intéressés à la construction des paraboles discrètes épaisses $P(f, \alpha, \lambda)$ et l'idée principale est d'utiliser les 2 courbes d'appui de $P(f, \alpha, \lambda)$. Pour chaque valeur de d'abscisse, nous calculons les points réels situés sur les courbes d'appui inférieures et supérieures de P . Nous en déduisons alors incrémentalement la suite des points entiers de même abscisse situés entre ces deux points.

1.5.4 Perspectives

Ce premier travail sur les paraboles discrètes propose une définition avec une épaisseur non constante ainsi qu'une étude de la structure de ces nouveaux objets avec des propriétés intéressantes de topologie. Il doit être poursuivi par l'étude de la reconnaissance de ces objets discrets. De plus, nous souhaitons généraliser cette étude à l'ensemble des coniques. Les problèmes de déformation rencontrés pour une valeur élevée de λ nous amènent aussi à réfléchir à d'autres fonctions d'épaisseur.

1.6 Conclusion

Les travaux relatés dans ce chapitre utilisent des objets *réguliers* de la géométrie discrète comme les droites, les plans ou encore les paraboles. Les propriétés arithmétiques et combinatoires

de ces primitives discrètes m'ont permis d'obtenir des algorithmes performants pour résoudre des problèmes concernant la décomposition polygonale, la reconnaissance ou encore l'élaboration d'estimateurs de paramètres géométriques sur les courbes discrètes.

Dans les chapitres suivants, ces propriétés intéressantes des primitives discrètes droites et plans seront utilisées dans le cadre de l'élaboration d'une géométrie pour les objets discrets bruités.

Chapitre 2

Eléments de géométrie discrète pour l'étude des courbes bruitées

Sommaire

2.1	Introduction	57
2.1.1	Le bruit dans les images	58
2.1.2	Limites des outils existants de géométrie discrète	60
2.2	Segments flous arithmétiques	61
2.2.1	Approche arithmétique	62
2.2.2	Approche géométrique	68
2.2.3	Approximation polygonale multi-niveaux de courbes discrètes	74
2.2.4	Extension en dimension 3	78
2.3	Courbure discrète et autres paramètres géométriques	80
2.3.1	Tangente discrète	80
2.3.2	Courbure discrète	84
2.3.3	Segments flous maximaux et optimisation du calcul de la courbure	87
2.3.4	Application à la détection de cercles et arcs de cercles dans des images techniques	90
2.3.5	Estimateur de la longueur d'une courbe discrète bruitée	94
2.3.6	Extension en dimension 3	95
2.4	Conclusion	97

2.1 Introduction

Ce chapitre est le plus conséquent de mon mémoire, il contient l'ensemble de mes travaux réalisés dans le cadre de l'élaboration d'une géométrie discrète des courbes bruitées. Ceux-ci ont donné lieu à des encadrements, des collaborations et des publications que je résume ci-après.

Ce chapitre commence par une introduction sur le bruit présent dans les images avec l'évocation de solutions proposées dans le milieu du traitement d'image. Puis je montre certains problèmes rencontrés par les techniques développées dans le cadre de la géométrie discrète sur des données bruitées. Ceci m'a conduit à proposer la notion de *segments flous* permettant de tenir compte du bruit présent dans les contours d'objets discrets 2D.

Dans la section 2, je présente d'abord *l'approche arithmétique* des segments flous. Ce travail [2,16,40] a été réalisé avec Jocelyne Rouyer-Degli et Jean-Luc Rémy (équipe ADAGE/ADAGIo, Loria). Puis j'introduis *l'approche géométrique* [1,13], conçue en collaboration avec Fabien Feschet (LAIC, Clermont-Ferrand) et Jocelyne Rouyer-Degli. Ces travaux ont été ensuite utilisés dans le cadre d'une application en traitement d'image où la suite des segments principaux du contour d'un objet d'une image sont extraits automatiquement sans paramètre ; ce travail [4,14,26] a été réalisé en collaboration avec Antoine Tabbone et Laurent Wendling de l'équipe QGAR du Loria. A la fin de la section 2, je propose une extension de ces travaux à la dimension 3 pour laquelle j'ai encadré Franck Rapaport pendant son stage de DEA [Rap04].

Dans la section 3, j'utilise les travaux de la section 2 afin d'estimer des caractéristiques géométriques sur des courbes pouvant être bruitées. Un premier travail sur la notion de *tangente discrète* d'ordre d est présenté [15,34,36]. Puis un *estimateur de courbure* est proposé avec une méthode optimisant son calcul, j'ai encadré Thanh Phuong Nguyen pendant son stage de DPEA [Ngu05] puis pendant sa thèse sur ce travail [8]. Dans le cadre d'une application [11], ce travail a été utilisé et adapté à la reconnaissance de cercles et de segments dans des documents techniques en collaboration avec Jean-Pierre Salmon (doctorant dans l'équipe QGAR) et Laurent Wendling. Enfin, je propose un estimateur de longueur de courbes discrètes pouvant être bruitées ainsi qu'une extension à la dimension 3 pour l'ensemble de ces estimateurs.

Les travaux présentés dans ce mémoire concernant les courbes discrètes de dimension 3 n'ont pas encore été publiés.

2.1.1 Le bruit dans les images

Une image peut être altérée par du bruit. De nombreuses sources de bruit sont possibles comme par exemple :

- le mode d'acquisition (appareil photo numérique, scanner, échographie, IRM, ...) et son contexte (éclairage, poussière sur l'objectif, ...),
- l'échantillonnage,
- la quantification,
- la transmission des images peut aussi perdre ou corrompre certaines données.

En fonction des problèmes détectés, des modèles de bruit [CP95] ont été élaborés comme par exemple le *bruit impulsif* qui est défini par une densité de probabilité de la forme $f(a) = C.e^{-K|a|^\alpha}$ et tend à modéliser le bruit à l'acquisition ou encore le *bruit poivre et sel* qui modélise la poussière ou de petits objets sur une pellicule ou sur un scanner. Des filtres associés à ces différents types de bruit permettent de restaurer les images originales mais sur des images naturelles, ils apportent aussi des défauts. Des opérations comme l'extraction de régions ou de contours, de squelettisation, de calculs de paramètres géométriques des régions de l'image (aire, périmètre, diamètre, élongation, ...) sont ensuite réalisées sur les images obtenues. Ces techniques utilisent de nombreux paramètres et des seuils doivent être fixés par l'utilisateur selon la nature et la connaissance de l'image traitée.

Une autre approche consiste à utiliser la théorie des ensembles flous [Zad65] et de la géométrie floue [Ros84] qui permet de considérer et de gérer la nature incertaine des données sans la supprimer ou l'interpréter. Cette ambiguïté peut intervenir dans les niveaux de gris, la géométrie des objets de l'image ou encore dans le manque de connaissance de l'utilisateur sur le contenu de l'image.

Une des approches standards de l'analyse d'image commence par la segmentation de l'image en régions et le calcul de propriétés et de relations entre ces régions. Comme les régions d'une image n'ont pas forcément des bords bien définis, il est souvent plus approprié de les considérer comme des ensembles flous [Zad65].

Un sous-ensemble flou de l'ensemble \mathcal{E} est défini par une fonction μ de \mathcal{E} dans $[0, 1]$. Ceci généralise la définition d'un sous-ensemble *classique* S de \mathcal{E} qui peut être défini par une fonction μ de \mathcal{E} dans $\{0, 1\}$, avec $S = \{P \in \mathcal{E} \mid \mu(P) = 1\}$. Les opérateurs et les relations standards de la théorie des ensembles s'étendent aux sous-ensembles flous : \subseteq devient \leq , \cup et \cap deviennent *sup* et *inf*.

A partir de cette notion, A. Rosenfeld a introduit le concept de la géométrie discrète floue [Ros84]. Des définitions et des propriétés de base ont été établies comme la connectivité, la convexité, les mesures d'aire, de périmètre et de diamètre ainsi que de compacité d'un objet d'une image à niveaux de gris [Ros84, RH84, Ros90, SNS03, Sven07].

Algorithme 6 : Algorithme pour *bruiter* une courbe discrète

Données : \mathcal{C} une courbe discrète de \mathbb{Z}^2 constituée de S_1, \dots, S_m , une suite de segments discrets issus de la discrétisation OBQ de segments réels ; l un nombre entier correspondant à l'espace entre chaque point devant être bruité ; h un réel permettant de contrôler l'amplitude du bruit généré.

Résultat : $\mathcal{C}_{l,h}$ la courbe discrète bruitée de \mathcal{C} .

pour i de 1 à m **faire**

 Soit Δ_i le segment de \mathbb{R}^2 dont les points extrémités sont ceux de S_i ;

 Choisir k points $M_{1\Delta_i}, \dots, M_{k\Delta_i}$ régulièrement espacés sur Δ_i tels que pour tout j de 0 à k , $l = \max(|x(M_{j+1\Delta_i}) - x(M_{j\Delta_i})|, |y(M_{j+1\Delta_i}) - y(M_{j\Delta_i})|)$ ($M_{0\Delta_i}$ et $M_{k+1\Delta_i}$ sont les extrémités de S_i) ;

pour chaque $M_{j\Delta_i}$ **faire**

 Choisir une valeur aléatoire d entre 0 et h ; (*)

 Déplacer $M_{j\Delta_i}$ orthogonalement à Δ_i à une distance d ; (**)

pour i de 2 à m **faire**

 Déplacer $M_{0\Delta_i}$ selon les étapes (*) et (**);

Relier par des segments l'ensemble des points obtenus ;

Faire une discrétisation OBQ des segments obtenus $\rightarrow \mathcal{C}_{l,h}$;

Nous avons cependant utilisé une autre approche reposant sur des définitions arithmétiques d'objets discrets et correspondant mieux à nos objets d'étude. En effet, nous nous intéressons dans ce chapitre à l'étude des courbes discrètes pouvant correspondre aux bords d'objets dans des images ou encore à des squelettes obtenus à la fin d'une chaîne de traitement d'un document. Du bruit peut être présent dans ces données. Dans le cadre de la géométrie discrète, de nombreux outils ont été développés pour l'étude des courbes discrètes (par exemple [7] et [FT99, CMT01, CS03, CK04]) qui, comme je le montre dans la section suivante, ne s'adaptent pas toujours au bruit présent dans les objets à analyser. Mon objectif est de proposer dans ce chapitre des outils adaptés à l'analyse de courbes discrètes plus ou moins bruitées.

Concernant la squelettisation de formes issues de documents techniques, Xavier Hilaire [Hil04] a formulé deux hypothèses sur le bruit présent dans ces squelettes : il est de nature quasi uniforme et la proportion de points invalides sur les segments et les arcs de cercle est inférieure à 50% pour des documents numérisés dans des conditions moyennes. Il a utilisé dans son travail le modèle de bruit de T. Kanungo [KHBSM00] qui a été établi dans le contexte de l'OCR ⁷ et qui est un des rares modèles validés. Dans cet esprit, il a élaboré un algorithme (cf. Algorithme 6) pour bruitez des courbes discrètes que j'ai utilisé dans mes travaux sur les courbes discrètes 2D. Ces travaux sont décrits dans ce chapitre.

2.1.2 Limites des outils existants de géométrie discrète

La reconnaissance exacte des segments de droites discrètes naïves [7] a été utilisée dans de nombreuses applications comme, par exemple, la segmentation de courbes discrètes ou encore la détection de paramètres géométriques comme la courbure, la longueur, . . . Cependant, dans le cadre applicatif, les courbes discrètes étudiées contiennent souvent des irrégularités induites par les outils utilisés pour les obtenir. Sur ce type de courbes, les algorithmes de segmentation issus des travaux de géométrie discrète ne fournissent pas de résultats satisfaisants comme nous pouvons le constater à la figure 2.1, la segmentation obtenue contient beaucoup trop de segments alors que la courbe ne semble pas bruitée.

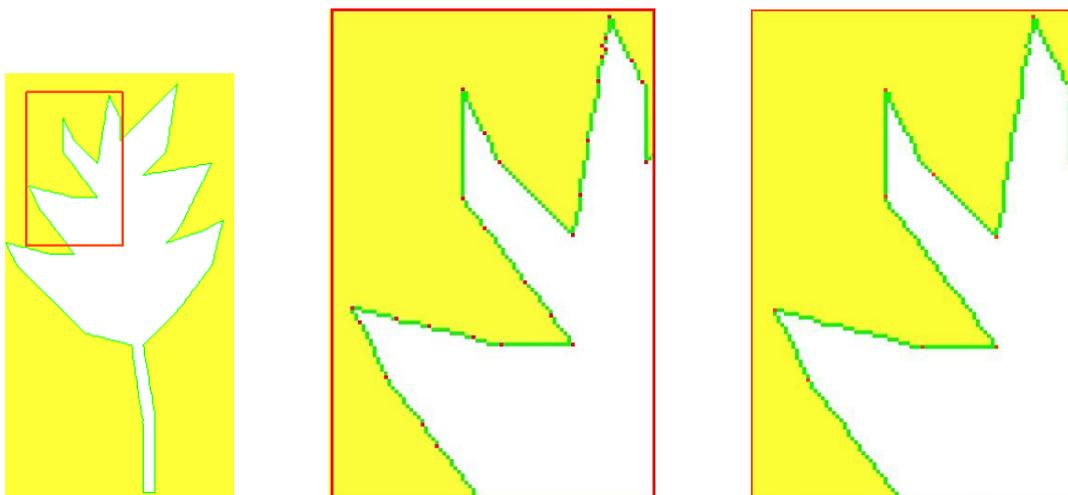


FIG. 2.1 – De gauche à droite : un exemple de courbe discrète obtenue dans un contour de l'image d'une feuille; agrandissement sur une partie du contour et segmentation obtenue avec un algorithme de reconnaissance de segments de droites naïves; segmentation souhaitée sur la même partie de la courbe.

Dans le cas de courbes fortement bruitées, il est clair que l'algorithme de reconnaissance de segments de droites discrètes naïves produira de très petits segments (par exemple sur la courbe bruitée de la figure 2.2). En traitement d'images, ce problème de sur-segmentation a été beaucoup étudié en utilisant une multitude de méthodes différentes qui peuvent être classées essentiellement selon 2 types d'approche :

- la détection de points critiques (par exemple [WB91, RC92]),
- l'approximation polygonale avec des seuil de tolérance à définir (par exemple [WD84, RW95, LR02]).

⁷Optical Character Recognition

La gestion du bruit présent dans la courbe est faite en utilisant des informations sur la nature des documents étudiés et sur la nature du bruit présent [Bai00] en générant ainsi des contraintes dans la segmentation.

Dans la continuité des travaux de géométrie discrète analytique, pour résoudre ce type de problème, nous avons eu l'idée de découper la courbe en droites épaisses qui permettaient d'obtenir ainsi de plus long segments (cf. Figure 2.2). C'est pourquoi, nous avons introduit un nouvel objet discret, le *segment flou arithmétique* reposant sur la définition d'une droite discrète arithmétique d'épaisseur variable.

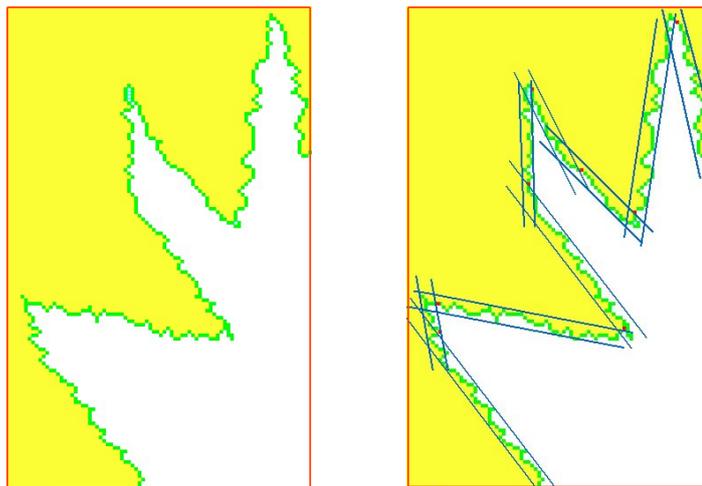


FIG. 2.2 – De gauche à droite : un exemple de courbe discrète fortement bruitée ; segmentation souhaitée.

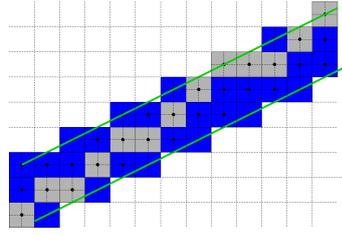
2.2 Segments flous arithmétiques

Nous avons défini la notion de *segment flou arithmétique 2D*, en relation avec la définition arithmétique des droites discrètes [Rev91] où l'épaisseur est paramétrable. Au début de nos travaux sur ce sujet, nous avons défini un segment flou comme une suite 8-connexe de points qui appartiennent à une droite discrète dont l'épaisseur est donnée. Un paramètre permet de contrôler l'amplitude du bruit autorisé en fixant l'épaisseur de la droite englobant le segment flou.

Par la suite, nous avons été amenés à faire évoluer la définition de segment flou arithmétique en supprimant la condition de connexité. Plusieurs de mes travaux [1,2,11,13,14,15,16] ont utilisé cette notion décrite ci-dessous.

Définition 2.1 Soit Sf une suite de points de \mathbb{Z}^2 . Une droite discrète $\mathcal{D}(a, b, \mu, \omega)$ est dite *englobante* pour Sf si tous les points de Sf appartiennent à \mathcal{D} (cf. Figure 2.3).

Parmi l'ensemble des droites englobantes d'une suite de points, nous nous intéressons à celles qui approchent *au plus près* l'ensemble des points de Sf . Pour cela, deux notions ont été définies, l'une arithmétique et l'autre géométrique dont les définitions sont données ci-dessous ainsi que les algorithmes résultants.

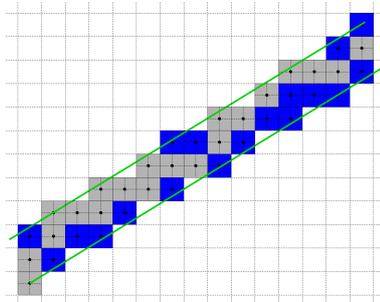
FIG. 2.3 – $\mathcal{D}(1, 2, -4, 6)$, droite englobante de la suite de points grisés.

2.2.1 Approche arithmétique

L'approche arithmétique suit une démarche similaire à celle employée pour la reconnaissance des segments de droites naïves [7,43], elle utilise les mêmes notions de *points d'appui*⁸ et de *pointillés de niveau* α ⁹ qui peuvent être transposées aux droites discrètes quelconques. Ce travail [2,16] a été effectué en collaboration avec Jocelyne Rouyer et Jean-Luc Rémy. Les démonstrations du théorème et de la proposition figurant dans ce paragraphe peuvent être consultées dans l'article [2] figurant en annexe (partie III) de ce mémoire.

Pour simplifier l'écriture, nous supposons par la suite que *les coefficients de la pente de la droite* $\mathcal{D}(a, b, \mu, \omega)$ *vérifient* $0 \leq a \leq b$ (*premier octant*).

Définition 2.2 Soient Sf une suite de points du premier octant définie dans l'intervalle des abscisses $[0, l - 1]$ et $\mathcal{D}(a, b, \mu, \omega)$ une droite englobante de Sf . \mathcal{D} est dite *strictement englobante pour* Sf si, \mathcal{D} possède au moins trois points d'appui dans l'intervalle $[0, l - 1]$ et si, Sf contient au moins un point d'appui inférieur et un point d'appui supérieur de \mathcal{D} (cf. Figure 2.4).

FIG. 2.4 – $\mathcal{D}(8, 13, -6, 32)$ droite strictement englobante de la suite de points grisés dans l'intervalle des abscisses $[-5, 9]$.

Définition 2.3 Sf est un *segment flou d'ordre* d si et seulement si il existe une droite discrète $\mathcal{D}(a, b, \mu, \omega)$ strictement englobante pour Sf telle que $\frac{\omega}{\max(|a|, |b|)} \leq d$.

L'ordre du segment flou permet de limiter l'épaisseur de la droite discrète englobant la suite 8-connective de points d'un segment flou, et ainsi, de contrôler la hauteur du palier vertical de la droite englobante.

⁸Les points d'appui de la droite discrète $\mathcal{D}(a, b, \mu, \omega)$ sont les points entiers qui appartiennent à une des droites réelles $ax - by = \mu$ ou $ax - by = \mu + \omega - 1$

⁹Le pointillé de niveau α de la droite discrète $\mathcal{D}(a, b, \mu, \omega)$ est la droite réelle $ax - by = \alpha$

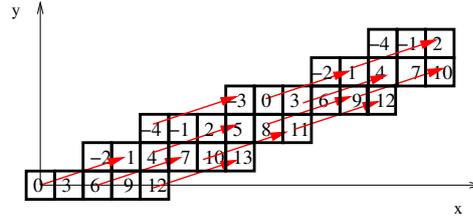


FIG. 2.5 – Représentations du vecteur de décalage positif de la droite $\mathcal{D}(3, 8, -4, 18)$, la valeur du reste en fonction de \mathcal{D} est indiquée sur chaque pixel du segment.

CROISSANCE ARITHMÉTIQUE D'UN SEGMENT FLOU

À partir de cette définition, un théorème étudiant la croissance d'un segment flou d'ordre d a été élaboré. Celui-ci permet une reconnaissance incrémentale d'un segment flou d'ordre d . Ce théorème repose sur des propriétés arithmétiques des droites discrètes. Il se situe dans le même esprit que celui proposé pour la reconnaissance des droites discrètes naïves [7].

Le problème est le suivant : soient $\mathcal{S}f$ un segment flou arithmétique d'ordre d sur l'intervalle des abscisses $[0, l-1]$ et $\mathcal{D}(a, b, \mu, \omega)$ une droite discrète strictement englobante pour $\mathcal{S}f$. Le point M , connecté à $\mathcal{S}f$, d'abscisse x_M égale à l ou $l-1$, est ajouté à $\mathcal{S}f$. La droite discrète $\mathcal{D}(a, b, \mu, \omega)$ est-elle toujours strictement englobante pour $\mathcal{S}f \cup M$ et, si ce n'est pas le cas, comment pouvons-nous déterminer une droite strictement englobante pour $\mathcal{S}f \cup M$? Une solution est proposée dans le théorème 2.2, énoncé ci-après. Son principe est le suivant, deux cas sont possibles :

- $M \in \mathcal{D}$, dans ce cas \mathcal{D} est strictement englobante pour $\mathcal{S}f \cup M$,
- $M \notin \mathcal{D}$, supposons que $r(M) \geq \mu + \omega$ (cas symétrique si $r(M) < \mu$), M est alors situé sous la droite d'appui inférieure de \mathcal{D} , le principe utilisé consiste à épaissir si nécessaire la droite \mathcal{D} de sorte que la droite d'appui inférieure soit la droite d'équation $ax - by = r(M) - 1$, et ensuite de prendre comme nouvelle pente celle obtenue à partir du vecteur \overrightarrow{PM} où P est le point entier de cette droite dont l'abscisse vérifie $x_P \in [0, b-1]$, l'épaisseur de cette nouvelle droite est calculée à partir du dernier point d'appui supérieur de \mathcal{D} présent dans $\mathcal{S}f$ (cf. Figure 2.6). Le vecteur \overrightarrow{PM} est calculé à partir des vecteurs de décalage de \mathcal{D} dont je rappelle la définition ci-dessous.

Proposition 2.1 Soit $\mathcal{D}(a, b, \mu, \omega)$ une droite discrète, pour chaque entier k , il existe exactement un point entier P_k dont les coordonnées (x_{P_k}, y_{P_k}) satisfont les 2 conditions suivantes :

$$\begin{cases} r(P_k) = k \\ 0 \leq x_{P_k} \leq b-1 \end{cases}$$

Définition 2.4 Nous appelons *vecteur de décalage positif (resp. négatif)* de $\mathcal{D}(a, b, \mu, \omega)$, noté V_{+1} (resp. V_{-1}), le vecteur entre 2 points de \mathcal{D} pour lesquels les valeurs des restes sont différentes de $+1$ (resp. -1) et tel que $0 \leq x_{V_{+1}} \leq b$ (resp. $0 \leq x_{V_{-1}} \leq b$) (cf. Figure 2.5).

Théorème 2.2 Considérons un segment flou arithmétique $\mathcal{S}f$ dont les abscisses appartiennent à l'intervalle $[0, l-1]$, pour lequel $\mathcal{D}(a, b, \mu, \omega)$ est une droite strictement englobante. Dans ce cas l'ordre de $\mathcal{S}f$ est $\frac{\omega}{b}$. Soit $M(x_M, y_M)$ un point entier connecté à $\mathcal{S}f$ dont l'abscisse égale à l ou $l-1$.

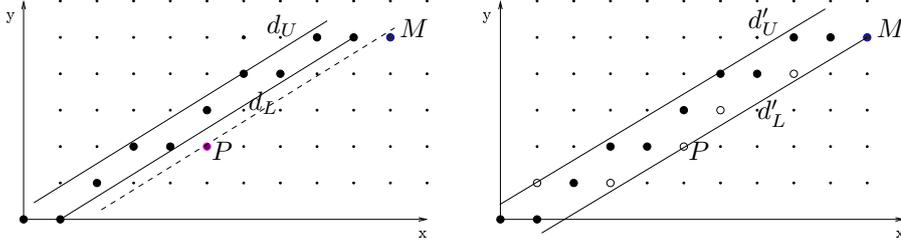


FIG. 2.6 – A gauche, un segment flou $\mathcal{S}f$ d'ordre 1, $\mathcal{D}(5, 8, -2, 8)$ est strictement englobante et nous lui ajoutons le point $M(10, 5)$, $r_{\mathcal{D}}(M) = 10$. P est le point de l'intervalle $[0, 7]$ tel que $r_{\mathcal{D}}(P) = 9$. A droite, $\mathcal{D}'(3, 5, -2, 7)$ droite strictement englobante pour $\mathcal{S}f \cup \{M\}$ dont la pente est calculée à partir du vecteur \overrightarrow{PM} , en noir les points de $\mathcal{S}f \cup \{M\}$ et en blanc les points de \mathcal{D}' qui n'appartiennent pas à $\mathcal{S}f \cup \{M\}$.

(i) Si $\mu \leq r(M) < \mu + \omega$, alors $M \in \mathcal{D}$.

$\mathcal{S}f \cup M$ est un segment flou arithmétique d'ordre $\frac{\omega}{b}$ avec \mathcal{D} comme droite strictement englobante.

(ii) Si $r(M) \leq \mu - 1$, M est extérieur à \mathcal{D} .

$\mathcal{S}f \cup M$ est un segment flou arithmétique d'ordre $\frac{\omega'}{b'}$ et la droite $\mathcal{D}'(a', b', \mu', \omega')$ est strictement englobante avec,

– b' et a' les coordonnées du vecteur $\overrightarrow{P_{r(M)+1}M}$, $P_{r(M)+1}$ étant le point dont le reste en fonction de \mathcal{D} est $r(M) + 1$ et tel que $x_{P_{r(M)+1}} \in [0, b - 1]$,

– $\mu' = a'x_M - b'y_M$

– $\omega' = a'x_{L_L} - b'y_{L_L} - \mu' + 1$, avec $L_L(x_{L_L}, y_{L_L})$ le dernier point d'appui inférieur de la droite \mathcal{D} présent dans $\mathcal{S}f$.

De plus, les vecteurs de décalage de \mathcal{D}' sont :

$$\overrightarrow{V'_{-1}} = (b' - b, a' - a) + [\frac{b'}{b'}](b', a') \text{ et } \overrightarrow{V'_{+1}} = (b, a) - [\frac{b}{b'}](b', a')$$

(iii) Si $r(M) \geq \mu + \omega$, alors M est extérieur à \mathcal{D} (cf. 2.6).

$\mathcal{S}f \cup \{M\}$ est un segment flou arithmétique d'ordre $\frac{\omega'}{b'}$ et la droite $\mathcal{D}'(a', b', \mu', \omega')$ est strictement englobante avec,

– b' et a' les coordonnées du vecteur $\overrightarrow{P_{r(M)-1}M}$, $P_{r(M)-1}$ étant le point dont le reste en fonction de \mathcal{D} est $r(M) - 1$ et tel que $x_{P_{r(M)-1}} \in [0, b - 1]$,

– $\mu' = a'x_{U_L} - b'y_{U_L}$ avec $U_L(x_{U_L}, y_{U_L})$ le dernier point d'appui supérieur de la droite \mathcal{D} présent dans $\mathcal{S}f$,

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

De plus, les vecteurs de décalage de \mathcal{D}' sont :

$$\overrightarrow{V'_{-1}} = (b, a) - [\frac{b}{b'}](b', a') \text{ et } \overrightarrow{V'_{+1}} = (b' - b, a' - a) + [\frac{b'}{b'}](b', a')$$

Remarques :

1. Dans les cas (ii) et (iii) de ce théorème, la nouvelle pente est calculée avec les vecteurs $\overrightarrow{P_{r(M)+1}M}$ et $\overrightarrow{P_{r(M)-1}M}$. Grâce aux vecteurs de décalage de \mathcal{D} , ils peuvent être facilement calculés de la manière suivante :

$$\overrightarrow{P_{r(M)-1}M} = [\frac{x_M - x_{\overrightarrow{V'_{+1}}}}{b'}](b, a) + \overrightarrow{V'_{+1}} \text{ et } \overrightarrow{P_{r(M)+1}M} = [\frac{x_M - x_{\overrightarrow{V'_{-1}}}}{b'}](b, a) + \overrightarrow{V'_{-1}}.$$

2. Quand $r(M) = \mu - 1$, le point M est dit **faiblement extérieur** à \mathcal{D} et le point $P_{r(M)+1}$

est le premier point d'appui supérieur de \mathcal{D} dans l'intervalle $[0, l - 1]$. De même, quand $r(M) = \mu + \omega$, le point M est faiblement extérieur à \mathcal{D} et le point $P_{r(M)-1}$ est le premier point d'appui inférieur de \mathcal{D} dans l'intervalle $[0, l - 1]$. Nous retrouvons alors dans ce cas le principe utilisé dans la reconnaissance des segments de droites discrètes naïves [7].

3. Ajouter un point M à un segment flou $\mathcal{S}f$ en utilisant le théorème 2.2 permet d'obtenir une droite strictement englobante. Cependant, celle-ci n'est pas toujours la plus proche des points de $\mathcal{S}f \cup \{M\}$. En fait, pour un ordre supérieur à 1, les caractéristiques de la droite strictement englobante du segment flou obtenu ne sont pas toujours telles que $\frac{\omega'}{l'}$ soit minimal. Cela est dû au choix du point P qui, dans le théorème 2.2, assure que nous obtenons une droite strictement englobante mais pas toujours la "meilleure". Cette situation peut arriver quand un grand nombre de points de la droite strictement englobante n'appartiennent pas au segment flou courant, c'est-à-dire quand la courbe analysée est très bruitée.

Ce théorème, démontré dans l'article [2] en annexe de ce mémoire (partie III), a été utilisé pour reconnaître des segments flous arithmétiques d'ordre fixé, le principe utilisé ici est différent de celui de la reconnaissance des segments de droites discrètes naïves. En effet, à chaque étape, la pente de la droite peut changer et l'épaisseur aussi. De plus, tous les points de la droite dans l'intervalle considéré n'appartiennent pas nécessairement au segment flou arithmétique. Quand le point M ajouté est faiblement extérieur à \mathcal{D} , l'étape d'épaississement n'existe pas, le point de la première période à considérer pour le calcul de la pente de la nouvelle droite englobante se situe sur la droite d'appui inférieure de \mathcal{D} .

L'algorithme 7 (voir page suivante) utilise le théorème 2.2 pour reconnaître des segments flous pour un ordre donné. Les opérations réalisées dans cet algorithme pour faire évoluer la pente sont peu coûteuses : mémorisation des derniers points d'appui parcourus, additions, soustractions entre les coordonnées en $O(1)$ ainsi que le calcul du pgcd entre les coefficients de la pente obtenue à chaque changement. Par conséquent la complexité de cet algorithme est en $O(n \log \delta)$ avec n le nombre de points de la suite à analyser et δ le plus grand coefficient (en valeur absolue) des pentes des droites strictement englobantes obtenues au cours de l'algorithme 7.

ALGORITHME DE DÉCOMPOSITION DE COURBES DISCRÈTES EN SEGMENTS FLOUS ARITHMÉTIQUES D'ORDRE FIXÉ

Nous avons ensuite élaboré un algorithme incrémental de découpage de courbes discrètes en segments flous d'ordre fixé.

Soit \mathcal{C} une courbe discrète à segmenter, d l'ordre de la segmentation, \mathcal{C} est incrémentalement parcourue et chaque point est testé.

Considérons $\mathcal{S}f$ le segment flou courant dans cette procédure, le point M de \mathcal{C} est ajouté à $\mathcal{S}f$ par une procédure utilisant l'algorithme 7 qui peut éventuellement faire évoluer les caractéristiques de la droite strictement englobante de $\mathcal{S}f \cup M$. En fonction du rapport $\frac{\omega}{\max(|a|, |b|)}$ obtenu, le segment flou courant inclut ou non le point M .

- Si $\frac{\omega}{\max(|a|, |b|)} > d$, M n'est pas ajouté au segment flou courant $\mathcal{S}f$, celui-ci se termine au point M_p précédant M dans \mathcal{C} , la droite strictement englobante de $\mathcal{S}f$ possède les mêmes caractéristiques que celles obtenues avant l'ajout du point M . Un nouveau segment débute alors avec les points M_p et M .

Algorithme 7 : Reconnaissance arithmétique des segments flous

Données : E une suite 8-connexe de points entiers, d l'ordre autorisé pour le segment flou.

Résultat : une valeur booléenne $isSegment$,

– faux si E n'est pas un segment flou d'ordre d selon le théorème 2.2

– vrai sinon, dans ce cas a, b, μ et ω sont les caractéristiques du segment flou

$a = 0$; $b = 1$; $\omega = b$; $\mu = 0$; $isSegment = true$;

M = premier point de E ; $L_L = U_L = M = (0, 0)$;

$\vec{V}_{-1} = (0, 1)$; $\vec{V}_{+1} = (0, -1)$;

tant que E n'est pas entièrement parcouru **et** $isSegment$ **faire**

M = point suivant de E ;

$r = ax_M - by_M$;

si $r = \mu$ **alors** $U_L = M$;

si $r = \mu + \omega - 1$ **alors** $L_L = M$;

si $r \leq \mu - 1$ **alors**

$a_{last} = \frac{a}{pgcd(a,b)}$; $b_{last} = \frac{b}{pgcd(a,b)}$; $\mu_{last} = \mu$; $\omega_{last} = \omega$;

$a = [\frac{x_M - x_{\vec{V}_{-1}}}{b_{last}}]a_{last} + y_{\vec{V}_{-1}}$;

$b = [\frac{x_M - x_{\vec{V}_{-1}}}{b_{last}}]b_{last} + x_{\vec{V}_{-1}}$;

$\mu = ax_M - by_M$;

$\omega = ax_{L_L} - by_{L_L} - \mu + 1$;

$U_L = M$; Mettre à jour L_L si nécessaire;

$\vec{V}_{-1} = (b - b_{last} + [\frac{b_{last}}{b}]b, a - a_{last} + [\frac{b_{last}}{b}]a)$;

$\vec{V}_{+1} = (b_{last} - [\frac{b_{last}}{b}]b, a_{last} - [\frac{b_{last}}{b}]a)$;

si $r \geq \mu + \omega$ **alors**

$a_{last} = \frac{a}{pgcd(a,b)}$; $b_{last} = \frac{b}{pgcd(a,b)}$; $\mu_{last} = \mu$; $\omega_{last} = \omega$;

$a = [\frac{x_M - x_{\vec{V}_{+1}}}{b_{last}}]a_{last} + y_{\vec{V}_{+1}}$;

$b = [\frac{x_M - x_{\vec{V}_{+1}}}{b_{last}}]b_{last} + x_{\vec{V}_{+1}}$;

$\mu = ax_{U_L} - by_{U_L}$;

$\omega = ax_M - by_M - \mu + 1$;

$L_L = M$; Mettre à jour U_L si nécessaire;

$\vec{V}_{-1} = (b_{last} - [\frac{b_{last}}{b}]b, a_{last} - [\frac{b_{last}}{b}]a)$;

$\vec{V}_{+1} = (b - b_{last} + [\frac{b_{last}}{b}]b, a - a_{last} + [\frac{b_{last}}{b}]a)$;

$isSegment = \frac{\omega}{b} \leq d$;

- Si $\frac{\omega}{\max(|a|,|b|)} \leq d$, M est ajouté à $\mathcal{S}f$, $\mathcal{S}f$ devient $\mathcal{S}f \cup \{M\}$ et les caractéristiques de sa droite strictement englobante sont les dernières ayant été calculées.

La gestion des changements d'octant pendant le parcours de la courbe peut être effectuée avec différentes stratégies. Nous avons choisi de faire tous les calculs dans le premier octant en considérant que le premier point de chaque segment a pour coordonnées $(0,0)$. Pour chaque point M ajouté, nous travaillons avec son *transformé* dans le premier octant M_c après avoir vérifié que M appartenait à l'octant du segment courant.

Quand le segment courant est entièrement parcouru, nous calculons ses caractéristiques dans l'octant d'origine en utilisant les propriétés de symétrie des droites discrètes [Rev91].

Une illustration de cet algorithme, détaillé dans [2], est donnée à la Figure 2.7.

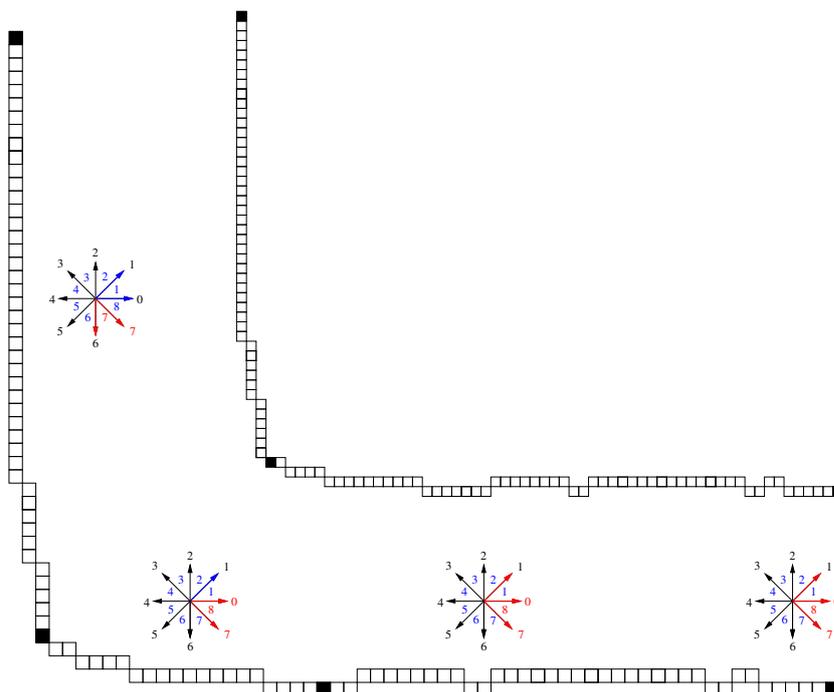


FIG. 2.7 – En bas segmentation de la courbe à l'ordre 2 avec les directions de Freeman associés pour chaque segment ; en haut segmentation à l'ordre 3.

Cependant, en raison du problème énoncé dans la remarque 3 après le théorème 2.2, nous obtenons quelquefois une sur-segmentation de la courbe discrète. Nous avons étudié ce problème avec Fabien Feschet du LLAIC et Jocelyne Rouyer-Degli [1,13]. C'est une conséquence du choix de la mesure de l'épaisseur que nous avons fait : $\frac{\omega}{\max(|a|,|b|)}$. De plus, le théorème suivant a été prouvé :

Théorème 2.3 *Pour tout ensemble fini \mathcal{S} , on considère \mathcal{W} la suite $(\omega(a_k, b_k)/b_k)$ avec k un entier positif, $a_k = k$ et $b_k = kb_0 + \lambda$, avec b_0 et λ entiers positifs et $w(\cdot)$ l'épaisseur arithmétique de la droite de paramètres (a_k, b_k) contenant \mathcal{S} . \mathcal{W} est décroissante et a pour limite $\frac{\omega(1, b_0) - 1}{b_0}$, quand k tend vers $+\infty$.*

Afin d'obtenir un algorithme de segmentation de courbe discrète permettant d'avoir les plus longs segments possibles pour une épaisseur donnée, nous avons considéré une sous-classe de segments flous. Nous avons choisi comme mesure la limite obtenue dans le théorème précédent et cela nous a conduits à l'approche géométrique décrite dans le paragraphe suivant.

2.2.2 Approche géométrique

L'approche géométrique que nous avons proposée repose sur l'enveloppe convexe de l'ensemble de points considérés.

Définition 2.5 *L'épaisseur verticale d'une droite discrète $\mathcal{D}(a, b, \mu, \omega)$ est la distance verticale (différence des ordonnées) entre les droites d'appui de \mathcal{D} .*

Proposition 2.4 *L'épaisseur verticale de $\mathcal{D}(a, b, \mu, \omega)$ est égale à $\frac{\omega-1}{\max(|a|, |b|)}$.*

Définition 2.6 *La distance verticale d'un ensemble convexe C est la longueur maximale de l'intersection de l'enveloppe convexe de C avec une droite verticale.*

Définition 2.7 *Une droite englobante $\mathcal{D}(a, b, \mu, \omega)$ d'une suite de points entiers $\mathcal{S}f$ est dite optimale si son épaisseur verticale est égale à l'épaisseur verticale de l'enveloppe convexe de $\mathcal{S}f$.*

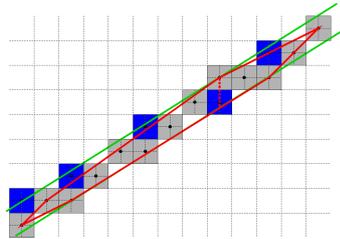


FIG. 2.8 – $\mathcal{D}(5, 8, -8, 11)$, droite englobante optimale (ep $\frac{10}{8} = 1.25$) de la suite de points grisés.

Cette définition nous a conduits à la nouvelle notion de segment flou (voir Figure 2.8) définie ci-dessous dont les points sont ordonnés mais ne sont pas forcément connectés :

Définition 2.8 *Soit $\mathcal{S}f$ une suite de points entiers, $\mathcal{S}f$ est un **segment flou d'épaisseur ν** si et seulement si sa droite optimale englobante a une épaisseur verticale inférieure ou égale à ν .*

CROISSANCE ET RECONNAISSANCE GÉOMÉTRIQUE D'UN SEGMENT FLOU

A partir de cette définition et en utilisant une approche similaire à celle du *rotating caliper* de Houle et Toussaint [HT88], nous avons obtenu un algorithme linéaire de reconnaissance de segment flou d'épaisseur ν . En fait, la reconnaissance d'un segment flou $\mathcal{S}f$ d'épaisseur ν est équivalente au calcul de la distance verticale de $\text{conv}(\mathcal{S}f)$ ¹⁰.

Pour cela nous considérons la fonction $\mathcal{V}(i)$ égale à la distance entre l_i et u_i avec l_i et u_i respectivement le plus bas et le plus haut point d'intersection de l'enveloppe convexe d'un ensemble C avec la droite verticale $x = i$. Il est clair que la fonction $\mathcal{V}(\cdot)$ est une fonction concave ce qui implique la propriété suivante :

¹⁰ $\text{conv}(\mathcal{S}f)$ est l'enveloppe convexe de $\mathcal{S}f$

Proposition 2.5 *Pour chaque ensemble convexe fini C , la fonction $\mathcal{V}(\cdot)$ a une valeur maximale à la position i avec l_i ou u_i pouvant être choisi comme un sommet de $\text{conv}(C)$.*

À partir de cette proposition, nous pouvons faire les déductions suivantes qui nous aideront dans la recherche de l'épaisseur verticale d'un convexe :

- nous pouvons ne considérer que les points du bord de l'enveloppe convexe de C ,
- l'extréma peut être localisé à partir de certaines positions des abscisses des sommets de $\text{conv}(C)$. Trois cas sont possibles : (arête, sommet), (sommet, arête) et (sommet, sommet) où le premier élément représente la partie inférieure et le second la partie supérieure du bord.

Cette proposition est identique à celle qui caractérise l'épaisseur d'un ensemble convexe dans [PS85].

Soit $\mathcal{S}_f = \{(x_i, y_i), 0 \leq i < n\}$, un segment flou du premier octant avec $\mathcal{D}(a, b, \mu, \omega)$ comme droite englobante optimale. Nous supposons que \mathcal{S}_f contient 2 points d'appui supérieurs, U_F et U_L , ainsi qu'un point d'appui inférieur, L_L . U_F , U_L et L_L sont des sommets de $\text{conv}(\mathcal{S}_f)$. De plus, la distance verticale de \mathcal{D} et de $\text{conv}(\mathcal{S}_f)$ peut être calculée au point L_L . Le calcul de l'enveloppe convexe de \mathcal{S}_f s'effectue avec l'algorithme de Melkman [Mel87]. Cet algorithme calcule incrémentalement l'enveloppe convexe de n points d'une ligne polygonale simple en temps $O(n)$. Comme nous ajoutons les points par abscisses croissantes, nous garantissons ainsi d'avoir une ligne polygonale simple.

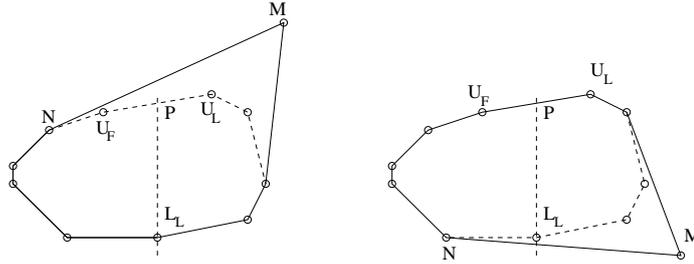


FIG. 2.9 – Ajout d'un point au-dessus ou en-dessous du convexe.

Supposons que nous ajoutons un point M à \mathcal{S}_f , $\mathcal{S}'_f = \mathcal{S}_f \cup \{M\}$. Trois cas sont possibles : M est en-dessous, au-dessus ou appartient à \mathcal{D} .

Si M appartient à \mathcal{D} alors, après l'application de l'algorithme de Melkman [Mel87], la distance verticale reste la même. Le nouvel ensemble de points, \mathcal{S}'_f , est un segment flou de même épaisseur et avec la même droite englobante optimale \mathcal{D} .

Dans les autres situations, la distance verticale doit être recalculée. Le point d'intersection P entre la droite verticale issue de L_L et $U_F U_L$, est strictement à l'intérieur de $[U_F U_L]$ (voir Figure 2.9).

Supposons que M est ajouté au-dessus de \mathcal{D} . L'enveloppe convexe est modifiée en conséquence en utilisant l'algorithme de Melkman. Nous appelons N le point situé avant M dans la partie supérieure de l'enveloppe convexe résultante, voir Figure 2.9 (gauche). N est nécessairement situé sur ou avant U_F . Par conséquent, la projection verticale de L_L est dans $[NM]$. Alors, la distance verticale de ce nouvel ensemble convexe croît strictement. N ne peut pas être localisé à une position maximale puisque $\mathcal{V}(N) \leq \mathcal{V}(L_L)$. De plus, M ne peut pas se projeter verticalement strictement à l'intérieur d'une arête de la partie inférieure de l'enveloppe convexe de $\text{conv}(\mathcal{S}'_f)$.

Par conséquent, la nouvelle position du maximum est nécessairement obtenue sur un point à droite de L_L dans la partie inférieure de $\text{conv}(\mathcal{S}'_f)$. Grâce à la propriété 2.5, la recherche des points candidats dans la suite stoppe au premier maximum trouvé, nommé C .

Supposons maintenant que M est ajouté en-dessous de \mathcal{D} . Après modification de l'enveloppe convexe, N est comme précédemment situé avant M dans la partie inférieure de l'enveloppe convexe résultante, voir Figure 2.9 (droite). N se trouve nécessairement à gauche de L_L . Par conséquent, N ne peut pas être à la position du maximum puisque $\mathcal{V}(N) \leq \mathcal{V}(L_L)$. Nous ne savons pas exactement où N se situe par rapport à $[U_F U_L]$. Cependant, il est aisé de constater qu'aucune position à gauche de U_L ne peut être la position maximale. Comme précédemment, ni M ni N ne sont à une position maximale. Ainsi la position du nouveau maximum se situe à droite de U_L , sur la partie supérieure de l'enveloppe convexe et il suffit de tester les points candidats de la suite de points et de s'arrêter sur le premier maximum obtenu, nommé C .

Dans ces deux situations, \mathcal{S}'_f est un segment flou avec pour droite englobante optimale \mathcal{D}' pour laquelle les points M , N , et C sont des points d'appui. De plus, la distance verticale de $\text{conv}(\mathcal{S}'_f)$ est égale à la distance verticale de \mathcal{D}' et peut être calculée au point C .

A partir de l'étude de la croissance d'un segment flou, nous pouvons en déduire un algorithme linéaire et incrémental de reconnaissance de segments flous d'épaisseur ν (voir Algorithme 8) où les points sont pris dans l'ordre des abscisses croissantes.

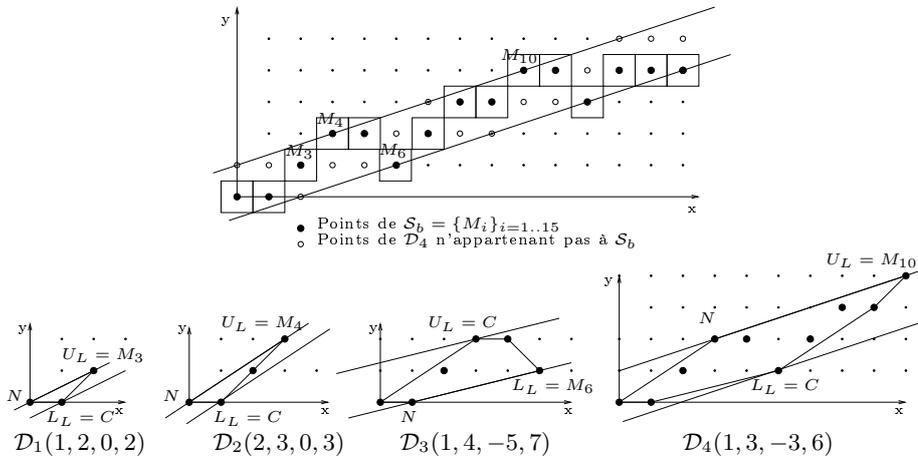


FIG. 2.10 – Un exemple de reconnaissance incrémentale pour l'épaisseur 2.

Complexité de l'algorithme 8. Considérons un ensemble S de n points. La partie de l'algorithme concernant l'insertion de chaque point de S à l'enveloppe convexe courante s'effectue avec une complexité linéaire grâce à l'algorithme de Melkman [Mel87]. La seconde partie de cet algorithme concerne la mise à jour de la position où se trouve la distance maximale. A chaque ajout d'un point, une partie de l'enveloppe convexe précédente est examinée afin de trouver la position du nouveau maximum. Cependant la fonction $\mathcal{V}(\cdot)$ étant concave, nous stoppons la recherche au premier maximum trouvé et cette partie a donc aussi une complexité linéaire.

Pour être strictement rigoureux, si nous voulons obtenir à chaque étape de la reconnaissance une droite discrète englobante optimale pour l'ensemble déjà parcouru, les coefficients de la pente

Algorithme 8 : Reconnaissance incrémentale d'un segment flou d'épaisseur ν dans le premier octant

Entrées : S une suite de points entiers, ν un réel

Résultat : $isSegment$ un booléen de valeur vraie si S est un segment flou d'épaisseur ν et faux sinon ; $a_f, b_f, \mu_f, \omega_f$ entiers, les caractéristiques de la droite englobante optimale de S si elle existe pour l'épaisseur ν

Initialisation : $isSegment = true$; $a = 0$; $b = 1$; $\omega = b$; $\mu = 0$; $M = (x_0, y_0)$;

tant que S n'est pas entièrement parcourue **et** $isSegment$ **faire**

$M =$ point suivant de S ;

 Ajouter M aux enveloppes convexes inférieure et supérieure de la suite de points de S parcourue ;

$r = ax_M - by_M$;

si $r = \mu$ **alors** $U_L = M$;

si $r = \mu + \omega - 1$ **alors** $L_L = M$;

si $r \leq \mu - 1$ **alors**

$U_L = M$;

$N =$ le point situé avant M dans l'enveloppe convexe supérieure ;

$a = y_M - y_N$; $b = x_M - x_N$; $\mu = ax_M - by_M$; (*)

 Trouver le premier point C dans la partie inférieure de l'enveloppe convexe en commençant par le point L_L , tel que : pente de $[C, Cnext] > \frac{a}{b}$;

$L_L = C$;

$\omega = ax_{L_L} - by_{L_L} - \mu + 1$;

sinon

si $r \geq \mu + \omega - 1$ **alors** *cas symétrique*

$isSegment = \frac{\omega-1}{b} \leq \nu$;

si $isSegment$ **alors** $a_f = \frac{a}{\gcd(a,b)}$; $b_f = \frac{b}{\gcd(a,b)}$; $\mu_f = \frac{\mu}{\gcd(a,b)}$; $\omega_f = \frac{\omega-1}{\gcd(a,b)} + 1$; (**)

de cette droite doivent être par définition premiers entre eux et nous devons donc ajouter une division par $\text{pgcd}(a, b)$ à la ligne (*) de l'algorithme mais cela ajoute un facteur $\log \delta$ à la complexité de l'algorithme (δ étant le plus grand coefficient en valeur absolue des pentes des droites strictement englobantes obtenues au cours de l'algorithme). Le résultat obtenu n'étant pas modifié, nous préférons utiliser l'algorithme tel qu'il est décrit et, selon les applications, l'étape de simplification des coefficients de la ligne (**) peut aussi être omise.

Un exemple du déroulement incrémental de l'algorithme 8 est montré à la Figure 2.10. La suite de points à reconnaître, \mathcal{S}_f , et les droites englobantes obtenues aux différentes étapes de la reconnaissance incrémentale y sont représentées. \mathcal{S}_f est un segment flou d'épaisseur 2 avec $\mathcal{D}_4(1, 3, -3, 6)$ pour droite optimale englobante.

ALGORITHME INCRÉMENTAL ET LINÉAIRE DE DÉCOMPOSITION DE COURBES DISCRÈTES EN SEGMENTS FLOUS ARITHMÉTIQUES D'ÉPAISSEUR FIXÉE

Un algorithme incrémental et linéaire de segmentation de courbes discrètes en segments flous d'épaisseur ν a été déduit de l'algorithme de reconnaissance précédent. Il permet de tenir compte du bruit présent dans la courbe en fixant l'épaisseur ν de la segmentation.

Cet algorithme utilise le même principe que celui présenté dans la section 2.2.1. La courbe est parcourue incrémentalement ; quand l'épaisseur du segment courant devient strictement supérieure à ν , le segment flou courant stoppe au point précédent et un nouveau segment est initialisé. Afin de gérer simplement les changements d'octants, nous effectuons le calcul de l'épaisseur et de la largeur du segment incrémentalement pendant le parcours de la courbe, la largeur étant calculée selon un principe similaire à celui décrit pour le calcul de l'épaisseur. Une illustration de l'algorithme de décomposition en segments flous d'ordre 2 est proposée à la Figure 2.11. Un algorithme similaire a été proposé par L. Buzer dans [Buz05].

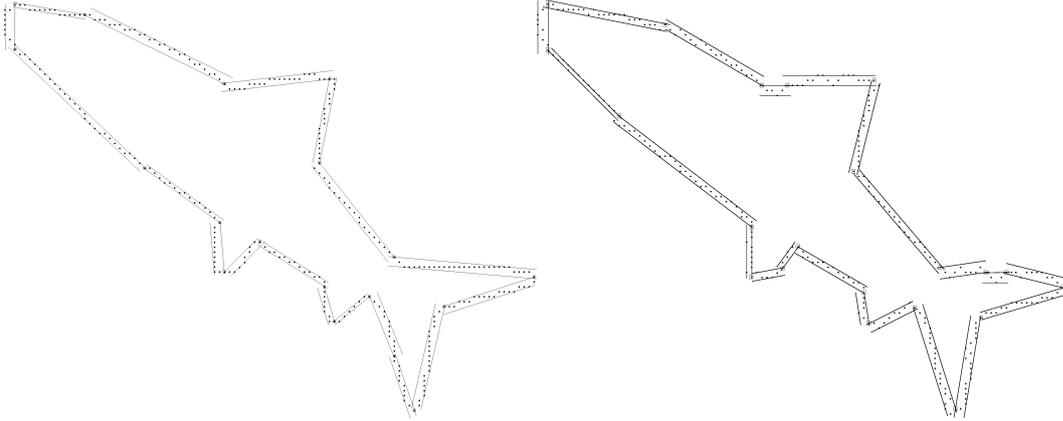


FIG. 2.11 – Contours d'images de requins (à gauche sans bruit, à droite avec bruit) décomposés en segments flous d'épaisseur 2.

Nous avons constaté que, plus la courbe est bruitée, plus les points extrémités de la segmentation sont décalés par rapport aux points attendus. Cette remarque est illustrée à la figure 2.12.

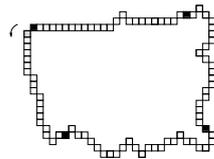


FIG. 2.12 – Segmentation d'épaisseur 4. Le point gris est le point de départ et les points noirs sont les points extrémités des segments flous d'épaisseur 4.

Nous avons proposé dans [1] un post-traitement permettant de décaler dans certains cas les points extrémités obtenus sur des positions jugées plus vraisemblables. Pour cela, nous utilisons les caractéristiques des droites optimales englobantes obtenues pour chaque segment flou du découpage de la courbe.

L'idée est la suivante : pour deux segments flous consécutifs \mathcal{S}_i et \mathcal{S}_{i+1} , de droites englobantes optimales $\mathcal{D}_i(a_i, b_i, \mu_i, \omega_i)$ et $\mathcal{D}_{i+1}(a_{i+1}, b_{i+1}, \mu_{i+1}, \omega_{i+1})$, ayant comme extrémité commune le point E_{i+1} , nous considérons les droites rationnelles d_i et d_{i+1} situées entre (au milieu) des droites d'appui de \mathcal{D}_i et \mathcal{D}_{i+1} et ayant respectivement les mêmes pentes. Le point d'intersection I_{i+1} , entre les deux droites d_i et d_{i+1} est alors considéré. Si les deux droites d_i et d_{i+1} n'appartiennent pas au même octant, alors le point de la courbe le plus proche de I_{i+1} devient le nouveau point

extrémité et remplace le point E_{i+1} . Par contre, quand la variation entre les pentes des droites englobantes des segments consécutifs est faible, il faut faire attention car leur intersection peut ne pas être pertinente pour une modification de l'extrémité existante. Pour palier ce problème, nous n'appliquons pas ce pré-traitement quand la variation entre les pentes est inférieure à un seuil devant être fixé par l'utilisateur.

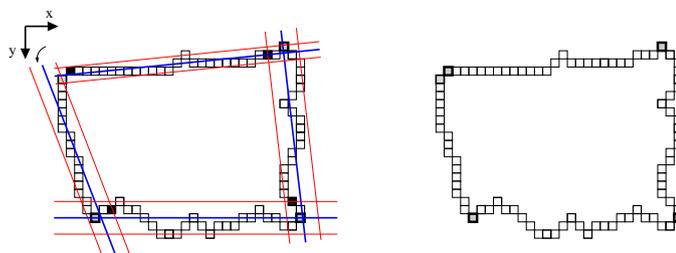


FIG. 2.13 – Exemple de post-traitement : à gauche, en noir les extrémités des segments flous obtenus après une décomposition d'épaisseur 4, à droite, les nouveaux points extrémités.

Des illustrations de l'utilisation de ce post-traitement sont proposées aux figures 2.13 et 2.14. A la figure 2.13, les extrémités obtenues après une segmentation floue d'épaisseur 4 sont : $E_1(0, 0)$, $E_2(6, 16)$, $E_3(28, 15)$, $E_4(25, -3)$ et $E_5(1, -1)$. Les droite englobantes optimales sont $\mathcal{D}_1(-8, -3, 0, 27)$, $\mathcal{D}_2(0, -1, 15, 5)$, $\mathcal{D}_3(17, 2, 431, 67)$, $\mathcal{D}_4(-1, 10, -3, 19)$. Les droites suivantes sont considérées : $d_1 : -8x + 3y = 13$, $d_2 : y = 17$, $d_3 : 17x - 2y = 464$ et $d_4 : -x - 10y = 6$. Les points d'intersection sont $I_2(4.75, 17)$, $I_3(29.29, 17)$ et $I_4(26.9, -3.29)$. Par suite, les nouvelles extrémités sont les points $E_2(4, 17)$, $E_3(29, 17)$ et $E_4(27, -4)$, points de la courbe discrète étudiée.

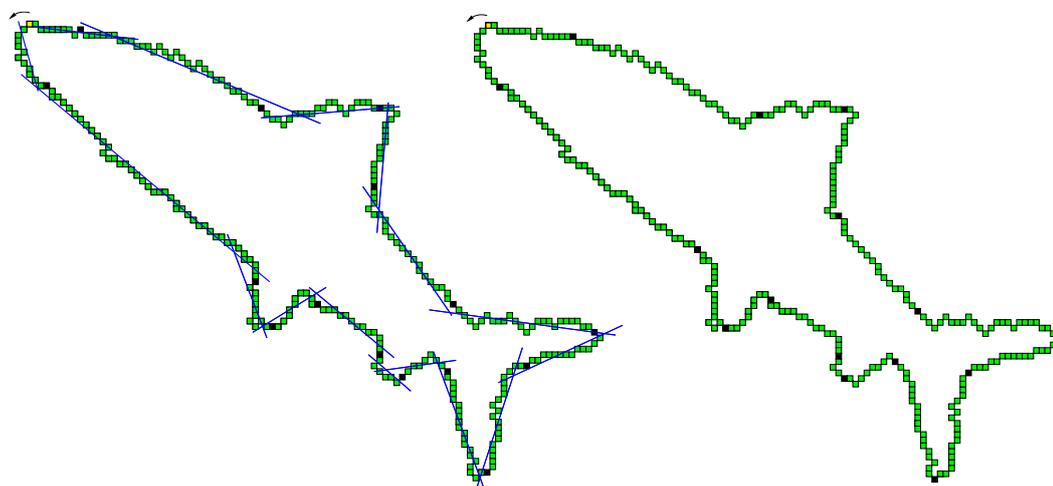


FIG. 2.14 – Exemple de post-traitement : à gauche, en noir les extrémités des points obtenus après une décomposition en segments flous d'épaisseur 3.9. A droite, les nouveaux points extrémité.

D'autres améliorations peuvent être proposées dans le cadre de la segmentation de courbes discrètes. Plus spécifiquement, nous avons utilisé les segments flous dans le milieu de l'analyse de documents et une collaboration avec l'équipe QGAR (Navigation dans les documents graphiques par l'analyse et la reconnaissance) du LORIA a permis d'appliquer l'algorithme de segmentation dans ce cadre. Cette application est décrite dans le paragraphe suivant.

2.2.3 Approximation polygonale multi-niveaux de courbes discrètes

En traitement d'images, l'objectif d'une méthode d'approximation polygonale est généralement de fournir un découpage d'une chaîne de points en segments de façon à minimiser un critère d'erreur global ou à ne pas dépasser localement une erreur. L'intérêt est de donner une description qui soit plus compacte et plus adaptée pour des traitements ultérieurs d'interprétation ou de reconnaissance de formes. Les chaînes de points sont principalement issues d'un détecteur de contours dans des images à niveaux de gris ou d'un squelette dans des documents graphiques.

De très nombreuses méthodes concernant l'approximation polygonale ont été proposées ces vingt dernières années (un état de l'art assez complet figure dans [Hil04]). Dans certains cas [Ram72, SG80, WD84], on examine successivement les points pour déterminer le plus long segment qui vérifie un seuil de tolérance prédéfini. Perez et Vidal [PV94] ont donné un algorithme qui minimise une erreur globale en se basant sur la programmation dynamique. L'approche souffre néanmoins de sa complexité élevée car toutes les configurations sont examinées et, pour chacune d'elles, une erreur globale d'approximation est calculée. Cet algorithme a suscité beaucoup d'intérêt et des travaux [KF03, Sal01] ont été proposés pour réduire la complexité de l'approche. Par ailleurs, des méthodes de division/fusion [RR95, RW89] se focalisent sur la division récursive d'une chaîne de points en segments si le découpage excède un seuil d'erreur prédéfini et, le cas échéant, fusionnent les segments. Dans d'autres approches [AD91, WW93], des points d'accroche correspondant à des points de forte courbure sont injectés dans le processus de segmentation.

Malgré le nombre important de méthodes d'approximation, il subsiste des problèmes majeurs de robustesse, de stabilité aux transformations géométriques et de complexité. De plus, les algorithmes reposent sur des seuils de tolérance d'erreur qui sont définis manuellement sans aucune connaissance sur la valeur du seuil le plus pertinent. Ceux-ci peuvent aussi être différents d'une chaîne de points à l'autre.

Dans cette perspective, en collaboration avec Antoine Tabbone et Laurent Wendling de l'équipe QGAR du LORIA, nous avons proposé [4,14,26] une analyse multi-niveaux qui permet de définir une méthode exempte de seuil et de décrire une chaîne de points par sa décomposition en segments flous arithmétiques les plus significatifs ¹¹.

La technique proposée a émergé en observant le comportement de l'algorithme de segmentation proposé à la section 2.2.1 avec l'approche arithmétique :

Pour un ordre faible, les extrémités obtenues sont correctement localisées. Cependant, pour des courbes bruitées, beaucoup trop de petits segments non significatifs sont obtenus. Quand l'ordre augmente, comme nous l'avons déjà signalé dans le paragraphe précédent, nous pouvons observer que les points extrémités des segments sont décalés par rapport aux points attendus. Cependant, le nombre de segments décroît et seuls les segments significatifs sont conservés. Par exemple, à la figure 2.15, nous pouvons constater que le nombre de segments obtenus à l'ordre 1 est de 15 alors qu'il décroît à 3 pour l'ordre 7.

Ces remarques nous ont conduits à l'élaboration de l'algorithme présenté dans le paragraphe suivant. Dans les publications sur ce sujet, seul l'algorithme présenté dans la section 2.2.1 (approche arithmétique) est utilisé car le second (approche géométrique) n'était pas encore paru. Cependant les remarques énoncées précédemment s'appliquent aussi à l'algorithme de segmentation en segments flous d'épaisseur fixée. Par conséquent dans la suite de ce paragraphe le mot *niveau* sera utilisé pour parler indifféremment de *l'ordre ou de l'épaisseur de la segmentation*, les deux algorithmes pouvant être utilisés.

¹¹Les segments significatifs sont ceux qui correspondent aux segments principaux de la courbe qui permettent d'en conserver la forme principale.

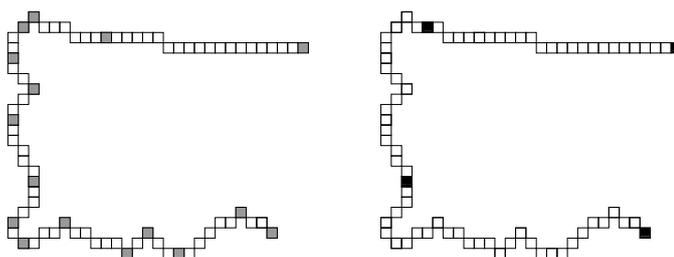


FIG. 2.15 – Exemples de segmentation à l'ordre 1 (à gauche) et à l'ordre 7 (à droite). La segmentation est effectuée depuis le point en bas à droite.

PRINCIPE DE L'ANALYSE MULTI-NIVEAUX

Nous avons proposé d'analyser un ensemble de points dans un espace multi-niveaux. Cet espace ressemble à la description en espace échelle proposée par A. Witkin [Wit83] pour la détection de contours dans un espace échelle.

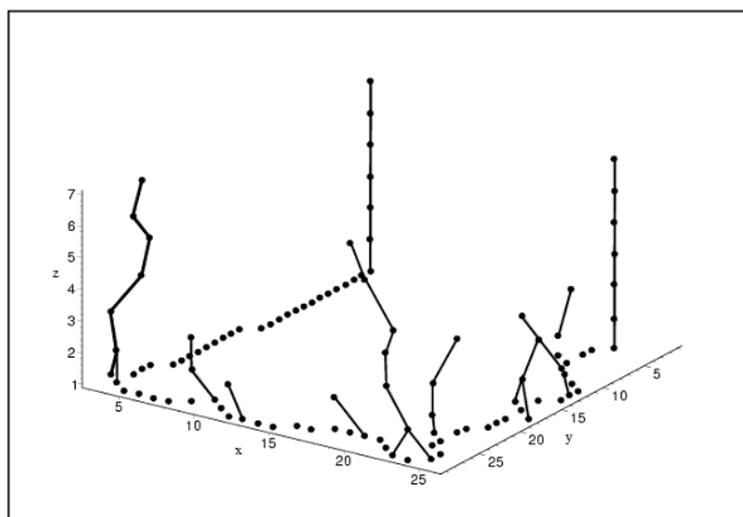


FIG. 2.16 – Exemple d'un espace multi-niveaux sur la courbe présentée à la figure 2.15.

La figure 2.16 montre un espace multi-niveaux déterminé à partir des segments flous obtenus à différents niveaux (ordre ou épaisseur) pendant une segmentation de la courbe de la figure 2.15.

Le principe est le suivant ; pour chaque niveau n , représenté sur l'axe Oz , nous indiquons les coordonnées des points extrémités des segments flous obtenus pendant la segmentation. Nous pouvons voir sur l'exemple de la figure 2.16 qu'au dernier niveau, 4 extrémités sont conservées. Elles correspondent à la description de la courbe en 3 segments (voir Figure 2.17). Dans cet espace, la position des points évolue, excepté pour les points extrémités. Lorsque le nombre de segments d'un niveau à l'autre ne change plus, nous décidons d'arrêter le processus. A partir du dernier niveau, les extrémités sont projetées à un niveau inférieur et leurs nouvelles positions sont définies à partir des extrémités les plus proches des segments issus de ce niveau inférieur. La taille du voisinage d'observation est proportionnelle au niveau en cours d'analyse.

L'algorithme général permettant d'extraire les points multi-niveaux est présenté ci-après.

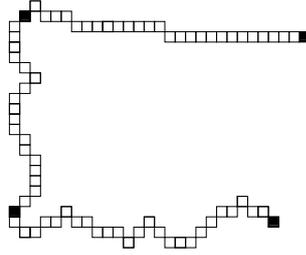


FIG. 2.17 – Points extrémités obtenus après une segmentation multi-niveaux.

Approx est une fonction, issue de l'algorithme 7 ou 8, permettant de segmenter une suite de points \mathcal{C} au niveau n (ordre ou épaisseur) et ayant pour résultat le nombre de points extrémités trouvés, noté *nbpoint*, ainsi que la suite des points extrémités obtenus à ce niveau $L[n]$.

La recherche du voisinage dans une branche de l'espace multi-niveaux d'un niveau α à un niveau $\alpha - 1$ est effectuée en calculant la distance suivante, notée *Dist*, entre les points trouvés telle que $Dist(P_\alpha, P_{\alpha-1}) = \max(|x_{P_\alpha} - x_{P_{\alpha-1}}|, |y_{P_\alpha} - y_{P_{\alpha-1}}|)$.

Algorithme 9 : Algorithme de recherche de points multi-niveaux

Entrées : \mathcal{C} une suite de points entiers

Résultat : La liste L_{MO} des points multi-niveaux

Initialisation : $n = 0$; *nbpoint* = 0 ;

répéter

$n = n + 1$;

nbprec = *nbpoint* ;

nbpoint = *Approx*(\mathcal{C} , n , $L[n]$) ;

jusqu'à *nbpoint* == *nbprec*

$L_{MO} = L[n]$;

pour $d_c = n - 1$; $d_c \geq 1$; d_c - **faire**

$\forall u \in L_{MO}$

si $\exists p \in L[d_c] / Dist(p, u) < Int((d_c + 1)/2) + 1$ **alors**

 % déplacement dans l'espace multi-niveaux (p devient u)

$L_{MO} = L_{MO} - \{u\}$;

$L_{MO} = L_{MO} \cup \{p\}$;

sinon

$L_{MO} = L_{MO} - \{u\}$; % le chemin multi-niveaux n'est pas valide

RÉSULTATS EXPÉRIMENTAUX

Sur la figure 2.18, nous avons dessiné la polygonalisation obtenue en appliquant la méthode de Wall et Danielsson [WD84]. Nous avons ajusté au mieux le seuil requis de sorte que le nombre de segments soit similaire à celui de notre approche. Dans notre méthode, le nombre de segments est automatiquement obtenu alors que pour utiliser celle de Wall and Danielsson, il a fallu faire plusieurs tests pour obtenir la valeur correcte du seuil afin d'obtenir le même nombre de segments.

De plus, un critère d'erreur [PV94, Sal01] a été implémenté pour tester la précision de l'approche multi-niveaux. L'erreur $e_{i,j}$ est définie à partir de la courbe dans l'intervalle $[c_i; c_j]$ et de sa projection orthogonale sur la droite $c_i c_j$. Soit $C = \{c_1, \dots, c_N\} = \{(x_1, y_1), \dots, (x_N, y_N)\}$

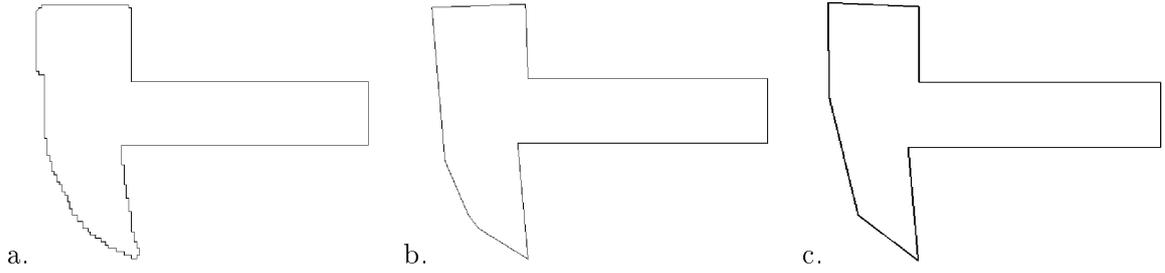


FIG. 2.18 – Résultats expérimentaux.

- a) Courbe représentant un marteau. b) Résultats obtenus automatiquement avec notre approche. c) Résultats obtenus avec la méthode de Wall and Danielsson [WD84] avec un seuil de 30.

une courbe définie par N points ordonnés. Soit $Q = \{q_1, \dots, q_{M+1}\}$ le polygone obtenu après segmentation de C en utilisant la segmentation multi-niveaux. Chaque point de Q appartient à C avec $q_1 = c_1$ et $q_{M+1} = c_N$. Considérons le segment (q_m, q_{m+1}) de Q et $\{c_i, \dots, c_j\}$ la série de points associés dans C . L'erreur entre eux est donnée par :

$$e^2(q_m = c_i, q_{m+1} = c_j) = \sum_{k=i+1}^{j-1} \frac{(y_k - a_{ij}x_k - b_{ij})^2}{1 + a_{ij}^2}$$

a_{ij} et b_{ij} sont les coefficients de la droite (c_i, c_j) . Considérons tous les segments de Q , l'erreur globale E est calculée par la formule :

$$E = \sum_{m=1}^M e^2(q_m, q_{m+1})$$

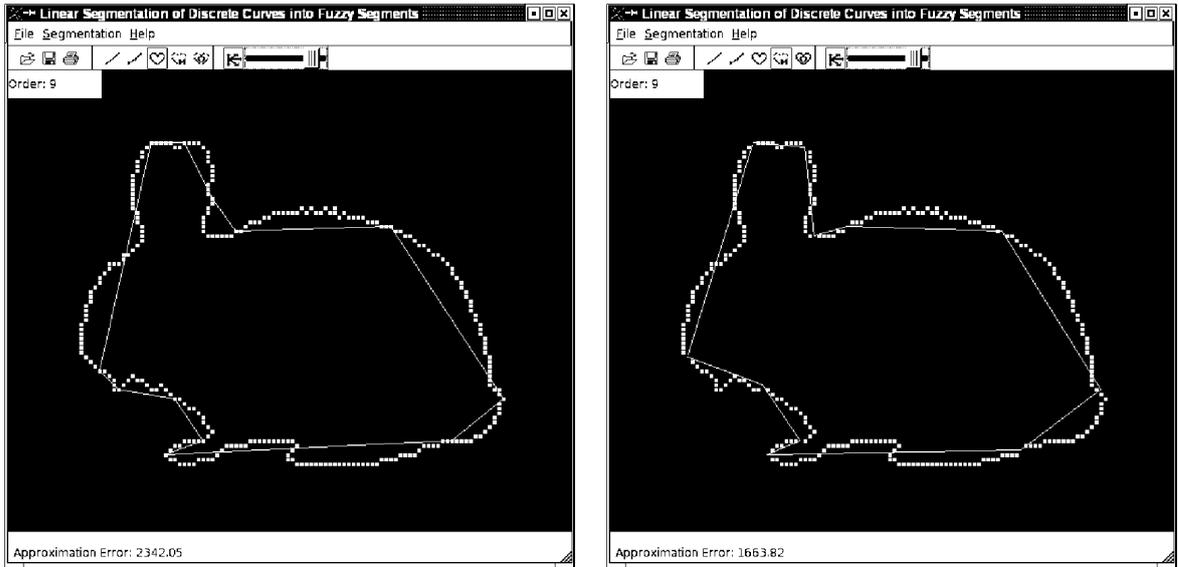


FIG. 2.19 – Erreurs (voir en bas de chaque fenêtre) obtenues en utilisant la segmentation floue (approche arithmétique, à gauche), $erreurs_{SF} = 2342.05$, et la segmentation multi-niveaux (à droite), $erreurs_{SMN} = 1663.82$. Le ratio des erreurs est égal à $\frac{erreurs_{SF}}{erreurs_{SMN}} = 0.71$.

Dans la majorité des cas, l'erreur calculée est inférieure en utilisant la segmentation multi-niveaux plutôt qu'en utilisant l'algorithme de segmentation au même ordre (voir Figure 2.19). L'utilisation de la segmentation multi-niveaux permet d'avoir une meilleure représentation de la forme globale de l'objet. D'autres résultats et comparaisons se trouvent dans l'article [4] figurant en annexe (partie III) de ce document.

Plus généralement, nous avons constaté que, plus la forme possède des points angulaires aigus, moins le ratio de l'erreur est grand donc plus la méthode multi-niveaux est efficace. Quand les formes sont lissées, les deux approches fournissent des résultats similaires.

Pour conclure sur ce sujet, contrairement aux autres approches où l'erreur d'approximation est définie manuellement, l'analyse multi-niveaux d'une courbe discrète est exempte de seuils et produit le meilleur compromis entre le nombre de segments et leur localisation : elle propose un découpage automatique de la chaîne de points en ses segments principaux. Cependant, des améliorations peuvent encore être apportées en étudiant les points suivants :

- construction des branches dans l'espace multi-niveaux en particulier gestion du voisinage où quelquefois plusieurs choix sont possibles (points à égale distance),
- gestion du point de départ et du point terminal de la courbe qui appartiennent à chaque niveau et qui peuvent éventuellement masquer des segments dominants.

2.2.4 Extension en dimension 3

La notion de segment flou arithmétique s'étend à la dimension 3 en utilisant les plans de projection, pour cela nous devons rappeler la définition d'une droite discrète 3D utilisée, nous supposons par la suite que $a \geq b \geq c \geq 0$:

Définition 2.9 Une *droite discrète 3D* [43], notée $\mathcal{D}_{3D}(a, b, c, \mu, \mu', e, e')$, de vecteur directeur (a, b, c) , avec $(a, b, c) \in \mathbb{Z}^3$, et $a \geq b \geq c$ est définie comme l'ensemble des points (x, y, z) de \mathbb{Z}^3 vérifiant :

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

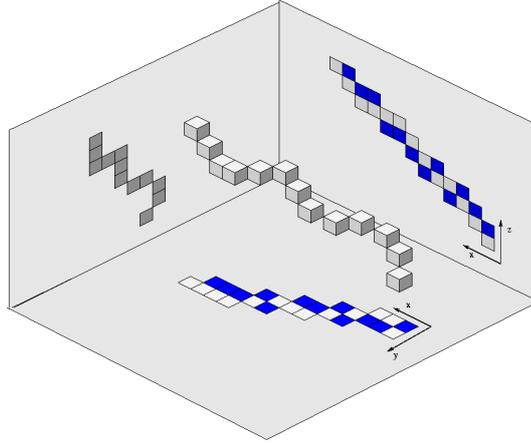
avec $\mu, \mu', e, e' \in \mathbb{Z}$. e et e' sont les épaisseurs arithmétiques de \mathcal{D} .

Selon la définition, il est clair qu'une droite discrète 3D se projette bijectivement dans deux plans de projection en deux droites discrètes arithmétiques 2D. Grâce à cette propriété, nous définissons la notion de droite discrète 3D englobante pour une suite de points de \mathbb{Z}^3 :

Définition 2.10 Soit $\mathcal{S}f_{3D}$ une suite de points de \mathbb{Z}^3 , une droite discrète $\mathcal{D}_{3D}(a, b, c, \mu, \mu', e, e')$ est dite *englobante* pour $\mathcal{S}f_{3D}$ si $\mathcal{D}(a, b, \mu', e')$ est englobante pour la suite des projections des points de $\mathcal{S}f_{3D}$ dans le plan (O, x, y) et si $\mathcal{D}(a, c, \mu, e)$ est englobante pour la suite des projections des points de $\mathcal{S}f_{3D}$ dans le plan (O, x, z) (voir Figure 2.20).

Les approches arithmétiques et géométriques et les notions de segments flous présentées dans les paragraphes précédents en dimension 2 s'étendent naturellement à la dimension 3 en considérant les projections de l'ensemble de points étudié dans les plans de projection :

Définition 2.11 Soit $\mathcal{S}f_{3D}$ une suite de points de \mathbb{Z}^3 , $\mathcal{S}f_{3D}$ est un *segment flou 3D d'épaisseur* ν (resp. *segment flou 3D d'ordre* d) s'il possède une droite discrète englobante, nommée $\mathcal{D}_{3D}(a, b, c, \mu, \mu', e, e')$, telle que

FIG. 2.20 – $\mathcal{D}_{3D}(45, 27, 20, -45, -81, 90, 90)$ droite englobante des points grisés

- $\mathcal{D}(a, b, \mu', e')$ est englobante et optimale (resp. strictement englobante) pour la suite des projections des points de $\mathcal{S}f_{3D}$ dans le plan (O, x, y) et $\frac{e'-1}{\max(|a|, |b|)} \leq \nu$ (resp. $\frac{e'}{\max(|a|, |b|)} \leq d$),
- $\mathcal{D}(a, c, \mu, e)$ est englobante et optimale (resp. strictement englobante) pour la suite des projections des points de $\mathcal{S}f_{3D}$ dans le plan (O, x, z) et $\frac{e-1}{\max(|a|, |c|)} \leq \nu$ (resp. $\frac{e}{\max(|a|, |c|)} \leq d$).

Des **algorithmes linéaires de reconnaissance de segments flous 3D** ont été déduits de cette définition pour des suites de points 18 ou 26-connexes pour l'approche arithmétique et pour des suites de points pouvant être déconnectées pour l'approche géométrique. Les algorithmes utilisés reposent sur le même principe que la reconnaissance des segments de droites discrètes 3D naïves (cf. [17,43]); il suffit de remplacer l'appel à l'algorithme de reconnaissance de segments de droites naïves 2D [7] par celui de reconnaissance de segments flous 2D (cf. Algorithme 10).

Algorithme 10 : Algorithme de reconnaissance d'un segment flou 3D d'épaisseur ν

Entrées : \mathcal{S} une suite de points de \mathbb{Z}^3 , un réel ν : l'épaisseur maximale autorisée.

Résultat : Vrai si \mathcal{S} est un segment flou de droite discrète 3D d'épaisseur ν , faux sinon.

si les voxels de \mathcal{S} ne se projettent pas bijectivement sur au moins deux plans de coordonnées pour former deux suites de points de \mathbb{Z}^2 , \mathcal{C}_1 et \mathcal{C}_2 alors

└ Retourner Faux; // \mathcal{S} n'est pas un segment flou de droite discrète 3D d'épaisseur ν

Appliquer l'algorithme de reconnaissance des segments flous 2D d'épaisseur ν sur \mathcal{C}_1 et \mathcal{C}_2 ;

si \mathcal{C}_1 et \mathcal{C}_2 sont des segments flous 2D d'épaisseur ν alors

└ Retourner Vrai; // \mathcal{S} est un segment flou de droite discrète 3D d'épaisseur ν

sinon

└ Retourner Faux; // \mathcal{S} n'est pas un segment flou de droite discrète 3D d'épaisseur ν

De plus, à partir de ces algorithmes de reconnaissance, il est aisé de déduire des algorithmes de découpage de courbes discrètes, éventuellement bruitées ou déconnectées, en segments flous 3D d'épaisseur ou d'ordre fixé qui peuvent être utilisés pour l'estimation de paramètres géométriques (voir section 2.3.6, page 95). Afin de préserver une complexité linéaire, la segmentation est réalisée incrémentalement : trois courbes discrètes 2D sont construites et segmentées, elles

correspondent aux projections du morceau de la courbe \mathcal{S} déjà parcourue dans les trois plans de coordonnées (cf. Algorithme 11). Le premier point d'un nouveau segment est le dernier du précédent.

Algorithme 11 : Algorithme de segmentation d'une courbe discrète en segments flous 3D d'épaisseur ν

Entrées : \mathcal{C} une suite de points de \mathbb{Z}^3 , un réel ν : l'épaisseur maximale autorisée.

Résultat : L suite des droites englobantes obtenues ainsi que les coordonnées des voxels de début de chaque nouveau segment.

M = premier voxel de \mathcal{C} ;

tant que \mathcal{C} n'a pas été entièrement parcourue **faire**

Initialiser les caractéristiques des nouveaux segments flous 2D d'épaisseur ν dans les plans Oxy , Oxz , Oyz ;

$Deb = M$;

SEGMENT = Vrai ;

tant que SEGMENT **faire**

M = point suivant de \mathcal{C} ;

Ajouter les projetés de M aux segments flous 2D d'épaisseur ν courants ;

SEGMENT = M possède au moins 2 projections dans lesquelles ses projections peuvent être ajoutées aux segments flous 2D courants ;

si !SEGMENT **alors**

Calculer les caractéristiques $(a, b, c, \mu, \mu', e, e')$ de la droite englobante du segment parcouru en fonction des caractéristiques des deux segments flous 2D d'épaisseur ν reconnus ;

$L = L \cup \{Deb, (a, b, c, \mu, \mu', e, e')\}$;

J'ai encadré en 2004 un étudiant de DEA, Franck Rapaport [Rap04], sur ce sujet et sur une application en Bioinformatique (cf. chapitre 4 section 4.2).

2.3 Courbure discrète et autres paramètres géométriques

2.3.1 Tangente discrète

Dans le cadre du travail sur les courbes bruitées, j'ai proposé en 2004 [15,34,36] une nouvelle notion de tangente discrète, appelée *tangente discrète d'ordre d* reposant sur la définition des segments flous avec l'approche arithmétique et adaptée aux courbes discrètes bruitées, ce qui n'était pas le cas des définitions antérieures [WS93, Via96b]. Pour cela, nous avons étendu l'algorithme de reconnaissance de segments flous avec l'approche arithmétique en étudiant l'ajout à l'extrémité droite et gauche d'un segment flou.

CROISSANCE SYMÉTRIQUE D'UN SEGMENT FLOU

Le problème est le suivant : soit $\mathcal{S}f$ un segment flou d'ordre d , dont l'intervalle des abscisses est $[x_L, x_R]$, et $\mathcal{D}(a, b, \mu, \omega)$ une droite strictement englobante de $\mathcal{S}f$. Le point $M(x_M, y_M)$, connecté à $\mathcal{S}f$, d'abscisse x_M égale à $x_L - 1$, x_L , x_R ou $x_R + 1$, est ajouté à $\mathcal{S}f$. Les différentes

configurations sur l'évolution de ce segment flou sont résumées dans le tableau 2.1 ci-après en fonction des points M , U_F et U_L respectivement le premier et le dernier point d'appui supérieur de \mathcal{D} présents dans $\mathcal{S}f$, L_F et L_L respectivement le premier et le dernier point d'appui inférieur de \mathcal{D} présents dans $\mathcal{S}f$.

$\mathcal{D}'(a', b', \mu', \omega')$ strictement englobante de $\mathcal{S}f \cup M$	M ajouté à l'extrémité droite de $\mathcal{S}f$ $x_R = x_M$	M ajouté à l'extrémité gauche de $\mathcal{S}f$ $x_L = x_M$
Agrandir un palier de \mathcal{D}	Cas $r(M) \geq \mu + \omega$ $a' = \left[\frac{x_R - x_L - x_{V_{+1}}}{b} \right] * a + y_{V_{+1}}$ $b' = \left[\frac{x_R - x_L - x_{V_{+1}}}{b} \right] * b + x_{V_{+1}}$ $\mu' = a' * x_{U_L} - b' * y_{U_L}$ $\omega' = a' * x_M - b' * y_M - \mu' + 1$ $\overrightarrow{V'_{+1}} = (b' - b + \left[\frac{b}{b'} \right] * b', a' - a + \left[\frac{b}{b'} \right] * a')$ $\overrightarrow{V'_{-1}} = (b - \left[\frac{b}{b'} \right] * b', a - \left[\frac{b}{b'} \right] * a')$	Cas $r(M) < \mu$ $\mu' = a' * x_M - b' * y_M$ $\omega' = a' * x_{L_F} - b' * y_{L_F} - \mu' + 1$ $\overrightarrow{V'_{+1}} = (b' - b + \left[\frac{b}{b'} \right] * b', a' - a + \left[\frac{b}{b'} \right] * a')$ $\overrightarrow{V'_{-1}} = (b - \left[\frac{b}{b'} \right] * b', a - \left[\frac{b}{b'} \right] * a')$
Raccourcir un palier de \mathcal{D}	Cas $r(M) < \mu$ $a' = \left[\frac{x_R - x_L - x_{V_{-1}}}{b} \right] * a + y_{V_{-1}}$ $b' = \left[\frac{x_R - x_L - x_{V_{-1}}}{b} \right] * b + x_{V_{-1}}$ $\mu' = a' * x_M - b' * y_M$ $\omega' = a' * x_{L_L} - b' * y_{L_L} - \mu' + 1$ $\overrightarrow{V'_{+1}} = (b - \left[\frac{b}{b'} \right] * b', a - \left[\frac{b}{b'} \right] * a')$ $\overrightarrow{V'_{-1}} = (b' - b + \left[\frac{b}{b'} \right] * b', a' - a + \left[\frac{b}{b'} \right] * a')$	Cas $r(M) \geq \mu + \omega$ $\mu' = a' * x_{U_F} - b' * y_{U_F}$ $\omega' = a' * x_M - b' * y_M - \mu' + 1$ $\overrightarrow{V'_{+1}} = (b - \left[\frac{b}{b'} \right] * b', a - \left[\frac{b}{b'} \right] * a')$ $\overrightarrow{V'_{-1}} = (b' - b + \left[\frac{b}{b'} \right] * b', a' - a + \left[\frac{b}{b'} \right] * a')$

TAB. 2.1 – Résumé des calculs effectués dans les quatre situations possibles concernant l'ajout à l'extrémité gauche ou droite d'un point à un segment flou

RECONNAISSANCE D'UN SEGMENT FLOU CENTRÉ

Grâce aux propriétés du paragraphe précédent, la reconnaissance d'un segment flou d'ordre d peut se faire autour d'un point T d'une courbe discrète en ajoutant alternativement un point à chacune des extrémités du segment flou, c'est ce que nous appelons *la reconnaissance d'ordre d d'un segment flou centré*.

La reconnaissance s'arrête quand tous les points de la courbe d'un côté de T ont été parcourus ou encore quand les caractéristiques de la droite englobante obtenue sont incompatibles avec l'ordre fixé pour la reconnaissance. Dans ce cas, la droite englobante retenue est celle calculée pour l'avant-dernier couple de points ajouté. T est l'origine de cette droite.

Un algorithme de reconnaissance d'un segment flou centré du premier octant est proposé ci-dessous, après les initialisations correspondant à la droite discrète horizontale, le point situé à gauche de T est ajouté.

Algorithme 12 : Algorithme de reconnaissance d'un segment flou centré du premier octant

Entrées : E une suite de points entiers 8-connexes, d l'ordre maximal autorisé, $T \in E$

Résultat : Caractéristiques de la droite strictement englobante du segment flou d'ordre d centré en T

Initialisation : $a = 0$; $b = 1$; $\omega = b$; $\mu = 0$; $isFuzzySegmentOrderd = true$;

$M = (0, 0)$; $\vec{V}_{-1} = (0, 1)$; $\vec{V}_{+1} = (0, -1)$; $left = true$; $x_R = x_L = 0$;

tant que E n'est pas entièrement parcouru d'un côté de T et $isFuzzySegmentOrderd$

faire

si $left$ **alors**

$M =$ point suivant de E à l'extrémité gauche ;

$x_L = x_M$;

sinon

$M =$ point suivant de E à l'extrémité droite ;

$x_R = x_M$;

$r = ax_M - by_M$;

si $r < \mu$ ou $r \geq \mu + \omega$ **alors**

 Calculer les points d'appui utiles ; (1)

 Calculer les nouvelles caractéristiques a, b, μ, ω de la droite strictement englobante incluant M ; (2)

 Mettre à jour les vecteurs de décalage ; (3)

$isFuzzySegmentOrderd = \frac{\omega}{b} \leq d$;

si $!left$ et $isFuzzySegmentOrderd$ **alors** $a_{last} = a$; $b_{last} = b$; $\mu_{last} = \mu$; $\omega_{last} = \omega$;

si $!isFuzzySegmentOrderd$ **alors** $a = a_{last}$; $b = b_{last}$; $\mu = \mu_{last}$; $\omega = \omega_{last}$;

$left = !left$;

Remarques :

1. Selon la valeur de $r(M)$ (variable r dans l'algorithme) et la position du point ajouté, le point d'appui nécessaire pour le calcul de μ ou ω (U_F, U_L, L_F ou L_L) est obtenu (1) grâce aux vecteurs de décalage de la droite strictement englobante du segment flou courant.

Par exemple, dans le cas où $r(M) \geq \mu + \omega$ et M un point ajouté à l'extrémité gauche du segment flou, un point d'appui supérieur de \mathcal{D} , U_i , localisé à droite de M , peut être obtenu par ce calcul : $U_i = M + (r(M) - \mu) * \vec{V}_{-1}$. Ensuite, pour obtenir les coordonnées de U_F , premier point d'appui supérieur de \mathcal{D} présent dans l'intervalle des abscisses de $\mathcal{S}f$, il suffit de translater U_i de $-[\frac{x_{U_i} - x_M}{b}] * (b, a)$.

2. Le calcul des caractéristiques de la droite discrète strictement englobante (2) et la mise à jour de ses vecteurs de décalage (3), sont réalisés en se basant sur les résultats présentés dans le

tableau précédent.

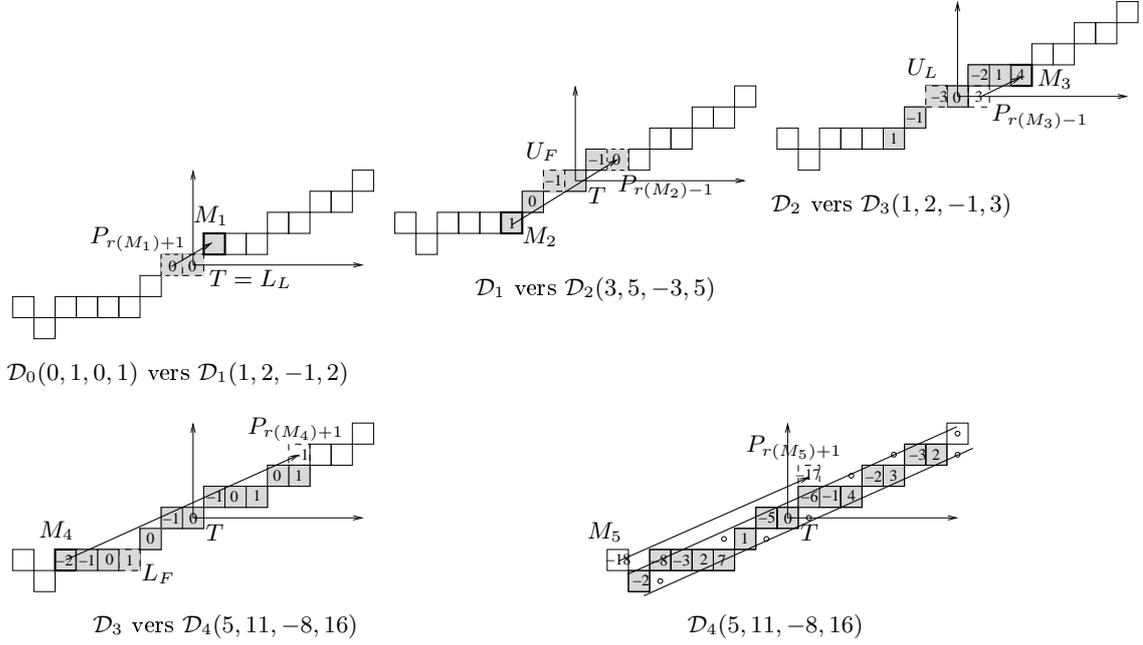


FIG. 2.21 – Exemple de déroulement de l’algorithme de reconnaissance de segments flous centrés d’ordre 2.

Un exemple de déroulement de l’algorithme de reconnaissance de segments flous centrés d’ordre 2 est montré à la figure 2.21. A chaque étape, le rapport $\frac{w_i}{b_i}$ de la droite englobante courante est comparé à 2. Pour les 4 premières droites englobantes calculées, on a $\frac{w_i}{b_i} \leq 2$, par contre l’ajout du point $M_5(-8, -2)$ avec $r_{\mathcal{D}_4}(M_5) = -18$ implique le calcul de la droite $\mathcal{D}_5(4, 9, -14, 21)$, or $\frac{w_5}{b_5} = \frac{21}{9} > 2$. Par conséquent le processus de reconnaissance s’arrête et le segment centré en T débute au point $(-7, -3)$ et se termine au point $(7, 3)$ avec $\mathcal{D}_4(5, 11, -8, 16)$ comme droite strictement englobante.

TANGENTE DISCRÈTE D’ORDRE d

Définition 2.12 La *tangente d’ordre d au point T* d’une courbe discrète est le segment flou d’ordre d , S_f , obtenu en T par l’algorithme présenté dans le paragraphe précédent. Soit $\mathcal{D}(a, b, \mu, \omega)$ la droite strictement englobante obtenue pour S_f . Alors, le *vecteur normal d’ordre d au point T* de la courbe discrète est le vecteur $(-a, b)$.

Pour l’ordre $d = 1$, cette définition correspond à celle proposée par A. Vialard [Via96b] en relation avec l’algorithme de reconnaissance de segments de droites naïves [7].

Dans le même esprit, nous pouvons aussi utiliser l’approche géométrique pour définir la notion de *tangente d’épaisseur ν* . Cette notion sera présentée succinctement dans le paragraphe suivant.

2.3.2 Courbure discrète

BREF HISTORIQUE DES ESTIMATEURS DE COURBURE

De très nombreux auteurs se sont intéressés au calcul de la courbure sur une courbe discrète en employant diverses techniques issues de la géométrie différentielle et des définitions équivalentes de la courbure d'une courbe continue. Sans pouvoir tous les citer de manière exhaustive, nous pouvons classer les approches utilisées dans trois groupes selon la définition de la courbure employée reposant sur le calcul :

- de la tangente et de la variation d'angle entre les tangentes (par exemple [SCM91, WV97]),
- du cercle osculateur (par exemple [RJ73, CMT01, Coe02]),
- de la dérivée seconde de la courbe discrète (par exemple [MM92, WB91, RC92]).

La majorité des méthodes discrètes proposées utilise une fenêtre de taille fixée *glissant* sur la courbe (par exemple [RJ73, TC89, RR91]). La taille de cette fenêtre est difficile à fixer et entraîne quelquefois des erreurs d'approximation en ne reflétant pas la forme de la courbe étudiée. Un état de l'art des différentes méthodes est donné dans deux chapitres du livre de R. Klette et A. Rosenfeld [KR04], paru en 2004 ainsi que dans la thèse de M. Marji [Mar03], soutenue en 2003 ; plus de 60 articles sur le calcul de la courbure y sont commentés.

Plus spécifiquement, certaines des techniques proposées utilisent des estimateurs de segments discrets, par exemple, dans le cadre de la géométrie discrète, en 1996, A. Vialard [Via96b] a combiné la *méthode de rééchantillonnage* présentée par M. Worring et A. Smeulders dans [WS93] et la notion de *chemin euclidien dirigé par la tangente* qu'elle avait développée pour obtenir un estimateur de courbure. L'algorithme permettant de calculer les chemins euclidiens repose sur le calcul de la tangente discrète en chaque point d'une courbe. Ce calcul est déduit de l'algorithme de reconnaissance de segment de droites discrètes naïves proposé dans [7]. En 1999, F. Feschet et L. Tougne [FT99] optimisent l'algorithme permettant de calculer les tangentes d'une courbe discrète (complexité $O(n)$ avec n le nombre de points de la courbe) et améliorent ainsi l'algorithme de calcul de la courbure discrète en chaque point d'une courbe discrète. L'estimation de la courbure utilisée dans ces algorithmes est obtenue par un filtrage gaussien appliqué aux différentes orientations de la tangente autour du point dont on veut estimer la courbure. Plus précisément la courbure $\kappa(P)$ au point P d'une courbe discrète est définie par

$$\kappa(P) = \frac{(\theta * \Gamma_\sigma)(P)}{1.1107}$$

avec $\theta(P)$ l'angle entre la tangente en P et l'axe horizontal et $\Gamma_\sigma(P) = \left(\frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{i^2}{2\sigma^2}\right) \right)_{i=-3\sigma \dots 3\sigma}$.

Le coefficient 1.1107 représente la distance moyenne entre deux points successifs de la courbe discrète et la taille de la fenêtre de 6σ limite l'influence de la fonction gaussienne.

En 2001, D. Coeurjolly, S. Miguet et L. Tougne [CMT01] proposent un estimateur basé sur le calcul du cercle osculateur en chaque point d'une courbe discrète. Celui-ci associe le calcul de la longueur de la tangente discrète en chaque point avec l'estimation du cercle osculateur discret. Si l est la demi-longueur de la tangente discrète au point P , on considère $r_{inf} = [(l - \frac{1}{2})^2 - \frac{1}{4}]$ et $r_{sup} = [(l + \frac{1}{2})^2 - \frac{1}{4}]$ et la courbure au point P est estimée par

$$\kappa(P) = \frac{2}{r_{inf} + r_{sup}}$$

Cette technique donne des résultats intéressants obtenus rapidement sur des courbes non bruitées et ne nécessite pas de paramètre. Cependant, elle n'est pas adaptée du point de vue de la conver-

gence asymptotique. D. Coeurjolly, dans sa thèse [Coe02], propose une autre méthode mieux adaptée à la convergence asymptotique, cet estimateur repose sur le calcul du rayon du cercle circonscrit au triangle (P_G, P, P_D) avec P_G (resp. P_D) le point extrémité de la demi-tangente discrète à gauche (resp. à droite) de P . Pour cela il utilise l'algorithme de reconnaissance de segments de droites discrètes [7] et ne nécessite pas de fixer une taille de fenêtre. Cette technique peut être vue comme une généralisation de la notion de courbure normalisée d'ordre m proposée par A. Rosenfeld [RJ73, CM91]. Cet estimateur fournit de bons résultats sur des courbes 2D et 3D [Coe02, CS03] non bruitées.

Des comparaisons et l'étude de la convergence de certains de ces estimateurs ont été effectuées dans [LVV05, VLF05, HK07].

Cependant, dans le cas de courbes bruitées, les algorithmes précédents ne donnent plus de bons résultats, cela est dû en partie à l'utilisation de l'algorithme de reconnaissance de segments de droites discrètes [7] qui, comme nous l'avons vu dans la section 2.1.2, provoque une sursegmentation dans le cas de courbes bruitées. C'est pourquoi nous avons utilisé la notion de segment flou pour étendre la méthode proposée dans [Coe02, CS03] et l'adapter aux courbes discrètes bruitées.

COURBURE DISCRÈTE ISSUE DES SEGMENTS FLOUS

En 2004, dans [15], j'ai défini la notion de *courbure d'ordre d* en chaque point d'une courbe discrète déduite de l'approche arithmétique des segments flous (voir section 2.2.1) et de l'estimateur proposé par D. Coeurjolly [Coe02, CS03] pour les courbes 2D et 3D sans bruit. Ensuite dans le cadre du stage de Master de Thanh Phuong Nguyen, en 2005, nous avons défini la notion de *courbure d'épaisseur ν* qui utilise l'approche géométrique des segments flous. Je décris ci-après les éléments en permettant le calcul.

Définition 2.13 *La demi-tangente discrète d'épaisseur ν à gauche (resp. à droite) d'un point T d'une courbe discrète est le plus long segment flou d'épaisseur ν englobant les points situés à gauche (resp. à droite) de T .*

L'estimation de la *courbure d'épaisseur ν au point T* est réalisée en calculant les demi-tangentes d'épaisseur ν à droite et à gauche de T , nous utilisons pour cela l'algorithme de reconnaissance de segments flous d'épaisseur ν (voir Algorithme 8, section 2.2.2).

Ces demi-tangentes commencent au point T et finissent respectivement aux points à coordonnées entières, P_R et P_L (voir Fig 2.22). Elles correspondent aux segments flous d'épaisseur ν auxquels sont associés, respectivement, les droites strictement englobantes $\mathcal{D}_R(a_R, b_R, \mu_R, \omega_R)$ et $\mathcal{D}_L(a_L, b_L, \mu_L, \omega_L)$.

Pour plus de précision, nous considérons les segments de droite réels $d_L : a_L * x - b_L * y = \mu_L + [\frac{\omega_L}{2}]$ et $d_R : a_R * x - b_R * y = \mu_R + [\frac{\omega_R}{2}]$ dont les abscisses appartiennent respectivement à $[x_{P_L}, x_T]$, d'une part, et à $[x_T, x_{P_R}]$ d'autre part.

Soit p_l le point sur d_L , d'ordonnée réelle et d'abscisse x_{P_L} , et p_r le point sur d_r d'ordonnée réelle et d'abscisse x_{P_R} . Alors, nous pouvons écrire que $p_l = (x_{P_L}, \frac{a_L * x_{P_L} - \mu_L - [\frac{\omega_L}{2}]}{b_L})$ et $p_r = (x_{P_R}, \frac{a_R * x_{P_R} - \mu_R - [\frac{\omega_R}{2}]}{b_R})$.

L'estimation de la *courbure d'épaisseur ν au point T* est déterminée grâce au rayon du cercle passant à travers les points p_l , T et p_r . Le *rayon de courbure d'épaisseur ν au point T* , $R_\nu(T)$, est défini comme le rayon du cercle circonscrit au triangle $[p_l, T, p_r]$, nous utilisons la formule définie dans [HS98] :

Soit $s_1 = \|\overrightarrow{Tp_r}\|$, $s_2 = \|\overrightarrow{Tp_l}\|$ et $s_3 = \|\overrightarrow{p_l p_r}\|$, alors

$$R_\nu(T) = \frac{s_1 s_2 s_3}{\sqrt{(s_1 + s_2 + s_3)(s_1 - s_2 + s_3)(s_1 + s_2 - s_3)(s_2 + s_3 - s_1)}}$$

La *courbure d'épaisseur ν au point T* est alors définie par la relation suivante :

$$C_\nu(T) = \frac{S}{R_\nu(T)}$$

avec $S = \text{sign}(\det(\overrightarrow{Tp_r}, \overrightarrow{Tp_l}))$.

Comme indiqué dans [Coe02, CS03], les cas dégénérés, qui correspondent par exemple aux demi-tangentes colinéaires, peuvent être examinés indépendamment et, ainsi, une courbure nulle est affectée au point considéré.

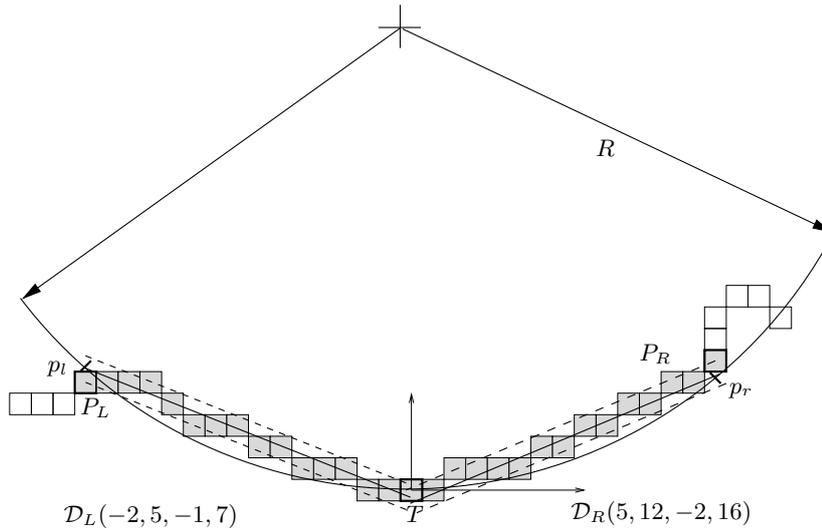


FIG. 2.22 – Un exemple de calcul de courbure d'épaisseur 1.3 au point T

Un exemple de calcul de rayon de courbure d'épaisseur 1.3 est illustré à la Fig 2.22. Les coordonnées du point T sont $(0,0)$. D'un côté, la droite englobante optimale correspond à la demi-tangente à droite d'épaisseur 1.3 notée $\mathcal{D}_R(5, 12, -2, 16)$ et de l'autre à la demi-tangente d'épaisseur 1.3 à gauche notée $\mathcal{D}_L(-2, 5, -1, 7)$. Pour trouver les points p_l et p_r , les droites réelles à considérer sont : $d_L : -2x - 5y = 2$ et $d_R : 5x - 12y = 6$. Nous obtenons $p_l = (-15, 5.6)$ et $p_r = (14, 5.33)$. En utilisant la formule ci-dessus, nous trouvons que $R_{1.3}(T)$, le rayon du cercle circonscrit au triangle $[p_l, T, p_r]$, est égal à 20.3, ainsi $C_{1.3}(T) \simeq 0.049$.

En considérant une courbe discrète \mathcal{C} de n points, le calcul de la courbure en chaque point de \mathcal{C} nécessite de calculer en chaque point les demi-tangentes à droite et à gauche pour une épaisseur

fixée ν . Ce calcul peut être fait en utilisant l'algorithme linéaire de reconnaissance de segments flous d'épaisseur ν (Algorithme 8, section 2.2.2). La complexité de l'estimation de la courbure pour \mathcal{C} est alors en $O(n^2)$. Une technique d'optimisation de ce calcul [8] a ensuite été proposée en considérant une extension de la notion de segment maximal [FT99, LVV05] que je décris dans le paragraphe suivant. Cette technique peut être utilisée sur des *courbes discrètes de structures générales* c'est-à-dire pouvant être par exemple *déconnectées ou non simples*.

2.3.3 Segments flous maximaux et optimisation du calcul de la courbure

Dans [FT99], la notion de segment maximal d'une courbe discrète a été introduite, l'ensemble des segments maximaux ou fondamentaux d'une courbe discrète permet d'avoir une compréhension globale de la structure de la courbe comme cela a été souligné dans la section 1.2 du chapitre 1. Nous avons proposé une extension de cette notion en utilisant les segments flous, F. Feschet dans [Fes06] évoque l'intérêt de cette extension. Les notations sont les mêmes que dans [LVV05].

Soit C une courbe discrète, les points de C sont indexés et ordonnés de 0 à $n - 1$. $C_{i,j}$ est l'ensemble des points successifs de C de l'indice i à j .

Définition 2.14 Le prédicat " $C_{i,j}$ est un segment flou d'épaisseur ν " est noté $BS(i, j, \nu)$. Le premier indice j , $i \leq j$, tel que $BS(i, j, \nu)$ et $\neg BS(i, j + 1, \nu)$ est appelé le front de i et est noté $F(i)$. De manière symétrique, le premier indice i tel que $BS(i, j, \nu)$ et $\neg BS(i - 1, j, \nu)$ est appelé queue de j et est noté $Q(j)$.

Définition 2.15 $C_{i,j}$ est un *segment flou maximal d'épaisseur ν* et est noté $MBS(i, j, \nu)$ si $BS(i, j, \nu)$ et $\neg BS(i, j + 1, \nu)$ et $\neg BS(i - 1, j, \nu)$.

Il est évident qu'une caractérisation équivalente d'un segment flou maximal d'épaisseur ν , $MBS(i, j, \nu)$, est de montrer que $F(i) = j$ and $Q(j) = i$.

Nous avons alors la propriété suivante (démontrée dans [8]) :

Proposition 2.6 Soit C une courbe discrète, $MBS_\nu(C)$ la suite des segments flous maximaux d'épaisseur ν de C . $MBS_\nu(C) = \{MBS(B_1, E_1, \nu), MBS(B_2, E_2, \nu), \dots, MBS(B_m, E_m, \nu)\}$ et $B_1 < B_2 < \dots < B_m$. Alors on a aussi : $E_1 < E_2 < \dots < E_m$.

L'algorithme 13 utilise un principe similaire à celui proposé dans [FT99, LVV05] pour obtenir la suite des segments flous maximaux d'épaisseur ν . Il faut pour cela pouvoir ajouter ou enlever un point à un segment flou d'épaisseur ν . L'ajout a été traité à la section 2.2.2 mais le retrait nécessite un traitement plus coûteux. L'épaisseur et la largeur de l'enveloppe convexe des points parcourus doivent être calculées incrémentalement après un ajout ou un retrait d'un point. Nous utilisons pour cela les résultats des travaux donnés dans [OL81] et [Buz05].

Dans l'article [OL81], M.H. Overmars, J. van Leeuwen considèrent l'enveloppe convexe comme l'union de deux parties; l'enveloppe convexe inférieure, $Conv_{Inf}$, et l'enveloppe convexe supérieure, $Conv_{Sup}$. $Conv_{Inf}$ et $Conv_{Sup}$ sont mises à jour après chaque opération d'ajout ou de suppression en $O(\log^2 n)$ (courbe discrète de n points), ils utilisent pour cela une structure d'arbre binaire. De plus, afin de déterminer la largeur et la hauteur de l'enveloppe convexe, nous utilisons la technique de recherche proposée par L. Buzer [Buz05] reposant sur la propriété de concavité de

la fonction $Hauteur(x) = Conv_{Sup}(x) - Conv_{Inf}(x)$. Cette opération est en $O(\log^2 n)$. Par conséquent, l'Algorithme 13 permettant d'obtenir la suite des segments flous maximaux d'épaisseur ν d'une courbe discrète de n points avec une complexité en $O(n \log^2 n)$.

Algorithme 13 : Algorithme de recherche des segments flous maximaux d'épaisseur ν d'une courbe C

Entrées : C - courbe discrète de n points, ν - épaisseur de la segmentation

Résultat : MBS_ν - la suite des segments flous maximaux d'épaisseur ν de C

$k=0$; $S_b = \{C_0\}$; $MBS_\nu = \emptyset$; $a = 0$; $b = 1$; $\omega = b$, $\mu = 0$;

tant que $\frac{\omega-1}{\max(|a|,|b|)} \leq \nu$ **faire**

└ $k++$; $S_b = S_b \cup C_k$; Déterminer $D(a, b, \mu, \omega)$ de S_b ;

$bSegment=0$; $eSegment=k-1$;

$MBS_\nu = MBS_\nu \cup C_{bSegment, eSegment}$;

tant que $k < n - 1$ **faire**

└ **tant que** $\frac{\omega-1}{\max(|a|,|b|)} > \nu$ **faire**

└└ $S_b = S_b \setminus C_{bSegment}$; $bSegment++$;

└└ Déterminer $D(a, b, \mu, \omega)$ de S_b ;

└ **tant que** $\frac{\omega-1}{\max(|a|,|b|)} \leq \nu$ **faire**

└└ $k++$; $S_b = S_b \cup C_k$;

└└ Déterminer $D(a, b, \mu, \omega)$ of S_b ;

└ $eSegment=k-1$; $MBS_\nu = MBS_\nu \cup C_{bSegment, eSegment}$;

Pour optimiser le calcul de la courbure d'épaisseur ν en tout point d'une courbe discrète C , nous utilisons la suite des segments flous maximaux d'épaisseur ν de cette courbe. Pour chaque point C_k de C , nous considérons les segments flous maximaux auxquels il appartient ; plus spécifiquement, nous recherchons l'extrémité gauche du segment flou maximal le plus long à gauche de C_k , et l'extrémité droite du segment flou maximal le plus long à droite de C_k . Nous devons donc trouver les indices $Q(k)$ (queue de k) et $F(k)$ (front de k) pour tout indice k de 0 à $n-1$. Le rayon de courbure d'épaisseur ν est alors le rayon du cercle passant par les trois points suivants : $C_{Q(k)}$, C_k , $C_{F(k)}$.

Considérons alors la décomposition de C en segments flous maximaux :

$$MBS_\nu(C) = \{MBS(B_1, E_1, \nu), MBS(B_2, E_2, \nu), \dots, MBS(B_m, E_m, \nu)\}$$

avec $B_1 < B_2 < \dots < B_m$ et $E_1 < E_2 < \dots < E_m$. Nous avons alors la propriété suivante qui permet d'obtenir pour chaque indice k d'un point de C , $F(k)$ et $Q(k)$ en fonction des indices des segments flous maximaux :

Proposition 2.7

- $\forall k$ tel que $E_{i-1} < k \leq E_i$, alors $Q(k) = B_i$
- $\forall k$ tel que $B_i \leq k < B_{i+1}$, alors $F(k) = E_i$

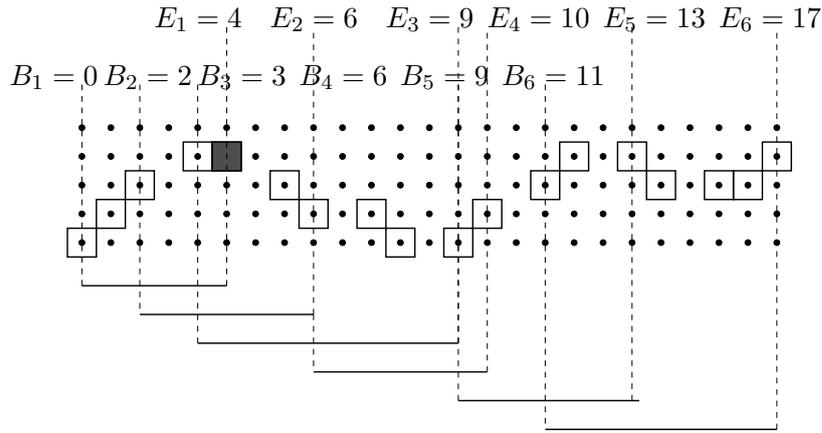


FIG. 2.23 – Décomposition d'une courbe en segments flous maximaux d'épaisseur 1

Par exemple, sur la figure 2.23, le point d'indice 4 (en noir) est tel que $Q(4) = B_1 = 0$ et $F(4) = E_3 = 9$. Cette méthode est utilisée dans l'Algorithme 14.

Algorithme 14 : Estimateur de la courbure d'épaisseur ν en chaque point de C

Entrées : C courbe discrète de n points, ν épaisseur

Résultat : $\{C_\nu(C_k)\}_{k=0..n-1}$ - Courbure d'épaisseur ν en chaque point de C

Construire $MBS_\nu = \{MBS(B_i, E_i, \nu)\}$;

$m = |MBS_\nu|$; $E_{-1} = -1$; $B_m = n$;

pour $i = 0$ à $m - 1$ **faire**

pour $k = E_{i-1} + 1$ à E_i **faire** $Q(k) = B_i$;

pour $k = B_i$ à $B_{i+1} - 1$ **faire** $F(k) = E_i$;

pour $i = 0^{(*)}$ à $n - 1^{(*)}$ **faire**

$R_\nu(C_i) = \text{Rayon du cercle circonscrit à } [C_{Q(i)}, C_i, C_{F(i)}]$;

$C_\nu(C_i) = \frac{s}{R_\nu(C_i)}$;

Complexité de l'Algorithme 14

Les deux étapes : recherche de $F(k)$ et $Q(k)$ pour chaque point C_k de la courbe et estimation de la courbure en chaque point sont exécutées en temps linéaire. Cependant, la recherche des segments flous maximaux est réalisée en $O(n \log^2 n)$. La complexité de l'Algorithme 14 est alors en $O(n \log^2 n)$.

Remarque :

(*) Les bornes mentionnées dans l'Algorithme 14 sont correctes pour une courbe fermée dont le code est dupliqué. Dans le cas d'une courbe ouverte, les conditions deviennent : *Pour* $i = l$ à $n - 1 - l$ avec l fixé à une valeur constante. Et comme il n'est pas possible de calculer correctement la courbure aux points extrémités de la courbe ouverte, ce calcul commence (resp. stoppe) au l^{eme} (resp. $(n - 1 - l)^{\text{eme}}$) point de la courbe.

Deux illustrations du calcul de la courbure d'épaisseur 2 se trouvent à la figure 2.24. La courbe d'un cercle de rayon 20 (a) ainsi que le contour d'une forme de lapin (b) s'y trouvent ; les pics présents sur le graphe de courbure de celui-ci correspondent aux pixels noirs sur cette courbe et nous pouvons constater que ces points permettent de détecter des changements dans la structure de la courbe comme la convexité/concavité ou encore une rupture dans la courbe pouvant correspondre à un changement de forme.

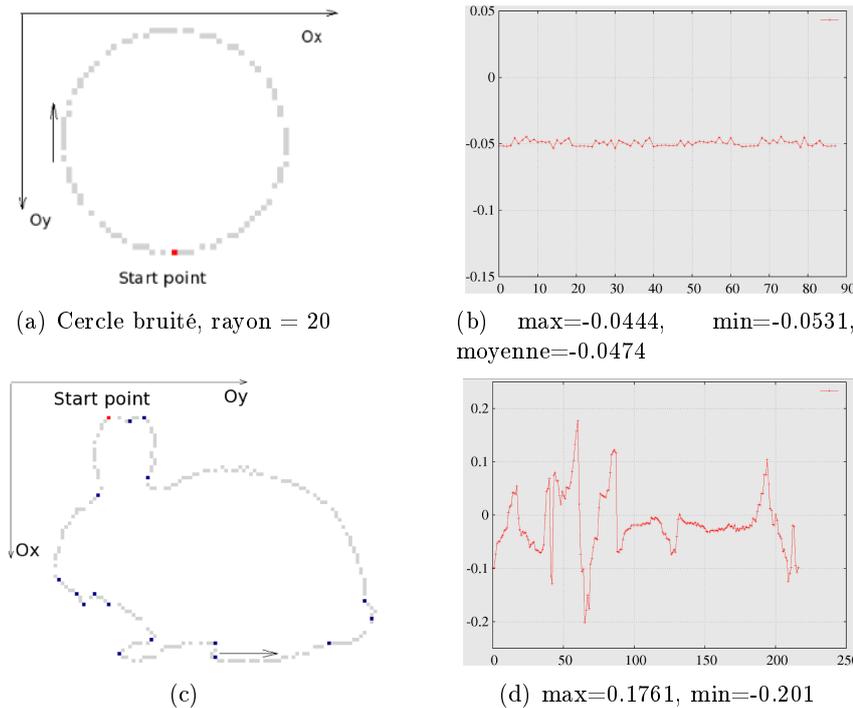


FIG. 2.24 – Exemples de calcul de courbure d'épaisseur $\nu = 2$

En traitement d'image, cet algorithme de calcul de la courbure d'épaisseur ν est utilisé dans le cadre d'une application à la reconnaissance de cercles, d'arcs de cercles et de segments dans des documents techniques que nous décrivons ci-après.

2.3.4 Application à la détection de cercles et arcs de cercles dans des images techniques

Les verrous scientifiques permettant une recherche précise de symboles dans des documents techniques sont loin d'être levés : évaluation de la qualité des documents et des descripteurs utilisés lors de la reconnaissance, gestion des occultations, passage à l'échelle, etc. Une collaboration a été engagée avec Laurent Wendling et Jean-Pierre Salmon de l'équipe QGAR du LORIA au sujet de l'étude des descripteurs les plus adéquats pour la recherche de symboles complexes dans des documents techniques. Les arcs et les cercles sont des primitives graphiques de base contenues dans des documents techniques (plans d'architectes, schémas de câblage, mécanique ...) et représentent souvent une partie d'un symbole [Kas90].

Plus précisément, l'objectif de ce travail est de segmenter des courbes discrètes issues de

documents techniques en segments de droites ou en arcs de cercles. De nombreux travaux ont été réalisés sur ce sujet, une partie des publications citées dans la section 2.2.3 sur l'analyse multi-niveaux abordent ce problème de détection de segments et d'arcs de cercles. Ces travaux peuvent donc aussi être regroupés selon deux approches :

- la **recherche de points dominants** (dits encore points d'accroche, points critiques ou points de rupture), basée sur le calcul d'une **approximation de la courbure** en chaque point de la courbe à analyser. Les fortes variations de la courbure indiquent alors les points dominants de la courbe et la nature de la primitive (segment ou arc) entre deux de ces points doit être analysée (par exemple les méthodes de W. Wan et J. A. Ventura [WV97] et P. Saint-Marc et al [SCM91]),
- l'**approximation polygonale** qui permet d'obtenir une suite de segments en utilisant des seuils reposant sur des calculs de distance ou d'aire entre la courbe discrète (par exemple la méthode de K. Wall et P. Danielson [WD84]). Dans certaines méthodes comme celle de Rosin et West [RW89, RW95], les segments successifs obtenus sont dans un second temps analysés selon différents critères pour les regrouper et les détecter comme segments ou comme arcs de cercles.

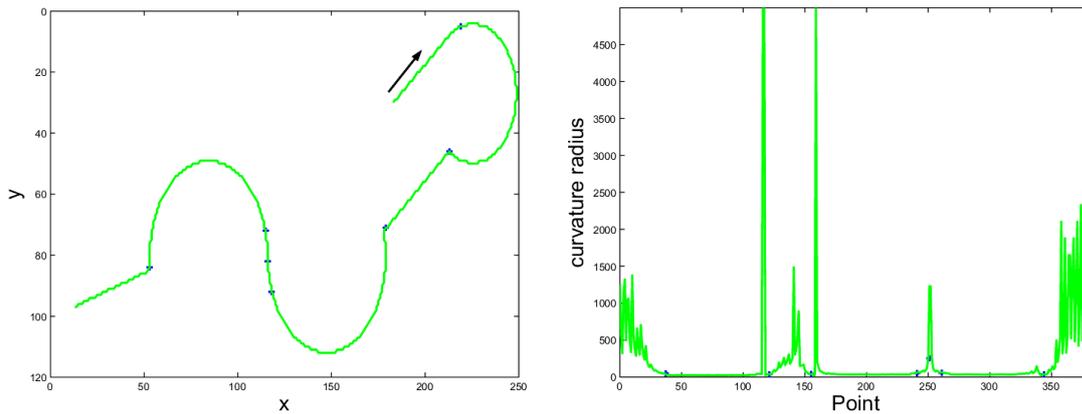
Citons aussi la méthode RANVEC, développée par X. Hilaire et K. Tombre [HT06] qui résout le problème de la segmentation en arcs et segments en utilisant une technique qui ne teste qu'un sous-ensemble de points définis aléatoirement dans une chaîne du squelette et évalue en parallèle sa corrélation avec un modèle de segment ou d'arc de cercle. Cette méthode s'est révélée très efficace lors de la compétition GREC de 2005 [Hil06], cependant la technique utilisée ne testant qu'un sous-ensemble aléatoire de points, cela peut engendrer dans certains cas des parties non prises en compte.

Le travail présenté dans cette section se situe dans le cadre de la première approche, nous avons étudié les descripteurs obtenus en utilisant les graphes de courbures des symboles présents dans des documents techniques de type avionique ¹² en utilisant la notion de courbure discrète d'épaisseur ν décrite dans le paragraphe précédent [11,7]. Notre approche a la spécificité de pouvoir s'adapter au bruit et à la taille des éléments à détecter dans les documents techniques.

PRINCIPE GÉNÉRAL DE LA MÉTHODE

Soit \mathcal{C} une courbe 8-connexe. Nous définissons son **profil de rayons de courbure d'épaisseur** ν , appelé $\mathcal{P}_\nu(\mathcal{C})$, défini comme la suite des rayons de courbure d'épaisseur ν calculés en chaque point de \mathcal{C} . L'objectif est de segmenter \mathcal{C} en un ensemble de primitives (arcs ou segments) connectées. Pour ceci, nous utilisons le profil $\mathcal{P}_\nu(\mathcal{C})$. Des filtres sont ensuite appliqués sur $\mathcal{P}_\nu(\mathcal{C})$ afin de supprimer le bruit et se focaliser sur l'information pertinente. Ce nouveau profil est appelé $\mathcal{P}_\nu^f(\mathcal{C})$. Alors les points de rupture sont déduits à partir de $\mathcal{P}_\nu^f(\mathcal{C})$. Entre chaque paire de points de rupture successifs, nous calculons les caractéristiques des primitives localisées sur \mathcal{C} . Pour une meilleure compréhension du comportement de la méthode, les étapes sont illustrées par un exemple donné à la figure 2.25 gauche. Des croix ont été ajoutées pour localiser le début et la fin des arcs ou des segments.

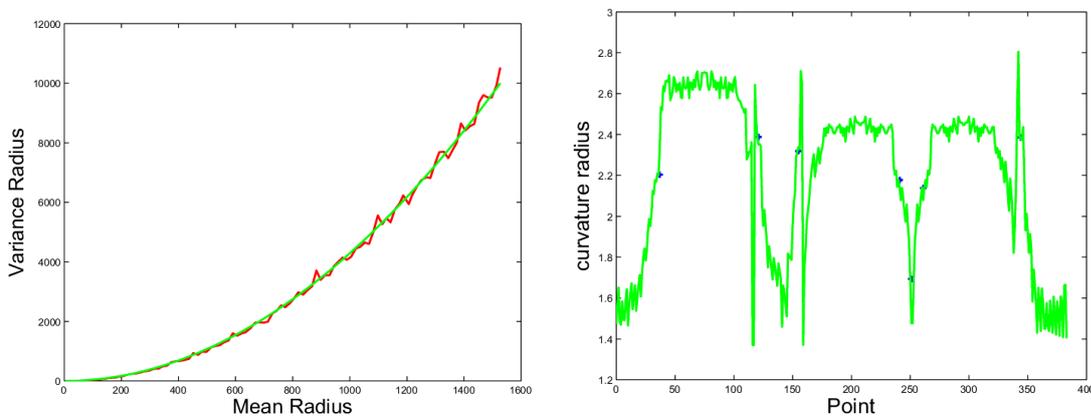
¹²Ce travail se situe dans le cadre du consortium européen FRESH auquel participe l'équipe QGAR et qui porte sur l'analyse et l'interprétation de documents techniques de type avionique.

FIG. 2.25 – Exemple d'une courbe \mathcal{C} et de son profil de rayons de courbure associé $\mathcal{P}_2(\mathcal{C})$.

ÉTAPE DE FILTRAGE

La représentation discrète de la courbe \mathcal{C} entraîne des perturbations dans le profil de rayons de courbure (comme le montre la figure 2.25 droite). Pour extraire l'information pertinente, nous utilisons successivement 3 filtres décrits brièvement ci-dessous (pour plus de détails, consulter la publication [11]) :

- **filtre quadratique** : ce filtre a été élaboré par J.-P. Salmon, doctorant dans l'équipe QGAR. Une étude de l'évolution du bruit généré sur les profils de rayon de courbure obtenus à partir de 100 cercles discrétisés de rayons variant de 5 à 1490 pour une épaisseur 2 a été réalisée. Sur la figure 2.26 gauche (courbe rouge), nous avons représenté sur l'axe Ox les rayons moyens obtenus à partir des profils des rayons de courbure d'épaisseur 2 des cercles discrets originaux et sur Oy , la variance v des rayons obtenus sur le profil des rayons de courbure associé au cercle discret correspondant. Après analyse, il est apparu que la relation $v = K * r^2$ était vérifiée avec $K = 0.004286$ (courbe verte sur la figure 2.26 gauche). Le filtre suivant dit quadratique en a été déduit : $\frac{\sqrt{r}}{K}$. Une illustration de son application sur le profil de rayon de courbure de la figure 2.25 est donnée à la figure 2.26 droite.

FIG. 2.26 – Variations et application du filtre quadratique sur $\mathcal{P}_2(\mathcal{C})$.

- **filtre médian** : l'application d'un filtre médian permet de supprimer les perturbations qui affectent le calcul du rayon de courbure autour des points de rupture. Les effets de l'application d'un tel filtre (appliqué sur une fenêtre de 11 points) sur l'exemple de profil étudié sont donnés à la figure 2.27 gauche.

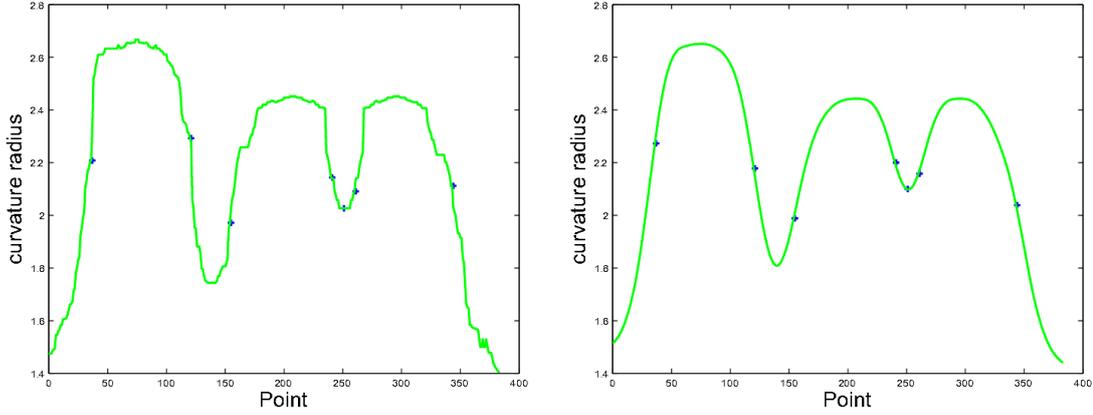


FIG. 2.27 – Illustration des profils résultant de l'application des filtres Médian et Moyenneur($\mathcal{P}_2^f(\mathcal{C})$).

- **filtre moyenneur** : l'application de ce filtre (sur une fenêtre de 11 points) finit de lisser le profil obtenu afin de faciliter l'étape de segmentation qui suit (illustration à la Figure 2.27 gauche).

ETAPE DE SEGMENTATION DE $\mathcal{P}_\nu^f(\mathcal{C})$

Nous considérons ici le profil résultant des différents filtrages $\mathcal{P}_\nu^f(\mathcal{C})$. L'objectif est de fournir une série de points connectés ayant un rayon de courbure similaire. Nous calculons alors la **valeur absolue du gradient** sur $\mathcal{P}_\nu^f(\mathcal{C})$. On associe alors les maxima et les extrémités de la séquence aux points de rupture (voir Figure 2.28, les points rouges sont les points de rupture calculés et les points bleus sont ceux de la figure initiale). En effet, ils correspondent à des points situés au milieu de fortes variations. Nous avons considéré la partie de la courbe \mathcal{C} entre chaque couple de points de rupture successifs et nous avons obtenu : $\mathcal{C} = \bigcup (\mathcal{C}_i)_{i=1, \dots, p}$ avec p points de rupture.

Pour chaque \mathcal{C}_i correspond un profil associé $\mathcal{P}_\nu(\mathcal{C}_i)$, la valeur médiane du rayon de courbure $m_{\mathcal{P}_\nu(\mathcal{C}_i)}$ est recherchée. Deux cas sont alors possibles. Si $m_{\mathcal{P}_\nu(\mathcal{C}_i)}$ est élevée, c'est-à-dire, plus grande que la largeur de la boîte englobante, les points associés à \mathcal{C}_i sont supposés appartenir à un segment, sinon ils sont supposés appartenir à un même arc.

Les extrémités des segments sont positionnées à partir des points de début et de fin associés à \mathcal{C}_i . Concernant les arcs, les rayons sont définis par la valeur $m_{\mathcal{P}_\nu(\mathcal{C}_i)}$. Un point de \mathcal{C}_i avec une courbure égale à $m_{\mathcal{P}_\nu(\mathcal{C}_i)}$ est considéré et les coordonnées du centre de l'arc sont calculées à partir de cette valeur de rayon et de la médiatrice au segment $[p_l, p_r]$ avec p_l et p_r les points utilisés dans le calcul de la courbure, (voir section 2.3.2). Le début et la fin de l'arc sont obtenus en calculant l'intersection entre l'arc trouvé et les deux droites passant respectivement par les points extrémités de \mathcal{C}_i et le centre calculé, ce qui peut engendrer des décalages des extrémités.

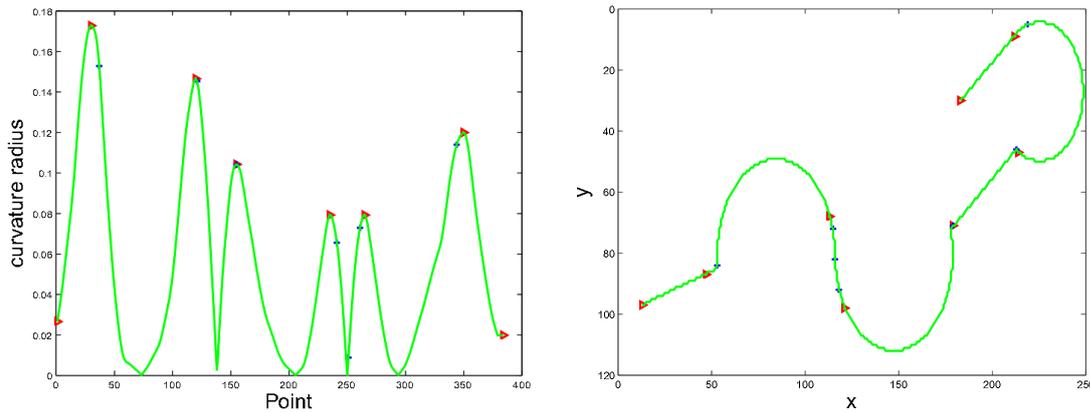


FIG. 2.28 – Extraction d'extrema.

D'autres techniques de reconstruction des arcs de cercles, plus coûteuses, ont aussi été proposées pour permettre un meilleur recalage avec les extrémités. Par exemple, on peut procéder de manière identique au calcul d'un centre potentiel sur un échantillon de points de C_i puis calculer le centre de gravité des centres obtenus. La figure 2.31 montre le résultat final et le bon comportement de notre approche sur de tels exemples.

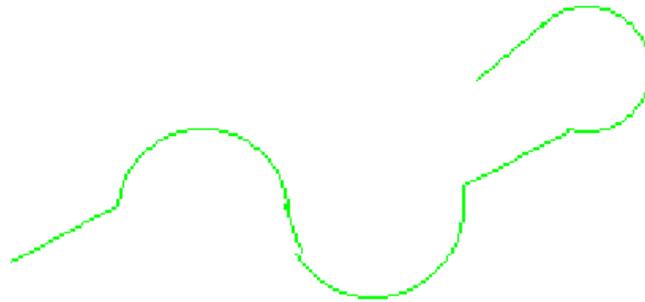


FIG. 2.29 – Résultat de la reconstruction.

Nous constatons aussi une stabilité de la méthode employée qui est :

- invariante par rotation et translation des contours étudiés,
- robuste au bruit ; le paramètre d'épaisseur permet un contrôle du bruit.

Plusieurs tests ont été effectués sur des documents techniques et se sont révélés satisfaisants (voir [11]), cependant, la technique de reconstruction est encore à améliorer.

2.3.5 Estimateur de la longueur d'une courbe discrète bruitée

Plusieurs travaux ont eu pour objectifs de comparer et d'évaluer différents estimateurs de longueurs de courbes discrètes 2D [Coe02, CK04] et [KR04] chapitre 10. Trois types d'estimateurs de longueur apparaissent :

- *Approche utilisant des métriques locales* ; elles consistent à estimer la longueur de chaque déplacement minimal dans la grille des entiers puis à sommer ces estimations sur l'ensemble de la courbe à analyser (par exemple [DS87, Ver91]).
- *Approche reposant sur l'approximation polygonale de la courbe à mesurer* ; la courbe discrète doit être découpée en segments, cela correspond à une approximation po-

lygonale de la courbe et sa longueur est alors estimée comme la somme des longueurs des segments la constituant. Cet estimateur dépend donc de l'approximation polygonale effectuée, par exemple celle reposant sur le découpage en segments de droites naïves de longueur maximale [7] ou l'algorithme de segmentation de V. Kovalevsky [Kov90] ou encore l'évaluation du polygone de longueur minimale (dit MLP : Minimum Length Polygon) passant à l'intérieur des pixels de la courbe [KKY99].

- **Approche basée sur l'intégration des normales discrètes** ; D. Coeurjolly [Coe02, CK04] a proposé une version discrète de la formule de calcul de la longueur d'une courbe basée sur l'intégration du champ de normale d'une courbe $\gamma : L(\gamma) = \int_{\gamma} \mathbf{n}(s) ds$. Pour cela il utilise un estimateur de tangente discrète comme par exemple [Via96b].

Différentes évaluations ont été réalisées dans [Coe02, CK04] et [KR04] chapitre 10 avec une analyse expérimentale ainsi qu'une analyse théorique basée sur la notion de convergence asymptotique. Il a été montré que les estimateurs reposant sur des métriques locales ne possèdent pas la propriété de convergence asymptotique, par contre, elle a été prouvée pour les deux autres approches.

Naturellement, dans le cadre de notre travail sur les courbes bruitées, la seconde approche peut être adaptée aux courbes bruitées en fixant une épaisseur de segmentation et en utilisant un des algorithmes décrits dans les sections 2.2.1 et 2.2.2 de ce chapitre qui décomposent une courbe discrète en segments flous. Une illustration de l'intérêt de cette approche se trouve à la figure 2.30 ; le polygone formé des points $(0, 0)$, $(45, 7)$, $(25, 31)$ et $(3, 16)$ a été discrétisé puis bruité avec l'algorithme proposé dans l'introduction de ce chapitre. Le périmètre de ce polygone est égal à 119.7. Avec l'approche utilisant la segmentation en segments de droites discrètes naïves la valeur obtenue est égale à 129.5 alors qu'avec les segments flous d'épaisseur 3, nous obtenons 119.4 ce qui est beaucoup plus proche de la valeur initiale.

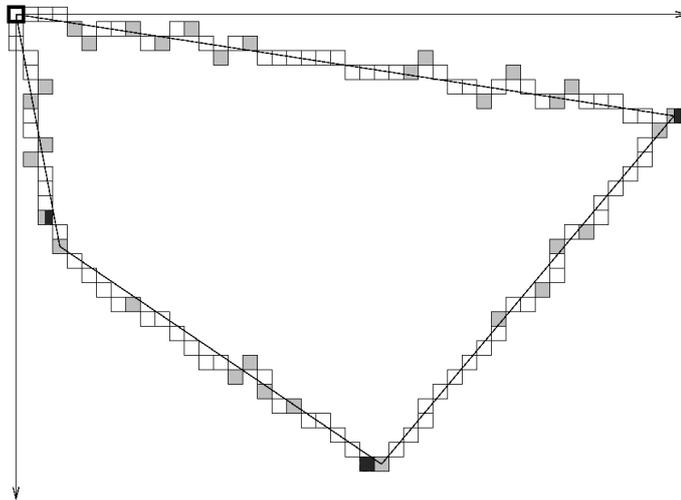


FIG. 2.30 – Illustration du calcul de la longueur d'une courbe bruitée avec une méthode de segmentation : en gris clair les extrémités obtenues avec l'algorithme utilisant les segments de droites discrètes naïves et en noir celles obtenues avec une segmentation en segments flous d'épaisseur 3.

2.3.6 Extension en dimension 3

Les travaux décrits dans les paragraphes précédents ont été étendus en dimension 3.

Mesure de la longueur d'une courbe discrète 3D bruitée

Peu d'auteurs se sont intéressés au calcul de la longueur d'une courbe discrète 3D, paramètre pourtant très utile dans des applications utilisant des objets en voxels comme par exemple en imagerie médicale. R. Klette et T. Bulow [BK00, BK02] ont proposé une version 3D de l'approche MLP évoquée pour la dimension 2 dans le paragraphe 2.3.5. La courbe considérée dans cet algorithme est fermée et strictement 6-connexe. Sa représentation en voxels est analysée : le polygone de longueur minimale passant à l'intérieur des voxels est calculé et la longueur de la courbe discrète en est déduite. De nombreux problèmes ont été détectés dans cet algorithme, des erreurs furent corrigées et des améliorations apportées qui produisirent plusieurs articles [LK05, LK06, LK07].

En collaboration avec David Coeurjolly (LIRIS, Lyon) et Olivier Teytaud (Institut des Sciences Cognitives, Bron), nous avons élaboré un algorithme simple de calcul de la longueur de courbes discrètes 3D reposant sur les droites discrètes 3D. De nombreuses définitions ont été proposées [Kim83, Kau87, FR95, CK97, JK97, And03], dans le cadre du travail réalisé, nous avons utilisé une définition arithmétique des droites discrètes 3D et l'algorithme linéaire de segmentation de courbes 3D que j'ai élaborés durant ma thèse [43].

Le principe de l'algorithme est le suivant : la courbe 3D est découpée en segments de droites de longueurs maximales et la longueur de la courbe est calculée en fonction de la longueur de la ligne polygonale obtenue. La convergence de cette technique d'estimation de longueur a été prouvée en démontrant que l'erreur entre la longueur d'une courbe, notée \mathcal{C} , et la longueur de cette courbe discrétisée, notée \mathcal{C}_d , dans une grille de taille δ est telle que :

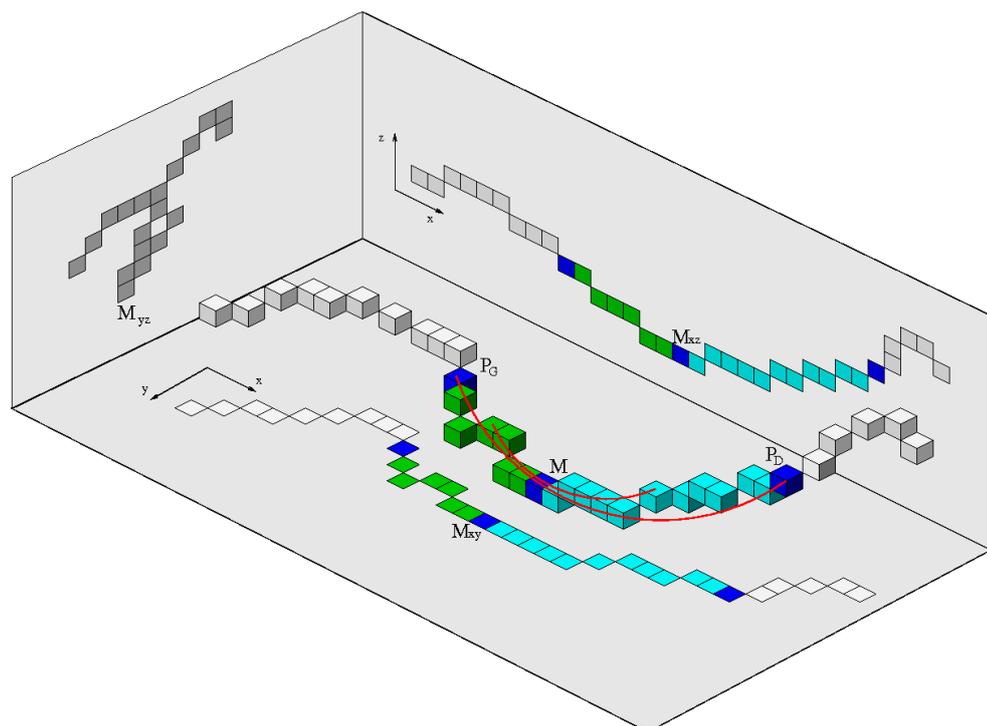
$$|l(\mathcal{C}) - l(\mathcal{C}_d)| \leq O(\delta)$$

Cet estimateur de longueur d'une courbe discrète 3D étend l'approche 2D reposant sur une approximation polygonale. Avec une démarche identique à la dimension 2, l'utilisation de l'algorithme de segmentation de courbes 3D en segments flous 3D (voir section 2.2.4) permet d'obtenir un estimateur de longueur adapté aux courbes bruitées.

Evaluation de la courbure en chaque point d'une courbe discrète 3D bruitée

D. Coeurjolly et S. Svensson [CS03] ont étendu à la dimension 3 le principe du calcul de courbure 2D reposant sur le calcul du cercle osculateur en utilisant la reconnaissance des segments de droites discrètes naïves 3D [17,43]. Nous reprenons cette idée et, en fonction de l'approche utilisée, nous définissons comme en 2D la notion de **courbure 3D d'épaisseur ν ou d'ordre d** . Le principe est similaire, par exemple pour l'approche géométrique, le calcul de la courbure d'épaisseur ν au point M d'une courbe \mathcal{C}_{3D} s'effectue de la manière suivante : la demi-tangente à droite du point M est évaluée en reconnaissant, à droite de M , le plus long segment flou 3D d'épaisseur ν , de point extrémité P_D , on procède de même à gauche, on obtient alors le point extrémité P_G . La courbure d'épaisseur ν au point M est égale à l'inverse du rayon du cercle circonscrit aux points $[P_G, M, P_D]$. Un exemple du calcul de la courbure d'ordre 1 et 2 se trouve à la figure 2.31 : à l'ordre 2, $R_2 = 10$ et la courbure $C_2 = 0.1$ et à l'ordre 1, $R_1 = 8$ et la courbure $C_1 = 0.125$.

En bioinformatique, où nous nous intéressons au calcul de la courbure de l'ADN, les modèles discrets de représentation de la structure 3D de l'ADN nécessitent le développement d'algo-

FIG. 2.31 – Rayon de courbure d'ordre 1 et 2 au point M .

rithmes spécifiques de géométrie discrète pour répondre au problème posé. Nous avons adapté à ce problème les algorithmes présentés précédemment (voir chapitre 4, section 4.2).

2.4 Conclusion

La notion de segments flous a été présentée dans ce chapitre ainsi que deux méthodes de reconnaissance l'une entièrement arithmétique et l'autre utilisant des techniques simples de géométrie algorithmique. La seconde est plus performante et s'adapte aux courbes discrètes déconnectées et non simples. Des algorithmes de polygonalisation de courbes discrètes bruitées ont été déduits de ces approches, notre objectif étant d'obtenir une représentation polygonale la plus représentative de la courbe discrète bruitée en faisant varier un paramètre d'épaisseur. Ces techniques ont été utilisées pour la détection de paramètres géométriques sur des courbes bruitées comme les tangentes, la courbure ou encore l'estimation de la longueur d'une courbe discrète. Une application à la détection de segments et d'arcs de cercles dans les documents techniques a été réalisée en combinant le calcul de la courbure pour une épaisseur donnée en chaque point de la courbe et en appliquant ensuite plusieurs filtres pour extraire des informations pertinentes du graphe de courbure filtré.

Nous souhaitons orienter nos recherches afin de détecter automatiquement le paramètre d'épaisseur le mieux adapté au bruit présent dans la courbe étudiée. Un premier travail a été réalisé dans cette direction en proposant une segmentation multi-niveaux qui permet d'obtenir une polygonalisation reflétant la forme générale de la courbe mais des problèmes persistent et nécessitent une étude approfondie.

De plus, d'autres types de polygonalisation tenant compte du bruit peuvent aussi être déterminés, par exemple en respectant les concavités/convexités de la courbe discrète. En utilisant une démarche similaire à celle décrite dans la section 1.2 du chapitre 1, une étude des segments flous maximaux d'une courbe discrète pour une épaisseur donnée pourra nous aider dans cette voie.

Une grande partie des résultats obtenus dans ce chapitre en dimension 2 a pu être étendue pour les courbes discrètes en dimension 3. L'étude de la torsion d'une courbe discrète bruitée reste encore à réaliser.

Chapitre 3

Eléments de géométrie discrète pour l'étude des surfaces discrètes bruitées

Sommaire

3.1	Introduction	99
3.2	Morceaux flous de plans discrets	101
3.2.1	Définitions	101
3.2.2	Reconnaissance incrémentale	102
3.3	Extraction de paramètres géométriques	106
3.3.1	Patch d'épaisseur ν centré en un point	106
3.3.2	Normale d'épaisseur ν en un point	107
3.3.3	Aire d'un patch d'épaisseur ν	108
3.3.4	Estimateur de forme	109
3.4	Segmentation en morceaux flous de plans discrets	113
3.5	Conclusion	115

3.1 Introduction

Plusieurs auteurs se sont intéressés aux objets discrets 3D ainsi qu'aux définitions de surfaces discrètes associées selon les propriétés souhaitées (par exemple [USH82, MR81, Her92, Udu94, Mal97]). Dans ce chapitre, comme dans le chapitre précédent, nous nous intéressons aux objets discrets pouvant être bruités. En effet, dans certaines applications, les données qui sont obtenues par des scanners ou des IRM sont souvent bruitées. Nous souhaitons obtenir sur ce type de données des propriétés géométriques comme l'estimation de la normale, de la courbure en un point du bord de l'objet ou encore la décomposition en primitives géométriques de ces objets discrets.

Dans le chapitre 1, section 1.3 de ce mémoire, un rapide état de l'art a été effectué au sujet des estimateurs de normales en dimension 3. Certains peuvent s'adapter aux données bruitées, citons par exemple L. Papier [PF98], qui propose d'estimer la normale en un point p de la surface en utilisant une moyenne pondérée des normales élémentaires aux surfels des voxels d'un voisinage de p (*ombrelle d'ordre n*). Le paramètre n permet de définir la taille du voisinage considéré. Dans [LMR96, Len97], A. Lenoir propose de calculer les tangentes 2D sur les courbes des coupes parallèles aux plans de coordonnées de la surface de l'objet en utilisant un calcul de dérivées partielles (implanté par des produits de convolution). Il estime ensuite la normale et la courbure

des points du bord d'un objet discret $3D$ à partir de ces tangentes. Un paramètre d'échelle permet de définir le voisinage considéré.

En 2001, G. Thürmer [Thu01] a proposé une autre technique de calcul améliorant son estimateur présenté en 1997 [TW97], celui-ci est basé sur le volume présent au voisinage du point dont la normale est calculée. Le principe de calcul de la normale au point p est le suivant : la méthode de [TW97] est appliquée dans un premier temps en considérant un voisinage de taille fixée autour de p et en sommant tous les vecteurs issus de p dans ce voisinage, un premier vecteur normal $n_p = (a, b, c)$ est obtenu. Un paramètre h est alors introduit et le plan discret P de caractéristiques (a, b, c, h) est considéré au voisinage $N_e(p)$ de p de taille e^*e puis la normale au point p est recalculée en considérant les points q de ce voisinage ainsi que les estimations de normale effectuées dans le pré-traitement et en pondérant par sa distance $w(q, p)$ à p . La formule suivante, nécessitant de choisir 2 paramètres, permet d'obtenir une estimation de la normale au point p placé sur le bord d'un objet S :

$$n_p^s = \frac{\sum_{\forall q \in (N_e(p) \cap S) | q \in P, (\widehat{n_q, n_p}) < \frac{\pi}{2}} w_s(q, p) n_q}{|\sum_{\forall q \in (N_e(p) \cap S) | q \in P, (\widehat{n_q, n_p}) < \frac{\pi}{2}} w_s(q, p) n_q|}$$

Ces trois méthodes permettent ainsi de prendre en compte les surfaces légèrement bruitées, néanmoins les voisinages considérés ne sont pas adaptatifs contrairement à l'approche que nous présentons dans cette section. D'autres estimateurs ont été proposés pour l'aire et la courbure, par exemple [KS01, CFTT03, HK03].

La segmentation d'objets discrets en morceaux de plans discrets arithmétiques a été entreprise par quelques auteurs, certains ont proposé des segmentations ou des polyédrisations réversibles¹³ comme Ph. Borianne [BF94] et L. Papier [FP99] avec J. Françon, I. Sivignon et al. [Siv04, SDC04, CDJS06] ou encore M. Dexet et al. [DCA06]. Des polyédrisations non-réversibles d'objets discrets ont aussi été proposées [KS01, BM03]. Cependant, ces techniques de segmentation reposent sur la reconnaissance des morceaux de plans discrets arithmétiques naïfs ou standards et une segmentation du bord d'objets discrets bruités engendre de multiples facettes et n'est donc pas satisfaisante. Une illustration de ce phénomène se trouve à la figure 3.1 où l'algorithme¹⁴ d'I. Sivignon [SDC04, Siv04] a été testé sur un cube avec et sans bruit.

Notre objectif est de définir une segmentation adaptée aux objets discrets bruités en utilisant une primitive géométrique plus souple que les plans discrets naïfs ou standards. De plus, nous souhaitons orienter le découpage selon la forme de l'objet étudié en utilisant des caractéristiques géométriques pré-calculées.

C'est pourquoi, dans le même esprit que les segments flous, nous avons introduit une nouvelle notion relative aux données discrètes tridimensionnelles et pouvant s'adapter aux données bruitées : **les morceaux flous de plans discrets**. Le travail présenté dans ce chapitre est récent et a été réalisé avec Laurent Provot, d'abord dans le cadre d'un stage de DEA [Pro05] puis pendant une thèse sur ce sujet. A partir de cette nouvelle notion, nous avons proposé des algorithmes de reconnaissance et des estimateurs de paramètres géométriques de surfaces discrètes pouvant être bruitées. Ces outils sont utilisés pour guider la segmentation de surfaces discrètes en morceaux

¹³Un polyédrisation réversible permet de retrouver les données discrètes d'origine après discrétisation du polyèdre obtenu.

¹⁴Le programme correspondant est disponible à l'adresse <http://liris.cnrs.fr/isabelle.sivignon/DSD.html>

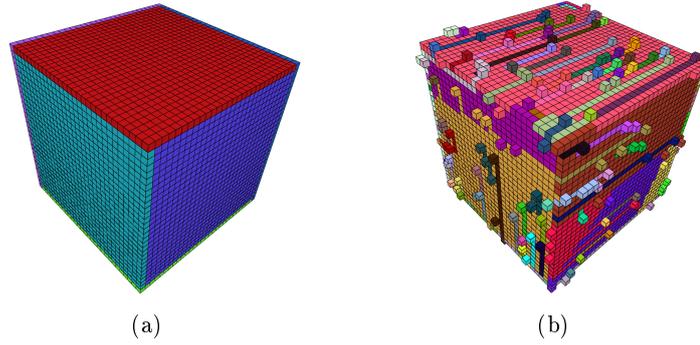


FIG. 3.1 – Segmentation utilisant l’algorithme d’I. Sivignon d’un cube d’arête de 25 voxels (a) non bruité et (b) bruité.

flous de plans discrets. Plusieurs stratégies de segmentation sont proposées.

Les travaux de ce chapitre ont été décrits dans trois articles [10,46,47] dont deux sont en cours de soumission. La publication [10] a été réalisée en collaboration avec Lilian Buzer.

3.2 Morceaux flous de plans discrets

3.2.1 Définitions

Nous proposons une généralisation de la notion de plan discret arithmétique en admettant que des points soient manquants. Pour cela nous introduisons la notion de **plan englobant** relativement à une norme N sur \mathbb{R}^3 :

Définition 3.1 Soit \mathcal{E} un ensemble de points de \mathbb{Z}^3 , le plan discret $\mathcal{P}(a, b, c, \mu, \omega)$ est un **plan englobant de \mathcal{E}** si tous les points de \mathcal{E} appartiennent à \mathcal{P} . La valeur $\frac{\omega-1}{N(a,b,c)}$ est appelée **épaisseur de $\mathcal{P}(a, b, c, \mu, \omega)$** .

L’épaisseur peut être interprétée de la manière suivante :

1. Si $N = \|\cdot\|_2$, l’épaisseur $\frac{\omega-1}{N(a,b,c)} = \frac{\omega-1}{\sqrt{a^2+b^2+c^2}}$ représente la distance euclidienne entre les deux plans d’appui du plan englobant $\mathcal{P}(a, b, c, \mu, \omega)$.
2. Si $N = \|\cdot\|_\infty$, l’épaisseur $\frac{\omega-1}{N(a,b,c)} = \frac{\omega-1}{\max(|a|,|b|,|c|)}$ représente la distance selon la direction principale du vecteur (a, b, c) .

Définition 3.2 Soit \mathcal{E} un ensemble de points de \mathbb{Z}^3 , un plan englobant de \mathcal{E} est dit **optimal** si son épaisseur est minimale.

La définition d’un morceau flou de plan discret d’épaisseur ν (cf. Figure 3.2) est alors la suivante :

Définition 3.3 Un ensemble de points \mathcal{E} de \mathbb{Z}^3 est un **morceau flou de plan discret d’épaisseur ν** si l’épaisseur de son plan englobant optimal est inférieure ou égale à ν .

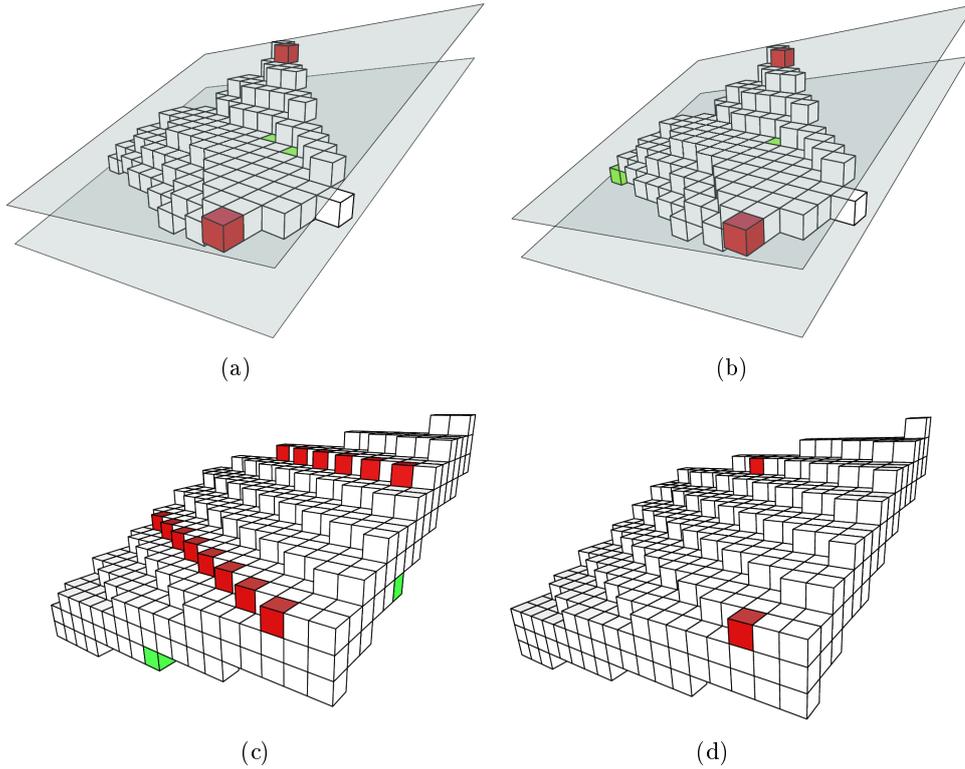


FIG. 3.2 – Un morceau flou de plan discret d'épaisseur 3 (a et b), ses plans englobants optimaux (c) pour la norme euclidienne : $\mathcal{P}_2(4, 8, 19, -80, 49)$ et l'épaisseur de $\mathcal{P}_2 = 2.28$ (d) pour la norme infinie : $\mathcal{P}_\infty(31, 65, 157, -680, 397)$ et l'épaisseur de $\mathcal{P}_\infty = 2.52$. Les plans et points d'appui correspondants de \mathcal{P}_2 et \mathcal{P}_∞ sont respectivement représentés sur (a,c) et (b,d).

A partir de ces définitions, nous avons proposé [10] en collaboration avec Lilian Buzer (ESIEE, Noisy-le-Grand) deux algorithmes pour résoudre le problème de la reconnaissance des morceaux flous de plans discrets. Pour un ensemble de points \mathcal{E} de \mathbb{Z}^3 et une épaisseur ν fixée, ces algorithmes déterminent si \mathcal{E} est un morceau flou de plan discret d'épaisseur ν et fournissent les caractéristiques du plan englobant optimal de \mathcal{E} .

3.2.2 Reconnaissance incrémentale

La première approche est *géométrique* et l'épaisseur considérée repose sur la norme euclidienne. Le principe consiste à calculer la largeur d'un ensemble de points dans un espace de dimension 3. Nous avons proposé un algorithme incrémental de reconnaissance de morceaux flous de plans discrets en adaptant la méthode développée par B. Gärtner et T. Herrmann [GH01] et en utilisant une mise à jour incrémentale de l'enveloppe convexe [BKO00].

La seconde approche est basée sur la *programmation linéaire* et permet de résoudre le problème en considérant la norme $\|\cdot\|_\infty$. Elle repose sur une adaptation de l'*algorithme du simplexe* [Chv83] à notre problème.

Ces 2 algorithmes ont été programmés et testés par Laurent Provot et sont accessibles à

l'adresse <http://www.loria.fr/~debled/BlurredPlane>.

L'approche géométrique ayant été utilisée dans la suite de nos travaux, le principe de l'algorithme incrémental est décrit ci-après. Cette approche s'inspire d'une méthode permettant de calculer la largeur d'un ensemble E de points de \mathbb{R}^3 [HT88, GH01]. Les définitions suivantes y sont proposées :

Définition 3.4 Soit E un ensemble de points de \mathbb{R}^3 et \mathcal{P} un plan réel. On dit que \mathcal{P} est un **plan support** de E si tous les points de E appartiennent à un seul des demi-plans délimités par \mathcal{P} et tels que $\mathcal{P} \cap E \neq \emptyset$.

Définition 3.5 Soit E un ensemble de points de \mathbb{R}^3 . On appelle **largeur** de E , la plus petite distance (euclidienne) entre deux plans support parallèles de E .

Si E est un ensemble de points de \mathbb{Z}^3 alors deux plans support parallèles dont la distance est égale à la largeur de E représentent les plans d'appui d'un plan englobant optimal de E , et la largeur de E est égale à l'épaisseur de ce plan englobant optimal. Par conséquent, **résoudre le problème de la reconnaissance de morceaux flous de plans discrets revient à calculer la largeur de E et à retrouver deux plans support parallèles qui lui sont associés.**

PRINCIPE DU CALCUL DE LA LARGEUR D'UN ENSEMBLE E

Nous recherchons deux plans parallèles $P_1 : \alpha x + \beta y + \gamma z + \delta_1 = 0$ et $P_2 : \alpha x + \beta y + \gamma z + \delta_2 = 0$ qui minimisent la distance $\frac{|\delta_2 - \delta_1|}{\sqrt{\alpha^2 + \beta^2 + \gamma^2}}$ entre P_1 et P_2 et tels que pour tout point $p(p_x, p_y, p_z) \in E$

$$\begin{aligned} \alpha p_x + \beta p_y + \gamma p_z + \delta_1 &\leq 0 \\ \alpha p_x + \beta p_y + \gamma p_z + \delta_2 &\geq 0 \end{aligned}$$

Comme l'enveloppe convexe d'un ensemble E est l'intersection de tous les demi-espaces contenant entièrement E , la largeur de E est égale à la largeur de son enveloppe convexe [HT88]. La notion de **paires antipodales** est ensuite introduite.

Définition 3.6 Soient E un ensemble de points et $\mathcal{CH}(E)$ son enveloppe convexe. Deux arêtes a_1 et a_2 de $\mathcal{CH}(E)$ sont dites **antipodales** s'il existe deux plans support de E parallèles, dont l'un contient a_1 et l'autre contient a_2 . On dit aussi que a_1 et a_2 forment une **paire arête-arête antipodale**. On définit de la même manière les paires **sommet-sommet**, **sommet-arête**, **sommet-face**, **arête-face** et **face-face antipodales**.

Dans [HT88], M.E. Houle and G.T. Toussaint montrent que pour calculer la largeur de E il suffit de s'intéresser aux plans support définis par une paire antipodale **sommet-face** ou **arête-arête**. Notre objectif est donc d'énumérer toutes les paires antipodales **sommet-face** et **arête-arête** de $\mathcal{CH}(E)$ et de retenir celle qui définit des plans support parallèles dont la distance est minimale.

Dans [GH01], B. Gärtner et T. Herrmann généralisent la méthode du *rotating caliper* en dimension 3, afin d'énumérer l'ensemble de ces paires antipodales.

Une face de départ f est choisie sur l'enveloppe convexe $\mathcal{CH}(E)$ et les sommets antipodaux $V = \{v_1, \dots, v_k\}$ sont calculés. Une rotation des plans P_1 et P_2 , supportant respectivement V et f , est effectuée autour d'une arête a incidente à f , jusqu'à ce que P_2 contienne l'autre face

f' incidente à a . Durant la rotation, le parallélisme entre les plans P_1 et P_2 est conservé et on reporte toutes les paires arête-arête antipodales contenant a ainsi que les paires face-sommet antipodales contenant f' .

On remarque qu'étant donné une paire sommet-arête antipodale (w, a) et des plans parallèles P_1 et P_2 contenant respectivement w et a , au cours d'une rotation de P_2 autour de a , deux événements peuvent se produire :

1. P_2 rencontre une nouvelle face f' incidente à a , dans ce cas une paire sommet-face antipodale (w, f') est détectée,
2. P_1 rencontre un nouveau sommet v , dans ce cas une paire arête-arête antipodale $((wv), a)$ est détectée.

Une rotation autour d'une arête de $\mathcal{CH}(E)$ permet ainsi de relever toutes les paires antipodales qui contiennent cette arête et toutes les paires qui contiennent les deux faces incidentes à l'arête. En effectuant la rotation autour de toutes les arêtes de $\mathcal{CH}(E)$, on relève donc toutes les paires antipodales sommet-face et arête-arête possibles de $\mathcal{CH}(E)$. Au moins l'une d'elles définit deux plans support parallèles dont la distance \mathcal{L} est minimale et qui correspond à la largeur de E .

\mathcal{L} représente l'épaisseur d'un plan englobant optimal $\mathcal{P}(a, b, c, \mu, \omega)$ de E , par conséquent, si $\mathcal{L} \leq \nu$, E est un morceau flou de plan discret d'épaisseur ν . De plus, les plans support correspondent aux plans d'appui du plan englobant optimal de E , nous obtenons alors $a = \alpha$, $b = \beta$ et $c = \gamma$, $\omega = |\delta_2 - \delta_1| + 1$ et $\mu = \min(-\delta_1, -\delta_2)$ puisque l'équation d'un plan d'appui est $ax + by + cz = \mu$.

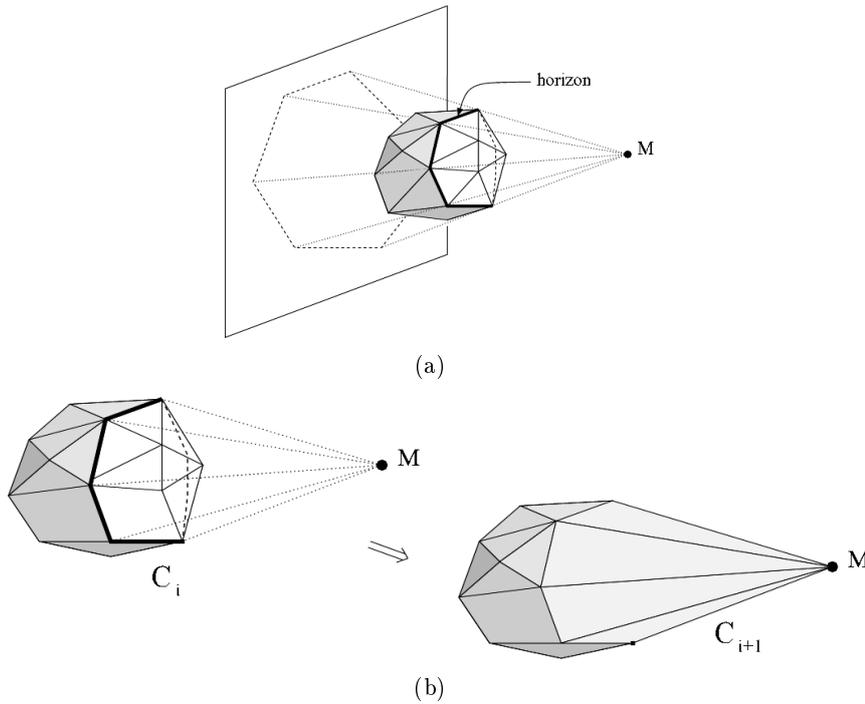


FIG. 3.3 – (a) Ligne d'horizon de l'enveloppe convexe; (b) Evolution de l'enveloppe convexe après ajout du point M .

ALGORITHME INCRÉMENTAL

Nous proposons une version incrémentale de la méthode décrite précédemment, de façon à obtenir un algorithme qui, à chaque ajout de point à un ensemble E , retourne les caractéristiques du plan englobant optimal de E .

Dans le cadre du processus incrémental, l'ensemble E ne diffère que d'un point entre chaque ajout de point à E et nous pouvons donc utiliser un calcul incrémental de l'enveloppe convexe ([BKO00] pp 235–246) :

À une étape i du processus, nous avons une enveloppe convexe C_i . Nous ajoutons un point M . S'il se trouve à l'intérieur de C_i alors nous ne faisons rien. Dans le cas contraire nous recherchons toutes les faces de C_i visibles depuis M . Cet ensemble de faces forme une région délimitée par une courbe qu'on appelle *horizon* (cf. Figure 3.3(a)). Ces faces sont alors supprimées et chaque extrémité des segments de l'horizon est reliée au nouveau point M (cf Figure 3.3(b)) afin de former de nouvelles faces triangulaires. Certaines des nouvelles faces de l'enveloppe convexe peuvent être coplanaires à des faces non visibles depuis M dans C_i , il faut alors fusionner ces faces coplanaires. Le polyèdre résultant est la nouvelle enveloppe convexe C_{i+1} .

De plus, il n'est pas nécessaire de calculer la largeur de l'ensemble à chaque étape, en effet, pour chaque point ajouté, nous connaissons les caractéristiques a, b, c, μ, ω du plan englobant optimal \mathcal{P}_e de E , et nous pouvons calculer la valeur du reste $r_{\mathcal{P}_e}(M) = ax_M + by_M + cz_M - \mu$ au point M ajouté en fonction de \mathcal{P}_e . Dans le cas où $r_{\mathcal{P}_e}(M) \in [0, \omega - 1]$ alors $M \in \mathcal{P}_e$, donc il est inutile de recalculer la largeur de E . Nous obtenons alors l'Algorithme 15 permettant une reconnaissance incrémentale d'un morceau flou de plan discret d'épaisseur ν .

Algorithme 15 : Etape de la reconnaissance incrémentale d'un morceau flou de plan discret d'épaisseur ν

Données : Un ensemble $\mathcal{E} \in \mathbb{Z}^3$, l'enveloppe convexe C de \mathcal{E} , les coefficients a, b, c, μ et ω du plan englobant optimal de \mathcal{E} , ν l'épaisseur maximale autorisée

Entrée : $M \in \mathbb{Z}^3$

Résultat : Les données mises à jour après l'ajout de M et le booléen *estPlan* vrai si $\mathcal{E} \cup \{M\}$ est un morceau flou de plan discret d'épaisseur ν , faux sinon

```

1 begin
2    $\mathcal{E} = \mathcal{E} \cup M$ ;
3   Mise à jour de  $C$  selon le processus incrémental;
4    $r_M = ax_M + by_M + cz_M - \mu$ ;
5   si  $r_M \notin [0, \omega - 1]$  alors
6      $\langle \alpha, \beta, \gamma, \delta_1, \delta_2 \rangle = \text{CalculPlanSupportMin}(C)$ ;
7      $a = \alpha$ ;
8      $b = \beta$ ;
9      $c = \gamma$ ;
10     $\mu = \min(-\delta_1, -\delta_2)$ ;
11     $\omega = |\delta_2 - \delta_1| + 1$ ;
12     $estPlan = \frac{\omega - 1}{\sqrt{a^2 + b^2 + c^2}} \leq \nu$ ;
13 end
```

La procédure `CalculPlanSupportMin(C)` appelée à la ligne 6 de l'Algorithme 15 calcule les plans support parallèles de C entre lesquels la distance euclidienne est minimale grâce à la mé-

thode décrite précédemment et les coefficients de ces plans sont alors retournés.

Complexité :

Dans [GH01], Gärtner et Herrmann démontrent que la procédure `CalculPlanSupportMin(C)` a une complexité en $\mathcal{O}(n^2)$ avec n le nombre de points de \mathcal{E} . La mise à jour de l'enveloppe convexe après ajout d'un point s'effectue en $\mathcal{O}(n)$. Comme les autres instructions de l'Algorithme 15 s'effectuent toutes en temps constant, on obtient une complexité en $\mathcal{O}(n^2)$ pour notre procédure incrémentale. Étant donné qu'il faut appliquer cette procédure incrémentale à chaque fois que l'on ajoute un point à \mathcal{E} , cela nous donne une complexité totale en $\mathcal{O}(n^3)$, dans le pire cas, pour un ensemble \mathcal{E} de n points. Cependant, en pratique, cette complexité ne semble pas atteinte. Le code de cet algorithme ainsi qu'un programme de test sont disponibles à l'adresse suivante : <http://www.loria.fr/~debled/BlurredPlane>

La notion de morceau de plan flou a été utilisée dans le cadre de l'extraction de paramètres géométriques d'une surface discrète pouvant être bruitée, ce travail est décrit dans la section suivante.

3.3 Extraction de paramètres géométriques

Nous notons \mathcal{O} , un objet discret 6-connexe. Le *bord* de \mathcal{O} est l'ensemble des points de \mathcal{O} qui ont un 6-voisin n'appartenant pas à \mathcal{O} , cet ensemble est noté \mathcal{B} . Dans cette section, le bord de l'objet discret est assimilé à sa surface.

3.3.1 Patch d'épaisseur ν centré en un point

La notion de *patch d'épaisseur ν centré en p* du bord d'un objet discret 3D présentée ci-dessous permet d'exploiter les informations géométriques des points situés au voisinage de p en considérant un voisinage adaptatif autour du point p en fonction de l'épaisseur ν fixée selon le bruit présent sur le bord de l'objet discret.

Définition 3.7 Soient \mathcal{B} le bord d'un objet discret et p un point de \mathcal{B} . Soient T un parcours donné des points de \mathcal{B} au voisinage de p et ν un réel représentant l'épaisseur autorisée. On appelle *patch d'épaisseur ν centré en p* , et on note $\Gamma_\nu(p)$, un morceau flou de plan discret d'épaisseur ν reconnu incrémentalement à partir du point p en ajoutant les points de \mathcal{B} selon le parcours T .

Dans les exemples présentés dans ce mémoire, un *patch d'épaisseur ν centré en p* est construit en utilisant l'algorithme incrémental de reconnaissance de morceaux flous de plans discrets présenté dans la section précédente. Les voxels sont ajoutés en suivant un parcours T et, dès que l'épaisseur du morceau flou de plan discret devient supérieure à ν , le processus de reconnaissance s'arrête.

Parcours T utilisé : Afin d'étendre le patch de manière uniforme dans toutes les directions, le mieux serait d'utiliser un parcours du voisinage de p selon une distance géodésique. Mais, pour des raisons d'efficacité, nous avons choisi d'effectuer un parcours en pondérant le voisinage d'un point à l'aide d'un masque de chanfrein $\langle 3, 4, 5 \rangle$, qui représente une bonne approximation de la distance géodésique [Bor96] et nous permet d'obtenir un patch équilibré autour de p avec une forme presque circulaire. Une file de priorité F est utilisée. Dans la phase d'initiation, p est inséré dans la file de priorité et un poids nul lui est associé. Puis, tant que l'épaisseur limite ν n'a pas

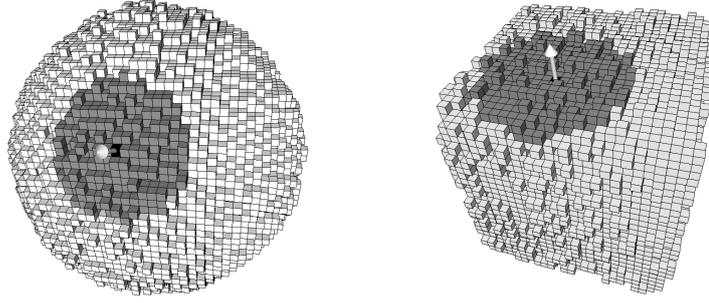


FIG. 3.4 – Exemple sur des objets bruités de deux patchs centrés sur les voxels noirs et d'épaisseur 2.

été dépassée, le voxel v de F , dont le poids w est le plus petit, est retiré de F et est ajouté au morceau flou de plan discret en cours de reconnaissance. Les 26-voisins de v sont ensuite ajoutés dans la file de priorité de la façon suivante :

- les 6-voisins sont ajoutés dans F avec un poids $w + 3$,
- les 18-voisins non 6-voisins sont ajoutés dans F avec un poids $w + 4$,
- et enfin les 26-voisins stricts sont ajoutés dans F avec un poids $w + 5$.

Afin de rester homéomorphe à un disque topologique, on s'assure également qu'un point n'est ajouté que s'il ne crée pas de trou dans le patch. On obtient ainsi des patchs comme ceux présentés à la figure 3.4.

Pour une épaisseur donnée, en chaque point p du bord d'un objet discret, le patch centré en p permet d'obtenir des informations sur la forme du bord de l'objet autour de p . Plus spécifiquement, grâce aux informations données par les patchs des points du bord d'un objet discret pour une épaisseur donnée, nous en avons déduit les estimateurs des paramètres géométriques décrits dans les paragraphes suivants.

3.3.2 Normale d'épaisseur ν en un point

Avec la notion de patch définie précédemment, nous avons défini un nouvel estimateur de la normale en tout point de la surface d'un objet discret.

Définition 3.8 Soient \mathcal{B} le bord d'un objet pouvant être bruité et p un point de \mathcal{B} . On appelle *normale d'ordre ν au point p* le vecteur

$$\vec{n}_\nu(p) = \vec{n}(\Gamma_\nu(p))$$

où $\vec{n}(\Gamma_\nu(p))$ représente le vecteur normal du patch $\Gamma_\nu(p)$.

Des résultats obtenus sur différentes sphères, bruitées et non bruitées (cf. Figure 3.5), de différents rayons sont présentés dans le tableau Tab. 3.1 ainsi qu'à la figure 3.6. Les sphères bruitées ont été synthétisées de la façon suivante : une sphère discrète (i.e. l'ensemble des points $\{(x, y, z) \in \mathbb{Z}^3 \mid x^2 + y^2 + z^2 < r(r+1)\}$ où r représente le rayon de la sphère) a été générée, puis du bruit a été ajouté en déplaçant aléatoirement les voxels du bord de la sphère vers l'extérieur de celle-ci. Pour obtenir un objet plus bruité, la dernière étape est itérée.

Nous avons comparé la normale estimée avec la normale théorique (obtenue en un point p en prenant la normale à la sphère continue qui passe par p). Nous présentons dans Tab 3.1 la

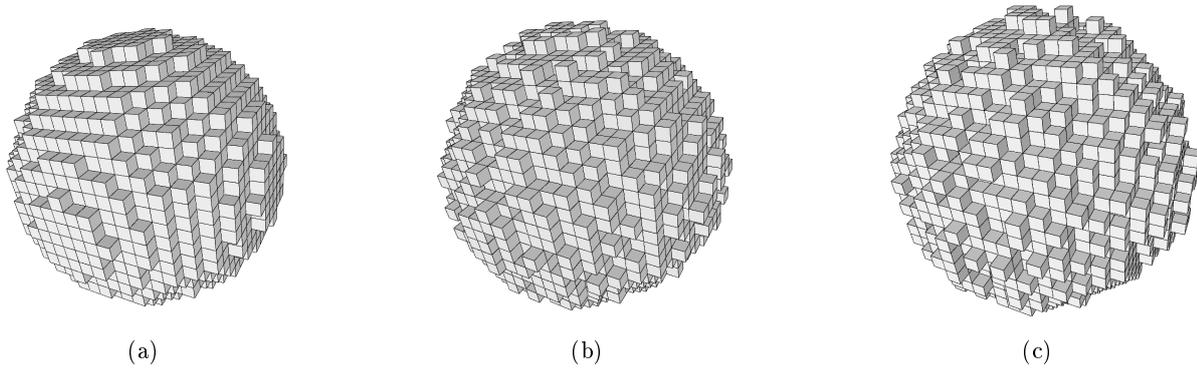


FIG. 3.5 – Une sphère de rayon 10 (a) non bruitée, (b) faiblement bruitée et (c) fortement bruitée.

Sphère testée			Non bruitée			Faiblement bruitée			Fortement bruitée		
Rayon			10	20	40	10	20	40	10	20	40
Ep.	$\nu = 2$	δ_{moy}	2.33	1.25	0.77	3.66	2.39	1.46	5.65	4.57	4.35
		δ_{max}	9.92	3.35	2.94	16.86	10.16	5.83	46.98	40.54	52.57
		σ	1.49	0.61	0.44	2.19	1.35	0.78	3.77	3.33	3.97
	$\nu = 3$	δ_{moy}	1.85	1.17	0.70	2.75	1.74	1.02	3.29	2.37	1.70
		δ_{max}	6.49	3.93	2.58	8.90	7.46	3.89	14.63	8.80	7.32
		σ	0.98	0.69	0.40	1.54	0.92	0.55	1.90	1.30	0.93
26-voisinage	δ_{moy}	4.61	4.50	4.22	7.55	7.50	7.51	8.83	8.69	8.57	
	δ_{max}	9.45	15.34	17.19	28.35	25.34	31.14	24.29	37.15	33.80	
	σ	2.70	2.75	2.81	4.19	4.10	4.19	4.52	4.87	4.80	

TAB. 3.1 – Comparaison entre l'estimateur de vecteur normal proposé et l'estimateur utilisant uniquement l'information des surfels dans un 26-voisinage du point considéré. Les différences d'angles δ sont exprimées en degrés $^\circ$.

moyenne des différences angulaires, exprimée en degrés, entre les normales estimées et théoriques (δ_{moy}), la différence angulaire maximale (δ_{max}), et l'écart type (σ).

Pour $\nu = 1$ et un objet même légèrement bruité, l'estimation obtenue n'est pas satisfaisante car les patchs obtenus sont très petits et ils ne couvrent pas le 26-voisinage d'un point p . On peut noter que, pour toutes les sphères, l'estimation de la normale avec cette méthode devient meilleure au fur et à mesure que le rayon de la sphère augmente, contrairement à la méthode de référence qui repose sur le 26-voisinage où l'on ne note pas de réels changements. Ceci est dû au fait que notre méthode est adaptative et considère un voisinage plus grand lorsque le rayon de la sphère augmente. D'autre part, si on augmente le bruit, on remarque une dégradation de l'estimation du vecteur normal et l'épaisseur du patch doit donc être augmentée.

3.3.3 Aire d'un patch d'épaisseur ν

Etant donné une surface euclidienne \mathcal{S} et son champ de vecteurs normaux $\{\vec{n}\}$, dans l'espace continu on obtient l'aire de \mathcal{S} avec la formule :

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

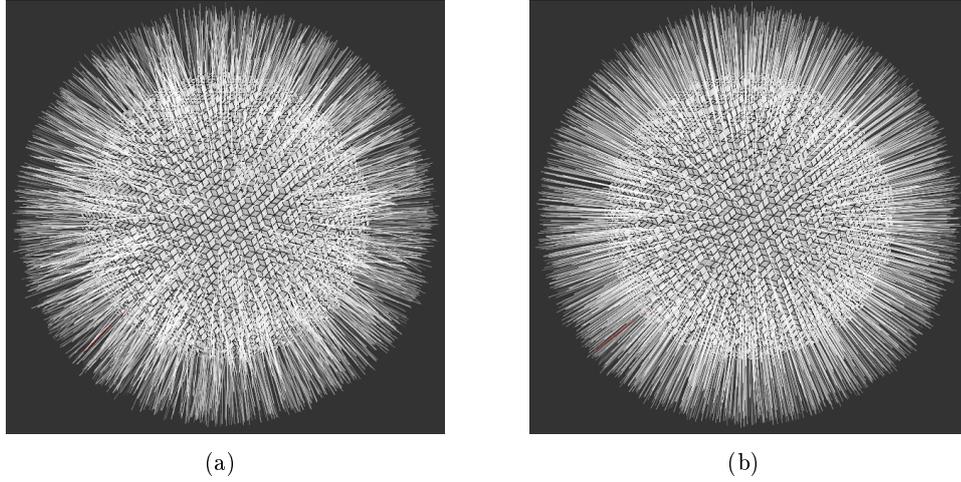


FIG. 3.6 – Estimation des normales sur une sphère faiblement bruitée de rayon 20 (a) en utilisant les informations du 26-voisinage et (b) avec l'estimateur reposant sur les patches d'épaisseur 3.

Si une discrétisation $D(\mathcal{S})$ de \mathcal{S} est maintenant considérée, on peut remplacer l'intégrale sur \mathcal{S} par une somme finie sur les surfels s de $D(\mathcal{S})$, le vecteur $\vec{n}(s)$ par une estimation de celui-ci et ds par un produit scalaire avec le vecteur unitaire orthogonal au surfel s et pointant vers l'extérieur de l'objet. L'idée est de calculer la contribution de chaque surfel à l'aire globale de $D(\mathcal{S})$ en le projetant relativement à sa normale estimée. On obtient alors l'estimateur discret suivant :

$$E_{\mathcal{A}}(D(\mathcal{S})) = \sum_{s \in D(\mathcal{S})} \vec{n}^*(s) \cdot \vec{n}_{el}(s)$$

où $\vec{n}^*(s)$ est une estimation du vecteur normal à un élément s de $D(\mathcal{S})$ et $\vec{n}_{el}(s)$ correspond au vecteur normal élémentaire en s .

Cette méthode a d'abord été proposée par A. Lenoir dans [LMR96]. Puis, dans [CFTT03], Coeurjolly et al. ont prouvé que cet estimateur d'aire converge si et seulement si l'estimateur de normale converge.

Nous avons ici utilisé l'estimateur de normale proposé dans la section 3.3.2 dans la formule précédente afin d'obtenir un estimateur d'aire pour la surface d'un patch d'épaisseur ν :

Proposition 3.1 *L'aire $E_{\mathcal{A}}$ d'un patch $\Gamma_{\nu}(p)$ d'épaisseur ν centré en p , point du bord d'un objet discret est estimé par*

$$E_{\mathcal{A}}(\Gamma_{\nu}(p)) = \sum_{s \in \mathcal{S}_{\Gamma_{\nu}(p)}} \vec{n}_{\nu}(p) \cdot \vec{n}_{el}(s) = \vec{n}_{\nu}(p) \cdot \sum_{s \in \mathcal{S}_{\Gamma_{\nu}(p)}} \vec{n}_{el}(s)$$

avec $\mathcal{S}_{\Gamma_{\nu}(p)}$ l'ensemble des surfels des voxels de $\Gamma_{\nu}(p)$ qui sont incidents à un voxel n'appartenant pas à l'objet discret.

3.3.4 Estimateur de forme

Les notions définies dans les sections précédentes permettent de définir un estimateur pouvant être utilisé pour caractériser la forme locale (dans le sens concave/convexe) du bord \mathcal{B} d'un

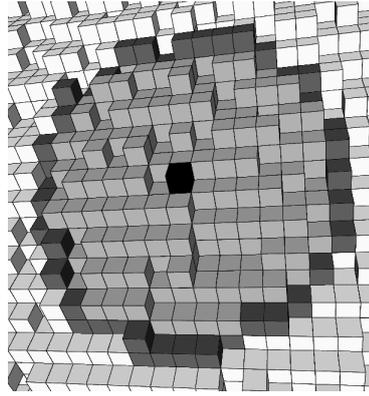


FIG. 3.7 – Contour (en gris foncé) du patch gris d'épaisseur 2 centré au point en noir.

objet discret autour d'un point p de \mathcal{B} .

On considère un patch $\Gamma_\nu(p)$ centré en p , un point du bord d'un objet discret et C la suite des points de son *contour*. Un point q se situe sur le *contour* d'un patch $\Gamma_\nu(p)$ si la représentation voxel de q contient un surfel $s \in \mathcal{S}_{\Gamma_\nu(p)}$ et s'il existe un surfel $s' \in \mathcal{S} \setminus \mathcal{S}_{\Gamma_\nu(p)}$ ¹⁵ tels que s et s' soient adjacents par arête (cf. Figure 3.7).

L'idée retenue pour cet estimateur est que les patches du contour de $\Gamma_\nu(p)$ donnent des indications sur la forme autour du point p au-delà de la zone *plate* du patch. De plus, pour chaque point $q_i, 1 \leq i \leq |C|$ du contour C du patch $\Gamma_\nu(p)$, le patch $\Gamma_\nu(q_i)$ centré en q_i est construit. Le champ des vecteurs normaux $\vec{n}_\nu(q_i)$ des patches des points du contour de $\Gamma_\nu(p)$ en est alors déduit, ceci est illustré sur la figure 3.8 pour une zone concave et une zone convexe. Nous observons que dans les zones concaves les vecteurs $\vec{n}_\nu(q_i)$ pointent vers le vecteur $\vec{n}_\nu(p)$, alors que pour les zones convexes, ils ont la tendance inverse. Il est donc possible de discriminer les zones concaves des zones convexes grâce à la valeur de l'angle orienté entre les vecteurs normaux des points du contour et le vecteur normal du patch central. De plus, selon la taille de patches du contour par rapport au patch en p , cet angle doit être plus ou moins considéré. Ces observations nous ont conduits à l'estimateur défini ci-dessous.

Soit C l'ensemble des points du contour du patch $\Gamma_\nu(p)$. L'estimateur de la *forme* de la surface autour du point p est donné par la formule :

$$\mathcal{F}_\nu(p) = \frac{1}{|C|} \sum_{\forall q \in C} (\widehat{\vec{n}_\nu(p), \vec{n}_\nu(q)}) \cdot \frac{E_{\mathcal{A}}(\Gamma_\nu(q))}{E_{\mathcal{A}}(\Gamma_\nu(p))}$$

où $(\widehat{\vec{n}_\nu(p), \vec{n}_\nu(q)})$ représente la valeur de l'angle orienté entre les deux vecteurs normaux. L'estimateur $\mathcal{F}_\nu(p)$ est donc une moyenne pondérée des différences d'angles entre $\vec{n}_\nu(p)$ et les $\vec{n}_\nu(q_i)_{1 \leq i \leq |C|}$. Le signe de l'angle $(\widehat{\vec{n}_\nu(p), \vec{n}_\nu(q)})$ est défini selon la figure 3.9.

¹⁵ \mathcal{S} est l'ensemble des surfels des voxels de \mathcal{O} qui sont incidents à un voxel n'appartenant pas à l'objet discret.

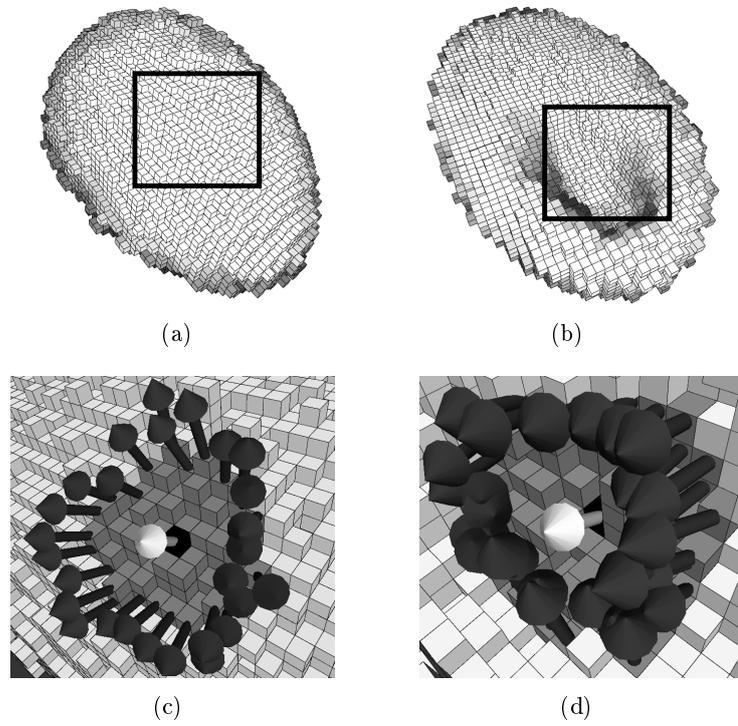


FIG. 3.8 – Organisation des vecteurs sur le contour d'un patch (a) d'une zone convexe et (b) d'une zone concave.

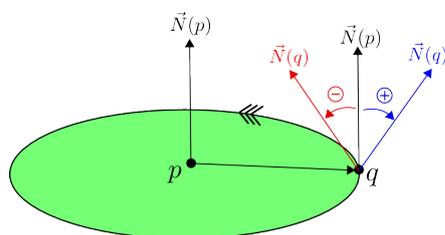


FIG. 3.9 – Orientation des angles

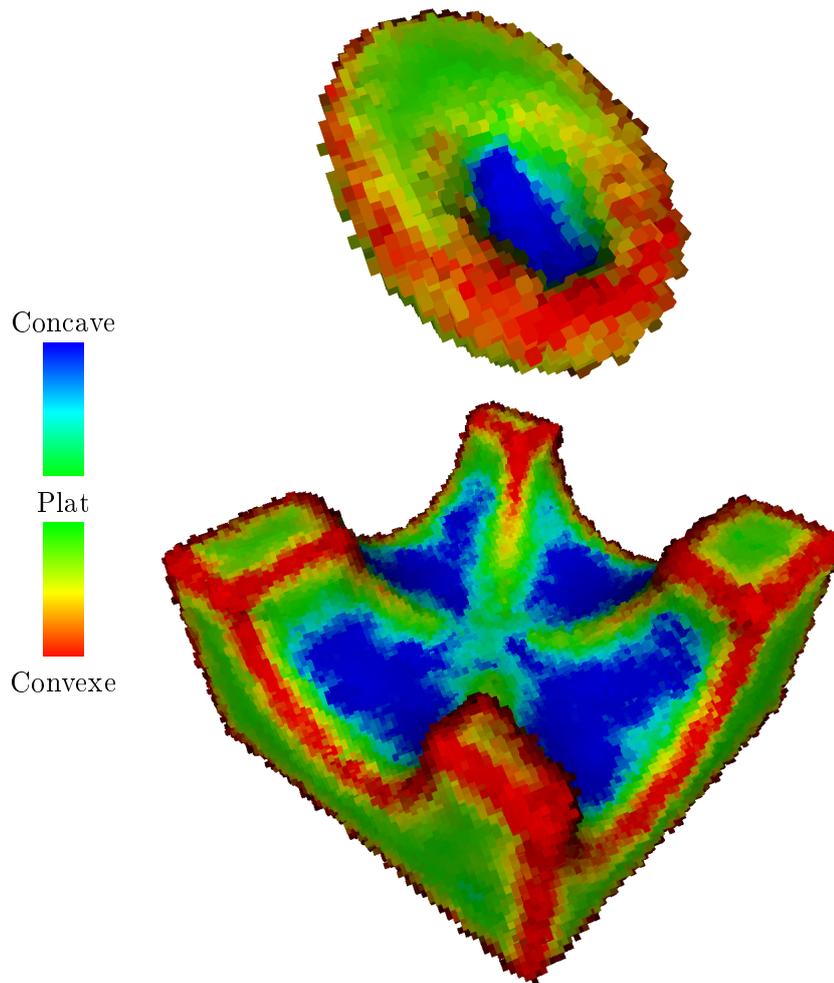


FIG. 3.10 – Estimation des parties concaves (bleu), convexes (orange), plates (vertes) en utilisant les patches d'épaisseur 3 dans le calcul de $\mathcal{F}_\nu(p)$.

Interprétation de $\mathcal{F}_\nu(p)$

Si $\mathcal{F}_\nu(p)$ est positif, la surface autour de p est *plutôt* convexe et si $\mathcal{F}_\nu(p)$ est négatif, la surface autour de p est *plutôt* concave. Une valeur $|\mathcal{F}(p)|$ qui augmente traduit un caractère concave ou convexe plus prononcé de la surface autour de p . Si $\Gamma_\nu(p)$ est grand, une valeur $\mathcal{F}_\nu(p)$ proche de 0 signifie que la zone autour de p est presque plate (relativement à l'épaisseur ν fixée). Si $\Gamma_\nu(p)$ est petit, alors la zone autour de p est fortement déformée, mais d'une manière qu'on ne peut ni qualifier de concave, ni de convexe (point selle par exemple).

Quelques résultats de cet estimateur sont présentés sur la figure 3.10. Pour le rendu de l'objet, à chaque face d'un voxel v , sont associés le vecteur normal $\vec{n}_\nu(v)$ et une couleur représentant le caractère concave/convexe de la surface autour du point v .

Perspectives d'application en bioinformatique

Dans le cadre du thème *Modélisation des Biomolécules et de leurs Interactions* du Contrat de Plan Etat Région signé en 2007, nous avons initié un travail en collaboration avec Bernard

Maigret et Matthieu Chavent (équipe ORPAILLEUR, Loria) dont l'objectif est d'étudier le **docking de protéines**, c'est-à-dire les différentes possibilités d'interaction entre les surfaces de deux protéines. Pour cela nous cherchons à caractériser la complémentarité des représentations volumiques des partenaires en découpant les surfaces moléculaires en régions caractéristiques.

Notre objectif est, d'une part, d'obtenir des caractéristiques géométriques suffisamment précises pour émettre des hypothèses d'assemblage entre 2 protéines et, d'autre part, d'adopter une approche *multirésolution* permettant de tenir compte du bruit éventuel présent à la surface des protéines et donnant ainsi la possibilité d'affiner localement le processus de détection de complémentarité. Nous utiliserons pour cela les informations obtenues avec les patches centrés en chaque point de la surface discrète de chaque protéine. De manière plus générale, nous cherchons à améliorer les méthodes de recherche d'informations géométriques sur la surface des protéines proposées dans [DNW02].

3.4 Segmentation en morceaux flous de plans discrets

La segmentation du bord d'un objet discret proposée utilise, pour une épaisseur ν donnée :

- l'algorithme de reconnaissance de morceaux flous de plans discrets présenté à la section 3.2,
- les estimateurs de normale, d'aire et de forme d'un objet discret présentés dans les paragraphes précédents.

La méthode employée repose sur un pré-traitement qui calcule le patch d'épaisseur ν en chacun des points du bord \mathcal{B} de l'objet discret \mathcal{O} à segmenter. A chacun de ces points p est ainsi associé :

- la normale en p : $n_\nu(p)$,
- l'aire estimée de $\Gamma_\nu(p)$: $E_{\mathcal{A}}(\Gamma_\nu(p))$,
- la valeur $\mathcal{F}_\nu(p)$.

Plusieurs stratégies de segmentation ont été testées que je résume ci-dessous.

Stratégie d'extension maximale

La segmentation est amorcée en commençant le premier morceau flou de plan discret d'épaisseur ν (noté MFP_ν par la suite) par un point g de \mathcal{B} , appelé **graine**, pour lequel la valeur de l'aire de $\Gamma_\nu(g)$ est la plus grande. Puis les points q , 26-voisins de g , sont ajoutés dans la liste des points à parcourir pour ce morceau en fonction de leur valeur $E_{\mathcal{A}}(\Gamma_\nu(q))$. Chaque point de cette liste est testé avec l'algorithme de reconnaissance en commençant par le point de la liste possédant la plus grande valeur d'aire. Si ce point peut être ajouté au MFP_ν courant, il est retiré de la liste et ses 26-voisins y sont ajoutés et ainsi de suite ...

Quand tous les points susceptibles d'être ajoutés pour ce premier MFP_ν ont été testés, un autre morceau est amorcé avec un point g' de \mathcal{B} , non encore parcouru, pour lequel l'aire de $\Gamma_\nu(g')$ est la plus grande des points non encore visités. Ce processus est itéré jusqu'à ce que tous les points de \mathcal{B} soient intégrés dans un MFP_ν .

Pendant ce processus, un contrôle est effectué vérifiant que le MFP_ν en cours de construction est bien homéomorphe à un disque. Des résultats de cette méthode de segmentation sont donnés

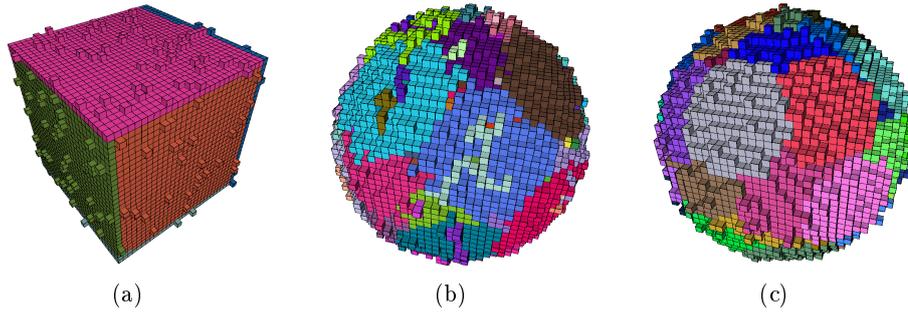


FIG. 3.11 – Segmentation d'épaisseur 2 (a) d'un cube faiblement bruité d'arête 30 ; (b) d'une sphère faiblement bruitée de rayon 20 avec la stratégie d'extension maximale ; (c) de la même sphère avec la stratégie d'extension basée sur les patches d'épaisseur 2.

aux figures 3.11(a) et 3.11(b). Le cube de la figure 3.11(a) est segmenté de manière satisfaisante en 6 morceaux correspondant aux 6 faces du cube. Par contre, la segmentation de la sphère à la figure 3.11(b) pourrait être améliorée car elle ne tient pas compte de sa forme courbée. Comme le processus utilisé ne stoppe pas dès qu'un point n'appartient pas au MFP_ν mais teste alors un autre point de la liste, les morceaux obtenus s'étendent et peuvent devenir *filaires*.

Afin d'éviter ce type d'extension, nous avons testé une **segmentation basée sur les patches d'épaisseur ν** qui utilise le même principe de croissance que celui décrit dans le paragraphe 3.3.1. Le résultat obtenu sur la sphère est plus satisfaisant (voir Figure 3.11(c)). Cela est dû au processus de croissance géodésique utilisé.

Stratégie hybride

A partir des observations précédentes, nous avons développé une méthode de segmentation hybride. A chaque nouveau MFP_ν , un point graine g est choisi selon la valeur de l'aire de son patch. Pour les graines appartenant à une zone *plate* de l'objet, la stratégie d'extension maximale est utilisée pour construire le MFP_ν et, pour les graines appartenant à une zone *courbée*, la stratégie d'extension basée sur les patches d'épaisseur ν est adoptée.

La distinction entre les zones *plates* et *courbées* est réalisée en utilisant la valeur en g de l'estimateur de forme $\mathcal{F}_\nu(g)$. Un paramètre de seuil σ doit donc être choisi et si $|\mathcal{F}_\nu(g)| < \sigma$, la stratégie d'extension maximale est utilisée sinon c'est la stratégie d'extension basée sur les patches d'épaisseur ν .

Des résultats obtenus avec cette méthode sont montrés à la figure 3.12. Des objets synthétiques, de différentes formes sont segmentés pour des épaisseurs fixées. Nous pouvons constater que la segmentation que nous proposons donne aussi de bons résultats sur des objets non bruités (cf. Figures 3.12(g) et 3.12(h)). La stratégie hybride permet une bonne segmentation des objets possédant à la fois des zones plates et des zones courbées (cf. Figures 3.12(c), 3.12(f) et 3.12(i)).

Un travail en cours concerne la détection automatique du seuil à fixer dans le cadre de la stratégie hybride. De plus, un travail de réflexion est à mener sur la forme des MFP_ν obtenus pendant la segmentation, que devons-nous obtenir ? Un nombre de MFP_ν minimum ? Des MFP_ν de même taille et de même forme ? Souhaitons-nous un respect des convexités et concavités de l'objet ?

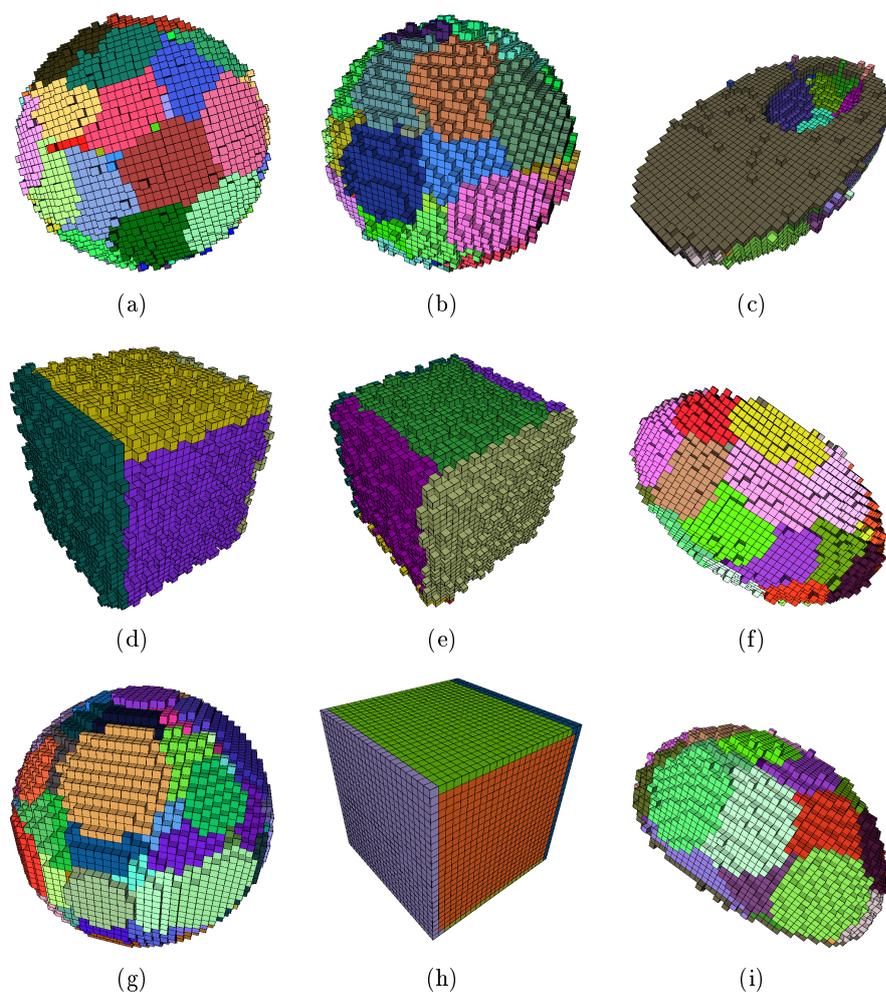


FIG. 3.12 – Résultats de la segmentation par la méthode hybride d’objets synthétiques : (a, b) segmentation d’épaisseur 2 d’une sphère de rayon 20 faiblement bruitée ; (d, e) segmentation d’épaisseur 3 d’un cube d’arête 25 fortement bruité ; (g, h) segmentation d’épaisseur 1 d’objets non bruités ; et (c, f, i) segmentation d’épaisseur 2 d’un demi-ellipsoïde.

Un de nos objectifs est aussi d’étudier *la polyédricisation d’objets discrets bruités* en contrôlant l’approximation effectuée ainsi que la taille et la forme des facettes obtenues. Du travail reste encore à réaliser en particulier au sujet du bord des facettes obtenues en utilisant par exemple les stratégies proposées dans [Bur02, Siv04, Dex06, DCA06].

3.5 Conclusion

Ce chapitre expose les premiers travaux effectués dans le cadre de l’étude des objets discrets bruités tridimensionnels. La notion de morceau flou de plan discret d’épaisseur ν a été introduite avec deux algorithmes de reconnaissance. Dans un souci d’efficacité, je souhaite tester et proposer d’autres algorithmes, par exemple adapter l’algorithme présenté au chapitre 1 section 1.4 (page 46) à la reconnaissance de morceaux flous de plans discrets pour une épaisseur donnée. A partir

de cette définition et des algorithmes de reconnaissance présentés, la notion de patch d'épaisseur ν a été proposée, à partir de laquelle des estimateurs de normale, d'aire et de forme ont été définis.

Afin d'accélérer les calculs effectués, le problème de la fusion de deux patches pour une épaisseur donnée doit être étudié. De plus, la piste suivante va être rapidement explorée : l'algorithme de reconnaissance des segments flous 2D et la tangente associée (cf. chapitre 2 section 2.2.2) peut être utilisé avec la même méthode que celle présentée au chapitre 1 section 1.3 (page 38) pour le calcul des normales d'un objet discret en considérant les coupes de l'objet tridimensionnel en chacun des points du bord et en calculant les normales dans chacune des coupes passant par le point considéré. Selon la même stratégie et en utilisant le calcul de courbure d'épaisseur ν de courbes discrètes 2D (cf. chapitre 2 section 2.2.2), un estimateur de la courbure en chaque point du bord de l'objet discret peut en être déduit selon la technique proposée par A. Lenoir (cf. [Len97]).

L'étude de la polyédrisation d'objets discrets bruités en contrôlant les approximations effectuées et en respectant la forme de l'objet est une voie de recherche à explorer qui pourra aussi nous conduire au lissage des objets discrets bruités.

Chapitre 4

Applications en bioinformatique

4.1 Introduction

A la création de l'équipe ADAGE (Algorithmique Discrète et ses Applications à la Génomique), le champ d'application privilégié de l'équipe étant la génomique, je me suis intéressée à ce sujet de recherche. Plus spécifiquement, j'ai étudié les interactions de la géométrie dans la biologie moléculaire, c'est-à-dire l'étude des propriétés géométriques des macromolécules biologiques (ADN, ARN, protéines). J'ai encadré plusieurs stages à différents niveaux sur des problèmes de bioinformatique (cf. chapitre 3 de la partie I).

Dans la section 4.2, je présente un travail sur le calcul de la courbure de l'ADN [39] qui utilise des résultats de géométrie discrète sur la courbure des courbes discrètes 3D bruitées et déconnectées (cf. chapitre 2 section 2.3.6). Ce travail a été initié avec Franck Rapaport [Rap04] qui fut en stage de DEA informatique dans notre équipe en 2004. Puis, il a été repris par des stagiaires de master bioinformatique afin d'exploiter des résultats biologiques.

Dans la section 4.3, un travail sur la recherche de motifs dans des séquences d'ADN est relaté. Il a fait l'objet de plusieurs co-encadrements de stages, et une thèse sera bientôt soutenue sur ce sujet. Ces travaux [25,31,32,44] se situent dans le champ de l'algorithmique discrète mais ne concernent pas la géométrie discrète, ils seront donc très brièvement décrits.

4.2 Courbure de l'ADN

Dans le cadre du thème de recherche de notre équipe sur la régulation des gènes (cf. 4.3 où des détails seront donnés sur le principe de la régulation des gènes), nous étudions la courbure de l'ADN [BMH91]. En effet, pour certains gènes, elle constitue un facteur important dans de nombreux processus biologiques : transcription, initiation, réplication de l'ADN, etc. L'objectif général de ce travail est d'étudier le rôle de la courbure de l'ADN dans la régulation des gènes.

La conformation spatiale de la molécule d'ADN est un facteur primordial pour la génétique. En effet, la forme de la macromolécule peut nous aider à déterminer les sites de certaines réactions chimiques (par exemple la fixation de protéines spécifiques) en nous permettant d'extraire les sites facilitant ces réactions. Il devient alors plus facile d'identifier le rôle de ces sites. Du point de vue chimique, l'ADN est une longue molécule formée par un enchaînement de blocs appelés nucléotides et composés d'un sucre (la désoxyribose), d'un groupement phosphate et d'une base nitrogenée parmi quatre bases possibles (l'adénine, la thymine, la guanine ou la cytosine). Les

groupements phosphate et sucre de deux nucléotides successifs se lient, formant un long polymère. Depuis 1953, il est connu que les brins sont complémentaires, c'est-à-dire qu'à chaque base sur un brin correspond une base analogue sur l'autre (l'adénine correspond à la thymine, la guanine à la cytosine), comme le montre la figure 4.1.

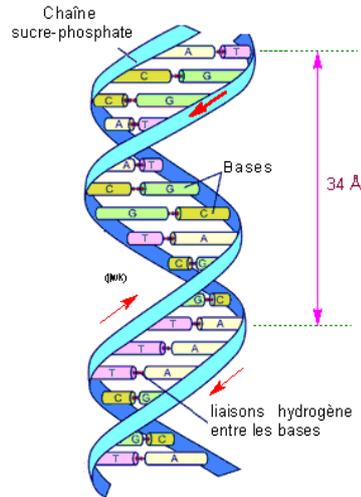


FIG. 4.1 – Une séquence d'ADN.

La forme de l'ADN est celle d'une échelle tordue en double hélice, les liaisons hydrogène entre les paires de nucléotides analogues formant les barreaux de l'échelle. Les deux brins étant complémentaires, il suffit de connaître la séquence de bases d'un seul d'entre eux pour connaître la composition chimique de la molécule. On peut alors représenter le code génétique par une suite de lettres caractérisant les nucléotides ; leurs initiales A, C, G, T.

Le polymorphisme structurel implique que selon le milieu dans lequel une séquence d'ADN évolue, une même séquence existe sous trois formes, A, B et Z qui diffèrent par l'espacement des nucléotides et la forme générale de la double hélice. La forme B étant de loin la plus répandue, nous ne nous intéressons qu'à celle-ci.

Un modèle simplifié de la structure 3D de l'ADN a été proposé, appelé *modèle tubulaire* [BMH91], qui ne s'intéresse qu'à la forme générale de l'ADN, c'est-à-dire à la courbe formée par les milieux des paires de nucléotides (cf. Figure 4.2). Il suffit alors de suivre l'axe du modèle tubulaire pour obtenir les déformations locales de la structure et donc la forme de la molécule.

Dans la littérature, il existe différents modèles de conformation 3D des séquences d'ADN. Ils sont construits à partir de constituants di- ou tri-nucléotidiques (par exemple [BMH91, FZL97, DH98, Dic98]). Dans [GB99], une comparaison a été effectuée entre certains modèles et il a été montré que les représentations obtenues sont le plus souvent comparables. Pour cette étude, nous avons choisi le modèle de A. Bolshoy et E.N. Trifonov [BMH91] qui est établi sur la base d'expériences biologiques sur gel de séquences d'ADN. Ce modèle est basé sur l'étude du déplacement entre deux nucléotides et il propose un algorithme, associé à un tableau de valeurs d'angles et une valeur de translation, pour construire la représentation 3D.

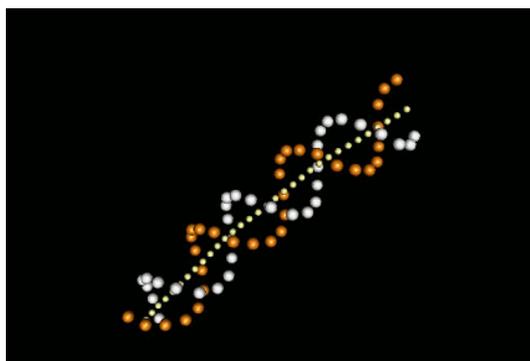


FIG. 4.2 – Représentation tubulaire d'une séquence d'ADN.

Avec A.S. Shpigelman et E.N. Trifonov, A. Bolshoy a ensuite écrit un programme nommé **CURVATURE** [STB93] qui, à partir d'une séquence donnée en paramètre, fournit les coordonnées de chacune des extrémités des segments dinucléotidiques et donc la forme du modèle tubulaire.

Ce programme propose aussi une estimation du rayon de courbure en chaque point du modèle tubulaire. Le principe de ce calcul est le suivant : un paramètre entier impair l correspondant à la largeur d'une fenêtre *glissant* sur la courbe est fixé, puis cette fenêtre est centrée en chaque point p de la courbe. Le point de la courbe p_G situé sur l'extrémité gauche de la fenêtre ainsi que celui situé sur l'extrémité droite, p_D , sont considérés et le rayon de courbure au point p est obtenu en calculant le rayon du cercle circonscrit aux trois points p_G, p, p_D . Il est clair que, selon le paramètre de largeur de la fenêtre choisi, les résultats seront très différents. De plus, dans les études biologiques menées avec ce logiciel, l'auteur prend des paramètres de l'ordre de cent. Il va donc négliger les petites variations de la courbe. Il a ainsi réussi à montrer la forte courbure de l'ADN dans certaines séquences promotrices, particulièrement celles de la bactérie *E. Coli* [GLB99]. Plusieurs autres logiciels de modélisation de la courbure de l'ADN existent (par exemple **DIAMOD**, **BEND**) et utilisent des approches diverses [STB93, DH98] mais, pour la plupart, le calcul de la courbure dépend d'un paramètre, fixé par l'utilisateur, correspondant à la largeur d'une fenêtre *glissant* le long de la séquence étudiée.

Le modèle tubulaire fourni par **CURVATURE** est un modèle qui a été utilisé par plusieurs biologistes pour différents travaux. C'est aussi le modèle 3D que nous adoptons pour proposer un nouvel estimateur de la courbure de l'ADN.

A l'occasion d'un stage de DEA, en 2004, avec Franck Rapaport, nous avons proposé un nouvel algorithme pour calculer la courbure en chaque point d'une courbe discrète 3D bruitée sans utiliser l'approche "fenêtre glissante" [Rap04]. Cet algorithme repose sur les résultats présentés dans le chapitre 2 section 2.3.6, page 95. L'algorithme de calcul de courbure présenté dans cette section est adapté pour les nombres réels et est utilisé sur la courbe 3D déconnectée formée de points réels correspondant à la courbe de l'axe de la double hélice de la modélisation 3D d'une séquence d'ADN. Cette modélisation est obtenue avec l'algorithme proposé dans [STB93]. Un logiciel prototype nommé **CurvaDis** a été réalisé [Ngu05]. Après plusieurs expérimentations, l'épaisseur choisie pour le calcul de la courbure de l'ADN est 2.

Une première expérimentation a été réalisée en étudiant le phénomène de la régulation avec

les protéines *H-NS/FIS* des opérons ¹⁶ *rrn* issus de protéobactéries. C'est un phénomène complexe et la fixation de ces deux protéines nécessite deux zones de forte courbure localisées dans un intervalle de 100 à 90 nucléotides en amont du début de la transcription. Sur les 60 séquences testées, le logiciel *CurvaDis* a permis une bonne localisation des deux pics de courbure alors que le logiciel *CURVATURE* ne détectait qu'une seule zone, trop large pour distinguer les 2 pics.

Les premiers résultats obtenus sont encourageants, cependant, l'aide de biologistes est indispensable pour valider cette approche et proposer des lots de séquences à tester. De plus, d'autres modèles tridimensionnels de l'ADN doivent être comparés pour le calcul de la courbure. Je souhaite aussi étudier le rôle de la torsion dans la représentation 3D de l'ADN.

4.3 Identification et classification des promoteurs

Le travail présenté dans cette section se situe en dehors du cadre de la géométrie discrète.

L'expression du génome est un processus biologique extrêmement complexe. Elle se passe à plusieurs stades – d'abord au stade de la *transcription* de la molécule d'ADN, ensuite au stade de la *maturation* et de l'*épissage* de l'ARN messager (chez les organismes supérieurs), et enfin au stade de la *traduction* de l'ARN messager en protéine. À chacun de ces stades (cf. Figure 4.3), un complexe protéique entre en interaction avec la molécule nucléotidique pour initier et piloter l'étape correspondante d'expression. La biologie de cette interaction entre la protéine et l'ARN/ADN, appelée *régulation*, n'est pas encore bien comprise. Cependant, la présence d'un *signal* dans la molécule d'ARN/ADN reconnu par la protéine est une condition nécessaire (mais pas suffisante) pour que cette interaction ait lieu. En terme de séquence, il s'agit donc d'un *motif* ¹⁷ qui code ce signal. Identifier et décrire ces motifs est un problème d'une très grande importance pour les biologistes, mais il est aussi d'une difficulté notoire. L'approche courante à ce problème consiste à se donner un échantillon de séquences censées contenir le signal en question, et ensuite chercher un motif significatif commun à ces séquences. Plusieurs méthodes et logiciels existent pour cette tâche. Certains sont comparés dans un article récent [TA105] qui montre la diversité et le nombre de solutions proposées pour la résolution de ce problème fondamental en bioinformatique. Cependant la plupart de ces logiciels (par exemple *MEME* [BE94]) ne peut pas utiliser d'espacement variable entre deux mots (forme majoritairement prise par les motifs recherchés) ou fait appel à des méthodes pour lesquelles le bruit de fond peut interférer avec le motif biologique (*BioProspector* [LBL01]).

En résumé, ce problème comporte des aspects à la fois biologiques et informatiques qui interagissent étroitement. Du côté biologique, il est capital de mieux comprendre le mécanisme d'interaction protéine-ADN/ARN pour spécifier la forme de motifs que l'on recherche. Du côté informatique, de nouvelles méthodes doivent être mises en place qui, compte tenu de ces réalités biologiques, pourront identifier des motifs communs de façon souple et efficace.

Nous travaillons sur ce thème en collaboration avec des chercheurs du Laboratoire de Génétique et de Microbiologie de l'Université Henri Poincaré de Nancy (P. Leblond, B. Aigle) depuis 2001, dans le cadre du thème *Bioinformatique* du PRST (Pôle de Recherche Scientifique et

¹⁶Opéron : groupe de gènes de même orientation dont la transcription globale dépend d'un promoteur situé en amont du premier gène

¹⁷Motif : site potentiel de fixation de facteur de transcription pouvant être composé de deux mots ou boîtes, dites -35 et -10 ce qui correspond aux positions de ces mots dans la zone située avant la transcription, appelée zone promotrice.

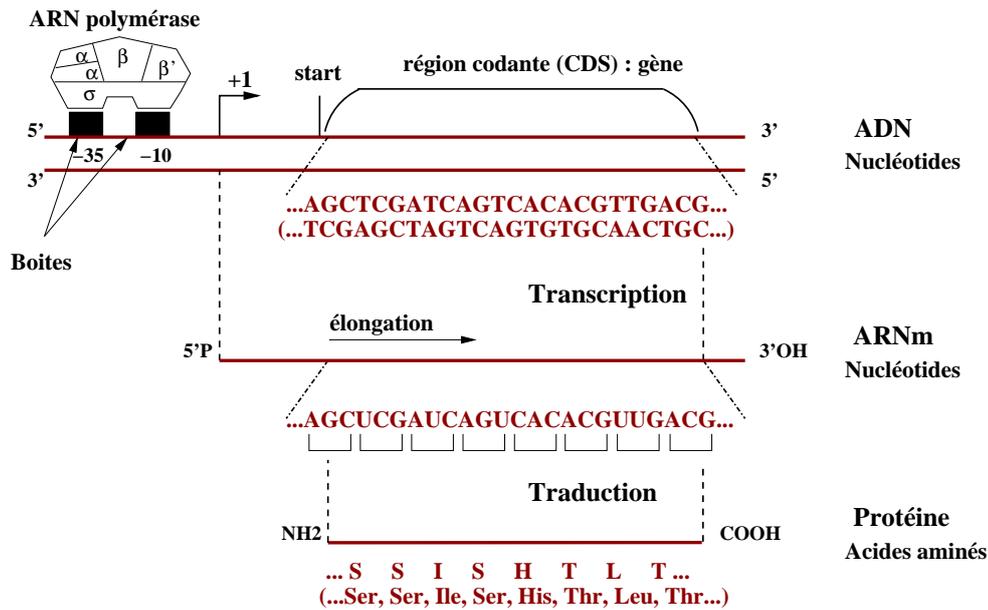


FIG. 4.3 – Transcription et traduction d'un gène codant une protéine.

Technologique) *Intelligence Logicielle* du CPER (Contrat de Plan Etat Région), de 2001 à 2006. Plusieurs étudiants y ont contribué (voir le chapitre 3 de la partie I), je participe à l'encadrement de la thèse de Fabrice Touzain qui se terminera fin 2007. Ce travail a pour objectif de décrire les sites de fixation dans les zones promotrices¹⁸ du génome de la bactérie *Streptomyces caelicolor*, dont la famille produit approximativement 70% des antibiotiques connus. Nous recherchons des motifs composés de deux *mots* (ou *boîtes*) avec un espacement variable.

Après une étude de la *sur-représentation* statistique supposée des sites de fixation des facteurs de transcription (utilisation du programme R'MES¹⁹ [Sch97]) afin de les identifier [30], nous avons utilisé une *approche comparative*, c'est-à-dire fondée sur la comparaison de génomes de plusieurs bactéries relativement proches sur le plan phylogénétique. Cette approche repose sur l'hypothèse que des protéines de deux bactéries issues d'une bactérie ancestrale doivent avoir gardé les mêmes mécanismes (et donc les mêmes sites) de régulation. Des résultats intéressants ont été obtenus, notamment avec deux espèces (*Streptomyces caelicolor* et *Streptomyces avermitilis*) et trois espèces (*Streptomyces caelicolor*, *Streptomyces avermitilis* et *Mycobacterium tuberculosis*) [31]. Néanmoins, l'approche, basée sur des modèles de Markov, s'est avérée inadaptée à la recherche de boîtes à espacement variable, soulevant la nécessité du développement d'une méthode apte à prendre en compte toutes les caractéristiques connues de ces motifs, pour mieux cibler nos recherches et choisir des modes statistiques de sélection à orientation biologique.

Le programme SIGffRid a été développé par Fabrice Touzain ; il définit les boîtes constituant les motifs *via* le programme R'MES [Sch97], il les recherche dans un premier temps par couples conservés dans les séquences intergéniques amonts de paires d'orthologues²⁰ en autorisant une légère variation de l'espace les séparant. Il tient donc compte à la fois des propriétés structurales

¹⁸Zones promotrices : zones situées dans le génome en amont des gènes.

¹⁹<http://www.inra.fr/bia/J/AB/genome/RMES2>

²⁰Orthologues : gènes d'espèces différentes résultant de l'évolution d'un seul et même gène ancestral, aux fonctions le plus souvent identiques

des motifs (espace entre les boîtes, boîtes, variations), et de la conservation des motifs régulateurs entre bactéries phylogénétiquement proches (orthologues). Des couples de mots sont recherchés dans les paires de séquences amonts d'orthologues selon les critères suivants :

- ils sont présents dans les deux séquences avec un espacement voisin (± 1 nucléotide),
- ils ont un écart typique entre motifs d'un site de régulation (14 à 20 lettres),
- ils sont statistiquement sur-représentés sur l'ensemble du génome.

Cette définition des mots recherchés limite fortement le nombre de résultats. Les couples de mots sélectionnés sont alors regroupés par similarité de doublet de trinuécléotides. Une étape d'évaluation est ensuite réalisée reposant sur des tests statistiques tenant compte de la sur-représentation dans les régions amonts des motifs consensus. Une liste de motifs est fournie comme résultat avec des scores associés permettant d'évaluer la pertinence de leur rôle dans la régulation.

Ce programme a été utilisé avec les génomes de *Streptomyces coelicolor* et *Streptomyces avermitilis*, nous avons pu retrouver directement trois motifs de sites de fixation de facteurs σ ou des variations de sites connus mais jusqu'à présent mal caractérisés (dont sigR : ggaat-n(18)-gttg, BldN : cgtaac-n(16)-cgttga, HrdB...) avec de nombreux nouveaux gènes potentiellement régulés par ces facteurs (un seul site est connu biologiquement pour BldN pour le moment).

Une interface graphique a été réalisée afin de permettre une utilisation simplifiée de ce logiciel par des biologistes et il sera bientôt accessible sur le site portail de la Bioinformatique en Lorraine : <http://bioinfo.loria.fr/>.

4.4 Conclusion

Je souhaite orienter mes recherches en Bioinformatique dans le cadre d'applications de mes travaux de géométrie discrète à l'étude des propriétés géométriques des macromolécules biologiques. Plus précisément, nous allons utiliser les travaux réalisés au chapitre 3 (cf. page 112 pour des explications plus précises) dans le cadre du docking des protéines, c'est-à-dire l'identification des possibilités d'interaction entre les surfaces de deux protéines. Nous participons à une opération sur ce thème dans le cadre du CPER (Contrat de Plan Etat Région) 2007-2013.

Conclusion et perspectives de recherche

Je reprends dans ce chapitre les différents thèmes abordés dans ce mémoire et, pour chacun d'eux, j'évoque les perspectives de recherche et les travaux que je souhaite entreprendre à court ou moyen terme.

Dans le premier chapitre de ce mémoire, j'ai proposé des éléments pour l'étude des courbes et surfaces *régulières*, c'est-à-dire non bruitées.

Un travail préliminaire sur une étude des coniques discrètes est présenté dans la section 1.5 de ce chapitre. Il me semble en effet intéressant d'*enrichir la panoplie des objets discrets* en proposant de nouvelles définitions, comme par exemple, celle de la parabole discrète proposée dans ce mémoire avec une épaisseur non constante. L'étape suivante sera de résoudre le problème de la reconnaissance de ces objets et ensuite de généraliser ce travail à l'ensemble des coniques discrètes.

Une étude précise des *représentations polygonales d'une courbe discrète* a été réalisée dans la section 1.2. Ces représentations polygonales sont obtenues en utilisant la suite des segments fondamentaux de la courbe discrète (appelée *couverture tangentielle* par F. Feschet [Fes06]). Nous avons montré que l'utilisation des segments fondamentaux d'une courbe discrète est une démarche pertinente pour obtenir une représentation polygonale *significative* de la courbe discrète étudiée, c'est-à-dire au plus proche de la courbe et respectant ses concavités et convexités. Nous avons mis en évidence le rôle essentiel des points d'appuis des segments fondamentaux de la courbe qui permettent d'obtenir une telle représentation polygonale. Nous nous intéressons à la dimension 3 et souhaitons étudier, dans le même esprit que pour les courbes, les *représentations polyédriques réversibles d'un objet discret 3D*. Une première étape dans cette voie est d'analyser la *meilleure* représentation polyédrique d'un objet constitué de deux morceaux de plans discrets naïfs. Cependant, une première question se pose : qu'appelons-nous *meilleure* représentation polyédrique d'un objet discret ? Les critères retenus dans ce mémoire pour les courbes discrètes 2D peuvent-ils être considérés pour la dimension 3 ? Ou bien faut-il obtenir des facettes de formes et de tailles identiques comme le suggère, pour les sphères, I. Sivignon [SDC04] ? Ou encore, un nombre minimal de facettes est-il souhaitable sachant qu'il a été montré [SC06] que polyédriser une surface discrète avec un nombre minimal de plans discrets connexes maximaux est un problème NP-complet ? Une étape préliminaire à ce travail est de répondre à ces questions. Ceci permettra de fixer les objectifs à atteindre et d'en déduire des critères d'évaluation pour comparer les différents algorithmes proposés.

L'approche utilisée dans la section 1.2 peut également être utilisée dans le cadre de mon travail sur les objets bruités ainsi que les résultats obtenus sur la reconnaissance des morceaux de plans naïfs (section 1.3) et sur l'estimation des normales du bord d'un objet discret (section

1.4). En effet, de manière générale, une étude théorique des objets discrets *réguliers* est indispensable et conduit à des résultats qui sont étendus puis exploités dans le cadre plus réaliste des applications où les objets d'étude sont bruités.

Une grande partie des travaux que j'ai entrepris ces dernières années se situe dans le cadre de *l'élaboration d'une géométrie discrète pour les objets discrets bruités*. Ce sujet est central dans l'équipe ADAGIo et c'est le thème du projet blanc de l'ANR GEODIB dont j'ai piloté la rédaction et dont je suis la coordinatrice depuis novembre 2006. Nous nous intéressons à l'étude des courbes et surfaces discrètes bruitées correspondant aux bords des objets discrets. Notre objectif est de développer dans le contexte de la géométrie discrète des outils adaptés permettant de tenir compte du bruit et des imperfections dus aux outils et aux méthodes d'acquisition des données, ceci tout en contrôlant les approximations effectuées.

Cette étude des courbes et surfaces bruitées peut se décomposer selon les 3 axes suivants :

- (i) *Définitions de primitives géométriques adaptées aux objets discrets bruités* : les notions de segments flous arithmétiques et morceaux flous de plans discrets définies dans les chapitres 2 et 3 ont initié ce travail et nous souhaitons définir d'autres primitives adaptées à notre problème.
- (ii) *Extraction de paramètres géométriques sur les objets discrets bruités* : l'étude des propriétés arithmétiques, géométriques et combinatoires de ces objets est fondamentale pour obtenir des algorithmes efficaces (reconnaissance, parcours, ...) ainsi que pour extraire des paramètres géométriques (périmètre, courbure, vecteur normal, aire, ...) sur les courbes et surfaces discrètes bruitées. Dans le chapitre 2, les premiers travaux sur des estimateurs de tangentes, de courbure et de longueur de courbes discrètes bruitées 2D et 3D sont présentés et, du côté des surfaces discrètes bruitées, des estimateurs de forme, d'aire et de normales ont été définis dans le chapitre 3.
- (iii) *Analyse et reconstruction des objets discrets bruités* : en utilisant les définitions et les algorithmes proposés dans le premier axe, nous cherchons à développer des nouveaux outils d'analyse de ces objets et en reconstruire des représentations continues.

Des algorithmes de reconnaissance des segments flous 2D, 3D et des morceaux de plans flous ont été exposés dans ce mémoire. Pour les segments flous, deux approches ont été proposées, l'une arithmétique reposant sur un résultat de la théorie des nombres [HW89] et l'autre géométrique utilisant des propriétés de l'enveloppe convexe [HT88] de l'ensemble à reconnaître. Pour les morceaux flous de plans discrets, l'approche géométrique a été explorée ainsi qu'une approche de programmation linéaire utilisant l'algorithme du simplexe. Des pistes restent à explorer, certains algorithmes de reconnaissance de morceaux de plans discrets naïfs, comme par exemple celui présenté dans la section 1.4, peuvent être adaptés à la reconnaissance des morceaux flous de plans discrets. L'approche dans l'espace dual proposée par J. Vittone [VC00] est aussi à considérer. Je souhaite faire une étude précise de toutes ces méthodes, où le paramètre d'épaisseur fixé possède différentes significations (distance euclidienne, distance verticale, ...). Des comparaisons des performances de ces différentes techniques devront être réalisées.

Le travail sur la recherche d'estimateurs de paramètres géométriques sur les courbes et surfaces bruitées de l'axe (ii) doit être poursuivi en améliorant les premiers estimateurs obtenus

et en l'enrichissant par de nouveaux. Certains pourront être déduits de ceux obtenus pour les courbes et surfaces non bruitées. Dans cette voie, la technique d'estimation de normales proposée dans le chapitre 1, section 1.3, travaillant sur les coupes de l'objet discret, peut être adaptée aux objets bruités par le calcul des tangentes d'épaisseurs ν , notion introduite dans le chapitre 2. Avec le même principe, en utilisant le calcul de la courbure d'épaisseur ν de courbes discrètes 2D, un estimateur de la courbure en chaque point du bord d'un objet discret 3D peut être obtenu selon une technique adaptée de celle proposée par A. Lenoir [Len97]. Une évaluation du comportement de ces différents estimateurs devra être réalisée et les modalités de cette évaluation sont encore à définir. En effet, des tests pourront être réalisés sur un panel d'objets discrets issus de la discrétisation d'objets continus dont les caractéristiques géométriques sont connues, ces objets discrets seront bruités selon une règle spécifique, puis les estimateurs seront évalués en fonction des différences entre les paramètres géométriques calculés et ceux des objets initiaux. L'évaluation des estimateurs pour les objets non bruités est aussi effectuée en étudiant la propriété de *convergence asymptotique* [Coe02], c'est-à-dire le comportement des estimateurs en considérant la discrétisation d'un objet continu selon une grille de plus en plus fine. Sur des objets bruités, ce type d'étude n'est pas réalisable sans connaître la nature *formelle* du bruit présent dans les objets étudiés. Cette formalisation des différentes natures de bruits présents dans les données étudiées et sa prise en compte dans nos travaux est une voie de recherche ambitieuse. Dans ce mémoire, j'ai uniquement travaillé avec du bruit correspondant au modèle proposé dans l'introduction du chapitre 2 mais sans le formaliser ; cette piste est encore à explorer.

De nombreux travaux pour l'axe (iii) doivent encore être réalisés afin d'obtenir des algorithmes de décomposition de courbes et de surfaces bruitées en primitives discrètes adaptées aux bruits, en contrôlant les approximations effectuées et en utilisant diverses stratégies de parcours. L'étape suivante concerne les représentations polygonales ou polyédrales des objets discrets. L'utilisation des paramètres géométriques locaux semble être une voie à approfondir afin d'obtenir les représentations reflétant au mieux les caractéristiques géométriques des objets discrets ainsi représentés.

Dans ce cadre, au vu des résultats obtenus sur les courbes non bruitées, il semble incontournable de faire l'analyse de la suite des segments flous maximaux (ou fondamentaux) d'une courbe bruitée pour une épaisseur fixée. D'autres types de polygonalisation tenant compte du bruit et de la forme de la courbe pourront ainsi être déterminés. Cependant, le paramètre d'épaisseur reste toujours à déterminer manuellement, un de mes objectifs est d'obtenir une méthode s'adaptant au bruit présent dans la courbe sans paramètre. La segmentation multi-niveaux présentée au paragraphe 2.2.3 du chapitre 2 est une première étape dans ce sens mais des problèmes persistent et nécessitent une étude approfondie. Comme F. Feschet le suggère dans [Fes06], la combinaison d'une segmentation en segments flous maximaux et d'une analyse multi-niveaux est une voie à explorer.

Une méthode de segmentation du bord d'un objet discret en morceaux de plans flous pour une épaisseur donnée a été exposée dans le chapitre 3. Cette segmentation est guidée par des paramètres géométriques pré-calculés. Les résultats obtenus sont très encourageants et confirment l'intérêt de la stratégie choisie qui est de suivre au mieux la forme de l'objet. Dans ce but, il semble naturel d'essayer d'obtenir une structure semblable à celle de la suite des segments flous maximaux pour les courbes discrètes. Cependant, sur les bords d'objets discrets 3D se pose le problème du parcours et la notion d'extrémités d'un segment se transforme en contour d'un morceau de plan . . . On perçoit alors la complexité de définir une telle structure. Cependant la notion

de *patch* introduite au chapitre 3 peut être un point de départ dans cette étude en considérant des *patches* consécutifs sur le bord d'un objet discret 3D.

Plusieurs applications dans l'analyse d'image ont été proposées dans ce mémoire, elles sont indispensables pour valider les approches proposées et mettre à l'épreuve nos algorithmes sur des données réelles. Dans ce cadre, nous souhaitons poursuivre les travaux engagés avec pour objectif d'obtenir des descripteurs de formes robustes pouvant passer avec succès les compétitions proposées dans le milieu de l'analyse de documents (ce type de compétitions est par exemple proposé au workshop GREC).

Une remarque importante est que les travaux proposés dans les chapitres 2 et 3 avec l'approche géométrique peuvent être adaptés à des ensembles de points déconnectés ce qui peut être intéressant pour des applications spécifiques où les données fournies sont de cette nature.

C'est le cas pour un des travaux que j'ai réalisés en bioinformatique, j'ai proposé un algorithme de calcul de la courbure de l'ADN dont la représentation tridimensionnelle est formée de points déconnectés.

Dans le cadre de l'étude de la géométrie des macromolécules, je m'intéresse au *docking* des protéines, c'est-à-dire à la compréhension et à la prédiction des interactions entre protéines. Ce phénomène est au centre d'applications pratiques comme l'ingénierie des protéines et la conception des médicaments. Dans le cadre du CPER (Contrat de Plan Etat Région) 2007-2013, je participe à l'action *Modélisation des assemblages protéines-protéines* dont l'objectif est de développer un outil permettant (1) d'identifier rapidement les modes d'assemblage les plus pertinents entre plusieurs unités protéiques, (2) de les manipuler interactivement et d'analyser les interfaces protéine-protéine ainsi créées. Un premier objectif correspondant à la première phase du projet est d'utiliser et d'adapter les estimateurs proposés dans le chapitre 3 en adoptant une approche *multirésolution* pour émettre des hypothèses d'assemblage entre 2 protéines.

Ces applications, permettant une validation des travaux en géométrie discrète, ne peuvent être entreprises que dans le cadre de collaborations avec des chercheurs de ces domaines. D'une manière générale, je souhaite accentuer ce type de collaborations toujours très enrichissantes. En effet, les discussions préambules à ces travaux communs apportent souvent de nouveaux problèmes et donc de nouveaux thèmes théoriques à développer en géométrie discrète.

Bibliographie

- [AAS97] E. ANDRES, R. ACHARYA, and C. SIBATA. Discrete analytical hyperplanes. *CVGIP : Graphical Model and Image Processing*, 59(5) :302–309, 1997.
- [AD91] N. ANSARI and E. J. DELP. On detecting dominant points. *Pattern Recognition*, 24(5) :441–451, 1991.
- [And92] E. ANDRES. Cercles discrets et rotations discrètes. Master’s thesis, Université Louis Pasteur, Strasbourg, 1992.
- [And00] É. ANDRES. Modélisation Analytique Discrète d’Objets Géométriques. Habilitation à diriger des recherches, Université de Poitiers, Décembre 2000.
- [And03] E. ANDRES. Discrete linear objects in dimension n : the standard model. *Graphical Models*, 65(1-3) :92–111, 2003.
- [ANF97] E. ANDRES, P. NEHLIG, and J. FRANÇON. Supercover of straight lines, planes and triangles. In *DGCI’97 : 7th International Conference on Discrete Geometry for Computer Imagery*, volume 1347 of *LNCS*, pages 243–257, 1997.
- [BAB00] V. BRIMKOV, E. ANDRES, and R. BARNEVA. Object discretization in higher dimensions. In *DGCI’00, 9th International Conference on Discrete Geometry for Computer Imagery*, volume 1953 of *LNCS*, pages 210–221, 2000.
- [Bai00] H. BAIRD. State of the art of document image degradation modeling. In *4th IAPR International Workshop on Document Analysis Systems*, volume 1, 2000.
- [BCK07] V. BRIMKOV, D. COEURJOLLY, and R. KLETTE. Digital planarity - a review. *Discrete Applied Mathematics*, 155(4) :468–495, 2007.
- [BE94] T.L. BAILEY and C. ELKAN. Fitting a mixture model by expectation maximization to discover motifs in biopolymers. In *Proceedings of the Second International Conference on Intelligent System for Molecular Biology*, pages 28–36, Stanford, California, USA, August 1994.
- [BF94] P. BORIANNE and J. FRANÇON. Reversible polyhedrization of discrete volumes. In *DCGI’94, 4th International Conference on Discrete Geometry for Computer Imagery*, pages 157–168, Grenoble, 1994.
- [BK00] T. BÜLOW and R. KLETTE. Rubber band algorithm for estimating the length of digitized space-curves. In *International Conference on Pattern Recognition*, pages 3551–3555, 2000.
- [BK02] T. BÜLOW and R. KLETTE. Digital curves in 3d space and a linear-time length estimation algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7) :962–970, 2002.
- [BKO00] M. DE BERG, M. VAN KREVELD, M. OVERMARS, and O. SCHWARZKOPF. *Computational Geometry : Algorithms and Applications*. Springer-Verlag, Heidelberg, 2000.

- [BM03] R. MALGOUYRES J. BURGUET. Strong thinning and polyhedric approximation of the surface of a voxel object. *Discrete Applied Mathematics*, 125(1) :93–114, 2003.
- [BMH91] A. BOLSHOY, P. MCNAMARA, P. HARRINGTON, and E.N. TRIFONOV. Curved DNA without a-a : experimental estimation of all 16 DNA wedge angles. *Proceedings of the National Academy of Science*, 88 :2312–2316, 1991.
- [Bor96] G. BORGEFORS. On digital distance transforms in three dimensions. *Computer Vision and Image Understanding*, 64(3) :368–376, 1996.
- [Bur02] J. BURGUET. *Surfaces discrètes multi-échelle, squelettisation, polyédricisation et opérations ensemblistes sur des polyèdres*. PhD thesis, Université de Caen, 2002.
- [Buz05] L. BUZER. An elementary algorithm for digital line recognition in the general case. In *DGCI'05, 12th International Conference on Discrete Geometry for Computer Imagery*, volume 3429 of *LNCS*, pages 299–310, 2005.
- [BV99] J.-P. BRAQUELAIRE and A. VIALARD. Euclidean paths : A new representation of boundary of discrete regions. *CVGIP : Graphical Models and Image Processing*, 61(1) :16–43, 1999.
- [CDJS06] D. COEURJOLLY, F. DUPONT, L. JOSPIN, and I. SIVIGNON. Optimization schemes for the reversible discrete volume polyhedrization using marching cubes simplification. In *DGCI'06, 13th International Conference on Discrete Geometry for Computer Imagery*, volume 4245 of *LNCS*, pages 413–424. Springer, 2006.
- [CFTT03] D. COEURJOLLY, F. FLIN, O. TEYTAUD, and L. TOUGNE. Multigrid convergence and surface area estimation. In *Theoretical Foundations of Computer Vision "Geometry, Morphology, and Computational Imaging"*, volume 2616 of *LNCS*, pages 101–119, 2003.
- [Chv83] V. CHVATAL. *Linear Programming*. Freeman, New York, 1983.
- [CK97] D. COHEN and A. KAUFMAN. 3d line voxelization and connectivity control. In *IEEE Computer Graphics and Applications*, pages 80–87, 1997.
- [CK04] D. COEURJOLLY and R. KLETTE. A comparative evaluation of length estimators of digital curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2) :252–257, 2004.
- [CM91] J.-M. CHASSERY and A. MONTANVERT. *Géométrie discrète en analyse d'images*. Hermès, 1991.
- [CMC07] D. COEURJOLLY, A. MONTANVERT, and J.-M. CHASSERY, editors. *Géométrie discrète et images numériques*. Traité IC2, Signal et Image. Hermès Paris, France, September 2007.
- [CMT01] D. COEURJOLLY, S. MIGUET, and L. TOUGNE. Discrete curvature based on osculating circle estimation. In *International Workshop on Visual Form*, volume 2059 of *LNCS*, pages 303–312, 2001.
- [Coe02] D. COEURJOLLY. *Algorithmique et géométrie discrète pour la caractérisation des courbes et des surfaces*. PhD thesis, Université Lumière, Lyon 2, décembre 2002.
- [CP95] J.-P. COCQUEREZ and S. PHILIPP, editors. *Analyse d'images : filtrage et segmentation*. Enseignement de la physique. Masson, France, 1995.
- [CS03] D. COEURJOLLY and S. SVENSSON. Estimation of curvature along curves with application to fibres in 3d images of paper. *LNCS*, 2749 :247–254, January 2003.

- [DCA06] M. DEXET, D. COEURJOLLY, and E. ANDRES. Invertible polygonalization of 3d planar digital curves and application to volume data reconstruction. In *International Symposium on Visual Computing*, volume 4292 of *LNCS*, pages 514–523. Springer, 2006.
- [Dex06] M. DEXET. *Architecture d'un modeleur géométrique à base topologique d'objets discrets et méthodes de reconstruction en dimensions 2 et 3*. PhD thesis, Université de Poitiers, décembre 2006.
- [DH98] M. DLAKIC and R.E. HARRINGTON. Diamod : display and modeling of DNA bending. *Bioinformatics*, 14 :326–331, 1998.
- [Dic98] R.E. DICKERSON. Dna bending : the prevalence of kinkiness and the virtues of normality. *Nucleic Acids Research*, 26 :1906–1926(21), 1998.
- [DNW02] D. DUHOVNY, R. NUSSINOV, and H.J. WOLFSON. Efficient unbound docking of rigid molecules. In *Workshop on Algorithms in Bioinformatics*, volume 2452 of *LNCS*, 2002.
- [Dör04] H. DÖRKSEN. *Shape representations of digital sets based on convexity properties*. PhD thesis, Hamburg University, Juni 2004.
- [DS87] L. DORST and A. W. M. SMEULDERS. Length estimators for digitized contours. *Computer Vision, Graphics, and Image Processing*, 40(3) :311–333, 1987.
- [Fes06] F. FESCHET. Méthodes de géométrie discrète pour la caractérisation, la représentation et la reconstruction d'objets discrets. Habilitation à diriger des recherches, Université d'Auvergne I, Janvier 2006.
- [FJT06] C. FIORIO, D. JAMET, and J.-L. TOUTANT. Discrete circles : an arithmetical approach with non- constant thickness. In Angela Y. Wu Longin Jean Latecki, David M. Mount, editor, *Vision Geometry XIV, Electronic Imaging*, volume 6066, San Jose (CA), USA, January 2006. SPIE.
- [FP99] J. FRANÇON and L. PAPIER. Polyhedrization of the boundary of a voxel object. In *DGCI'99, International Conference on Discrete Geometry for computer Imagery*, volume 1568 of *LNCS*, pages 425–434. Springer, 1999.
- [FR95] O. FIGUEIREDO and J.P. REVEILLÈS. A contribution to 3d digital lines. In *Proc. DCGI'95*, pages 187–198, 1995.
- [Fran95] J. FRANÇON. Arithmetic planes and combinatorial manifolds. In *Proceedings of the 5th International Workshop Discrete Geometry for computer Imagery*, pages 209–217, Clermont-Ferrand (France), September 1995.
- [Fre70] H. FREEMAN. Boundary encoding and processing. In *Picture Processing and Psychopictorics*, pages 241–266. New York Academic, 1970.
- [FT99] F. FESCHET and L. TOUGNE. Optimal time computation of the tangent of a discrete curve : application to the curvature. In *DGCI'99, 7th International Conference on Discrete Geometry for Computer Imagery*, volume 1568, pages 31–40, 1999.
- [FZL97] D. FLATTERS, K. ZAKRZEWSKA, and R. LAVERY. Internal coordinate modeling of dna : Force field comparisons. *Journal of Computational Chemistry*, 18(8), 1997.
- [GB99] A.E. GABRIELIAN and A. BOLSHOY. Sequence complexity and dna curvature. *Computers & Chemistry*, 23(3-4) :263–274, 1999.
- [Ger99] Y. GÉRARD. *Contribution à la géométrie discrète*. PhD thesis, Université d'Auvergne, 1999.

- [GH01] B. GÄRTNER and T. HERRMANN. Computing the width of a point set in 3-space. *J. Exp. Algorithmics*, 4 :3, 2001.
- [GLB99] A.E. GABRIELIAN, D. LANDSMAN, and A. BOLSHOY. Curved dna in promoter sequences. *In Silico Biology*, 1 :17, 1999.
- [Her92] G.T. HERMAN. Discrete multidimensional jordan surfaces. *CVGIP : Graphical Model and Image Processing*, 54(6) :507–515, 1992.
- [Her98] G.T. HERMAN. *Geometry of Digital Spaces*. Birkhauser, 1998.
- [Hil04] X. HILAIRE. *Segmentation robuste de courbes discrètes 2D et applications à la rétro-conversion de documents techniques*. PhD thesis, Institut National Polytechnique de Lorraine, 2004.
- [Hil06] X. HILAIRE. Ranvec and the arc segmentation contest : Second evaluation. In *postproceeding of GREC'05*, volume 3926 of *LNCS*, pages 362–368, 2006.
- [HK03] S. HERMANN and R. KLETTE. Multigrid analysis of curvature estimators. In *Proceedings of Image and Vision Computing New Zealand*, pages 108–112, 2003.
- [HK07] S. HERMANN and R. KLETTE. A comparative study on 2d curvature estimators. In *ICCTA*, pages 584–589, 2007.
- [HS98] J.W. HARRIS and H. STOCKER. *Handbook of mathematics and computational science*, Springer-Verlag. 1998.
- [HT88] M.E. HOULE and G.T. TOUSSAINT. Computing the width of a set. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 10(5) :761–765, 1988.
- [HT06] X. HILAIRE and K. TOMBRE. Robust and accurate vectorization of line drawings. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(6) :890–904, 2006.
- [HW89] G.H. HARDY and E.M. WRIGHT. *An introduction to the theory of numbers*. Oxford society, 1989.
- [JK97] A. JONAS and N. KIRYATI. Digital representation schemes for 3d curves. *Pattern Recognition*, 30(11) :1803–1816, 1997.
- [Kas90] R. KASTURI, S.T. BOW, W. ELMASRI, J. SHAH, J.R. GATTIKER, and U.B. MOKATE. A system for interpretation of line drawings. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(10), 1990.
- [Kau87] A. KAUFMAN. Efficient algorithms for 3-d scan conversion of parametric curves, surfaces, volumes. In *Computer Graphic 21, 4*, pages 171–179, 1987.
- [KF03] A. KOLESNIKOV and P. FRÄNTI. Reduced-search dynamic programming for approximation of polygonal curves. *Pattern Recognition Letters*, 24(14) :2243–2254, 2003.
- [KHBSM00] T. KANUNGO, R. M. HARALICK, H. S. BAIRD, W. STUEZLE, and D. MADIGAN. A statistical, nonparametric methodology for document degradation model validation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11) :1209–1223, 2000.
- [Kim82] C.E. KIM. Digital convexity, straightness and convex polygons. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 4, pages 618–626, 1982.
- [Kim83] C.E. KIM. Three-dimensional digital line segments. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 5, pages 231–234, 1983.

- [Kir97] Y. KIRSCHHOFFER. Estimation de normale dans un espace discret 3d. analyse comparative de méthodes. Mémoire de dea, Université Louis Pasteur, Strasbourg, 1997.
- [KKY99] R. KLETTE, V.A. KOVALEVSKY, and B. YIP. Length estimation of digital curves. *Vision Geometry VIII, SPIE. Denver*, pages 117–129, 1999.
- [Kov90] V. A. KOVALESKY. New definition and fast recognition of digital straight segments and arcs. In *10th International Conference on Pattern Recognition*, Atlantic City, New Jersey, 1990.
- [KR80] C. E. KIM and A. ROSENFELD. On the convexity of digital regions. In *Pattern Recognition, volume 5, pages 1010–1015*, 1980.
- [KR82] C.E. KIM and A. ROSENFELD. Digital straight lines and convexity of digital regions. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 4, pages 149–153, 1982.
- [KR96] T.Y. KONG and A. ROSENFELD. *Topological Algorithms for Digital Image Processing*. North Holland, 1996.
- [KR04] R. KLETTE and A. ROSENFELD. *Digital Geometry - Geometric Methods for Digital Picture Analysis*. Morgan Kaufmann, 2004.
- [KS01] R. KLETTE and H.J. SUN. Digital planar segment based polyhedrization for surface area estimation. In *International Workshop on Visual Form*, volume 2059 of *LNCS*, pages 356–366. Springer, 2001.
- [LBL01] X. LIU, D.L. BRUTLAG, and J.S. LIU. Bioprospector : discovering conserved dna motifs in upstream regulatory regions of co-expressed genes. *Pacific Symposium on Biocomputing*, pages 127–138, 2001.
- [Lef75] S. LEFSCHETZ. Applications of algebraic topology. In *Springer Berlin*, 1975.
- [Len97] A. LENOIR. Fast estimation of mean curvature on the surface of a 3d discrete object. In *DGCI'97, 7th International Workshop on Discrete Geometry for Computer Imagery*, volume 1347 of *LNCS*, pages 175–186, Montpellier, France, December 1997.
- [LK05] F. LI and R. KLETTE. The class of simple cube-curves whose mlps cannot have vertices at grid points. In *DGCI'05, 12th International Conference on Discrete Geometry for computer Imagery*, volume 3429 of *LNCS*, pages 183–194. Springer, 2005.
- [LK06] F. LI and R. KLETTE. Shortest paths in a cuboidal world. In *IWCIA'06, 11th International Workshop on Combinatorial Image Analysis*, volume 4040 of *LNCS*, pages 415–429. Springer, 2006.
- [LK07] F. LI and R. KLETTE. Euclidean shortest paths in simple cube curves at a glance. In *CAIP'07, 12th International Conference on Computer Analysis of Images and Patterns*, volume 4673 of *LNCS*, pages 661–668. Springer, 2007.
- [LMR96] A. LENOIR, R. MALGOUYRES, and M. REVENU. Fast computation of the normal vector field of the surface of a 3-d discrete object. In *DGCI'96, 6th International Workshop on Discrete Geometry for Computer Imagery*, volume 1176 of *LNCS*, pages 101–112, Lyon, France, November 1996.
- [LR02] L. J. LATECKI and A. ROSENFELD. Recovering a polygon from noisy data. *Computer Vision and Image Understanding*, 86(1) :32–51, 2002.

- [LVV05] J.-O. LACHAUD, A. VIALARD, and F. de VIEILLEVILLE. Analysis and comparative evaluation of discrete tangent estimators. In *DGCI'05, 12th International Conference on Discrete Geometry and Computer Imagery*, pages 240–251, 2005.
- [LW00a] C. LINCKE and C.A. WÜTHRICH. Properties of surface digitizations. In *Eurographics*, volume 19(3), 2000.
- [LW00b] C. LINCKE and C.A. WÜTHRICH. Towards a unified approach between digitizations of linear objects and discrete analytical objects. In *WSCG*, 2000.
- [Mal97] R. MALGOUYRES. A definition of surfaces of z : A new 3d discrete jordan theorem. *Theor. Comput. Sci.*, 186(1-2) :1–41, 1997.
- [Mar03] M. MARJI. *On the detection of dominant points on digital planar curves*. PhD thesis, Wayne State University, 2003.
- [Mel87] A. A. MELKMAN. On-line construction of the convex hull of a simple polyline. *Inf. Process. Lett.*, 25(1) :11–12, 1987.
- [MM92] F. MOKHTARIAN and A. K. MACKWORTH. A theory of multiscale, curvature-based shape representation for planar curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(8) :789–805, 1992.
- [MP69] M. MINSKY and S. PAPER. Perceptrons. In *M.I.T. Press*, 1969.
- [MR81] D. G. MORGENTHALER and A. ROSENFELD. Surfaces in three-dimensional digital images. *Information and Control*, 51(3) :227–247, 1981.
- [Ngu05] T.P. NGUYEN. Optimisation du calcul de la courbure de l'ADN. Mémoire de dea, IFI, Hanoï, Vietnam, October 2005.
- [OL81] M. H. OVERMARS and J. van LEEUWEN. Maintenance of configurations in the plane. *J. Comput. Syst. Sci.*, 23(2) :166–204, 1981.
- [PF98] L. PAPIER and J. FRANÇON. Évaluation de la normale au bord d'un objet discret 3d. *Revue internationale de CFAO et d'Infographie*, 13(2) :205–226, 1998.
- [Pro05] L. PROVOT. Reconnaissance de morceaux de plans discrets bruités. Mémoire de DEA, Université Henri Poincaré, Nancy, June 2005.
- [PS85] F. P. PREPARATA and M. I. SHAMOS. *Computational Geometry - An Introduction*. Springer, 1985.
- [PV94] J.C PEREZ and E. VIDAL. Optimum polygonal approximation of digitized curves. *Pattern Recognition Letters*, 15 :743–750, 1994.
- [Ram72] U. RAMER. An iterative procedure for the polygonal approximation of plane curves. *Computer Graphics and Image Processing*, 1 :244–256, 1972.
- [Rap04] F. RAPAPORT. Calcul de la courbure de l'ADN. Mémoire de DEA, Supelec, September 2004.
- [RC92] A. RATTARANGSI and R. T. CHIN. Scale-based detection of corners of planar curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(4) :430–449, 1992.
- [Rev91] J.P. REVEILLÈS. Géométrie discrète, calculs en nombres entiers et algorithmique, 1991. Thèse d'état. Université Louis Pasteur, Strasbourg.
- [RH84] A. ROSENFELD and S. HABER. The perimeter of a fuzzy set. *Pattern Recognition*, 17(6) :678, 1984.

- [RJ73] A. ROSENFELD and E. JOHNSTON. Angle detection on digital curves. *IEEE Transactions on Computers*, pages 875–878, 1973.
- [Ros84] A. ROSENFELD. The fuzzy geometry of image subsets. *Pattern Recognition Letters*, 2(5) :311–317, 1984.
- [Ros90] A. ROSENFELD. Fuzzy rectangles. *Pattern Recognition Letters*, 11(10) :677–679, 1990.
- [RR91] B. K. RAY and K. S. RAY. A new approach to polygonal approximation. *Pattern Recognition Letters*, 12(4) :229–234, 1991.
- [RR95] B.K. RAY and K.S. RAY. A new split-and-merge technique for polygonal approximation of chain coded curves. *Pattern Recognition Letters*, 16, February 1995.
- [RW89] P. L. ROSIN and G. A. WEST. Segmentation of edges into lines and arcs. *Image and Vision Computing*, 7(2) :109–114, May 1989.
- [RW95] P. L. ROSIN and G. A. WEST. Nonparametric segmentation of curves into various representations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(12) :1140–1153, 1995.
- [Sal01] M. SALOTTI. An efficient algorithm for the optimal polygonal approximation of digitized curves. *Pattern Recognition Letters*, 22(2) :215–221, February 2001.
- [SC06] I. SIVIGNON and D. COEURJOLLY. Minimal decomposition of a digital surface into digital plane segments is np-hard. In *DGCI'06, 13th International Conference on Discrete Geometry for Computer Imagery*, volume 4245 of *LNCS*, pages 674–685, 2006.
- [Sch97] S. SCHBATH. An efficient statistic software to detect over- and under-represented words in DNA sequences. *Journal Computer Biology*, 4 :189–192, 1997.
- [SCM91] P. SAINT-MARC, J.-S. CHEN, and G. MEDIONI. Adaptive smoothing : A general tool for early vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6) :514–529, 1991.
- [SDC04] I. SIVIGNON, F. DUPONT, and J.-M. CHASSERY. Decomposition of a three-dimensional discrete object surface into discrete plane pieces. *Algorithmica*, 38(1) :25–43, 2004.
- [SG80] J. SKLANSKY and V. GONZALEZ. Fast polygonal approximation of digitized curves. *Pattern Recognition*, 12(5) :327–331, 1980.
- [Siv04] I. SIVIGNON. *De la caractérisation des primitives à la reconstruction polyédrique de surfaces en géométrie discrète*. PhD thesis, Institut National Polytechnique de Grenoble, 2004.
- [Sla70] J. SLANSKY. Recognition of convex blobs. In *Pattern Recognition*, volume 2, pages 3–10, 1970.
- [SNS03] N. SLADOJE, I. NYSTRÖM, and P. K. SAHA. Perimeter and area estimations of digitized objects with fuzzy borders. In *DGCI'03, 3rd International Conference on Discrete Geometry for Computer Imagery*, pages 368–377, 2003.
- [STB93] A.S. SHPIGELMAN, E.N. TRIFONOV, and A. BOLSHOY. Curvature : Software for the analysis of curved DNA. *Computer Applications in the Biosciences*, 9(4) :435–440, 1993.
- [Sven07] S. SVENSSON. A decomposition scheme for 3d fuzzy objects based on fuzzy distance information. *Pattern Recognition Letters*, 28(2) :224–232, 2007.

- [TAI05] M. TOMPA, N. LI, T.L. BAILEY, G.M. CHURCH, B. DE MOOR, E. ESKIN, A.V. FAVOROV, M.C. FRITH, Y. FU, W.J. KENT, V.J. MAKEEV, A.A. MIRONOV, W.S. NOBLE, G. PAVESI, G. PESOLE, M. RÉGNIER, N. SIMONIS, S. SINHA, G. THIJS, J. VAN HELDEN, M. VANDENBOGAERT, Z. WENG, C. WORKMAN, C. YE, and Z ZHU. Assessing computational tools for the discovery of transcription factor binding sites. *Nature Biotechnology*, 23(1) :137–144, 2005.
- [TC89] Cho-Huak TEH and R. T. CHIN. On the detection of dominant points on digital curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(8) :859–872, 1989.
- [Thu01] G. THÜRMER. Smoothing normal vectors on discrete surfaces while preserving slope discontinuities. *Computer Graphics Forum*, 20(2) :103–114, 2001.
- [TW97] G. THÜRMER and C. A. WÜTHRICH. Normal computation for discrete surfaces in 3d space. *Computer Graphics Forum*, 16(3) :15–26, 1997.
- [TWR99] M. TAJINE, D. WAGNER, and C. RONSE. Hausdorff discretization and its comparison to other discretization schemes. In *DGCI'99, 8th International Conference on Discrete Geometry for Computer Imagery*, volume 1568 of *LNCS*, pages 399–412, 1999.
- [Udu94] J.K. UDUPA. Multidimensional digital boundaries. *CVGIP : Graphical Model and Image Processing*, 56(4) :311–323, 1994.
- [USH82] J.K. UDUPA, S.N. SRIHARI, and G.T. HERMANN. Boundary detection in multidimension. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 4(1) :41–50, 1982.
- [VC00] J. VITTONI and J.M. CHASSERY. Recognition of digital naive planes and polyhedrization. In *9th International Conference on Discrete Geometry for Computer Imagery*, *LNCS 1953*, 2000.
- [Ver91] B. J. H. VERWER. Local distances for distance transformations in two and three dimensions. *Pattern Recognition Letters*, 12(11) :671–682, 1991.
- [Via96a] A. VIALARD. Chemins euclidiens : un modèle de représentation des contours discrets, 1996. Thèse. Université Bordeaux I, Bordeaux.
- [Via96b] A. VIALARD. Geometrical parameters extraction from discrete path. In *DGCI'96, 6th International Conference on Discrete Geometry for Computer Imagery*, volume 1176 of *LNCS*, Lyon, France, 1996. Springer-Verlag.
- [VLF05] F. de VIEILLEVILLE, J.-O. LACHAUD, and F. FESCHET. Maximal digital straight segments and convergence of discrete geometric estimators. In *Scandinavian Conference on Image Analysis*, pages 988–997, 2005.
- [WB91] Daniel M. WUESCHER and Kim L. BOYER. Robust contour decomposition using a constant curvature criterion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(1) :41–51, 1991.
- [WD84] K. WALL and P. DANIELSSON. A fast sequential method for polygonal approximation of digitized curves. *Computer Vision, Graphics and Image Processing*, 28(2) :220–227, 1984.
- [Wit83] A.P. WITKIN. Scale-space filtering. In *IJCAI*, pages 1019–1022, 1983.
- [WS93] M. WORRING and W.M. SMEULDERS. Digital curvature estimation. *CVGIP : Image Understanding*, 58(3) :366–382, 1993.

- [WV97] W. WAN and J. A. VENTURA. Segmentation of planar curves into straight-line segments and elliptical arcs. *CVGIP : Graphical Model and Image Processing*, 59(6) :484–494, 1997.
- [WW93] W.Y. WU and M.J.J. WANG. Detecting the dominant points by the curvature-based polygonal approximation. *Graphical Models and Image Processing*, 55(2) :79–88, March 1993.
- [YCK92] R. YAGEL, D. COHEN, and A. KAUFMAN. Normal estimation in 3d discrete space. In *The visual computer*, volume 8, pages 278–291, 1992.
- [Zad65] L.A. ZADEH. Fuzzy sets. *Information and Control*, 8(3) :338–353, 1965.

Chapitre 5

Liste des publications

Articles publiés dans une revue ou un ouvrage d'audience internationale avec comité de rédaction

1. I. DEBLED-RENNESON, F. FESCHET and J. ROUYER-DEGLI, *Optimal blurred segments decomposition of noisy shapes in linear time*, Computers & Graphics, 30(1), 2006.
2. I. DEBLED-RENNESON, J.-L. REMY and J. ROUYER-DEGLI, *Linear segmentation of discrete curves into fuzzy segments*, Discrete Applied Mathematics, 151 :122-137, October 2005.
3. Y. GERARD, I. DEBLED-RENNESON and P. ZIMMERMANN, *An elementary digital plane recognition algorithm*, Discrete Applied Mathematics, 151 :169-183, October 2005.
4. I. DEBLED-RENNESON, S. TABBONE, L. WENDLING, *Multiorde polygonal approximation of digital curves*, Electronic Letters on Computer Vision and Image Analysis, 5(2) :98-110, August 2005.
5. H. DÖRKSEN-REITER, I. DEBLED-RENNESON, *Convex and concave parts of digital curves*. In R. Klette, R. Kozera, L. Noakes, and J. Weickert, editors, Geometric Properties from Incomplete Data, volume 31 of Computational Imaging and Vision. Springer-Verlag, January 2005.
6. I. DEBLED-RENNESON, J.-L. REMY and J. ROUYER-DEGLI, *Detection of the discrete convexity of polyominoes*, Discrete Applied Mathematics, 125(1) :115-133, January 2003.
7. I. DEBLED-RENNESON, J.-P. REVEILLÈS, *A linear algorithm for segmentation of digital curves*, International Journal of Pattern Recognition and Artificial Intelligence, Volume 9, N. 6, December 1995.

Conférences d'audience internationale avec comité de sélection

8. T.P. NGUYEN and I. DEBLED-RENNESON, *Curvature estimation in noisy curves*, 12th International Conference on Computer Analysis of Images and Patterns, Lecture Notes in Computer Science, Volume 4673, Vienna, Austria, August 2007.
9. D. JAMET, E. DOMENJOUR and I. DEBLED-RENNESON, *Arithmetic discrete parabolas*, Advances in Visual Computing, Second International Symposium (ISVC 2006), Lecture Notes in Computer Science, Volume 4292, Lake Tahoe, NV, USA, November 2006.
10. L. PROVOT, L. BUZER and I. DEBLED-RENNESON, *Recognition of blurred pieces of discrete planes*, 13th International Conference on Discrete Geometry for Computer Imagery (DGCI'06), Lecture Notes in Computer Science, Volume 4245, Sgezed, Hongria, October 2006.
11. J.-P. SALMON, I. DEBLED-RENNESON, L. WENDLING, *A new method to detect arcs and segments from curvature profiles*, ICPR'06, Hong-Kong, China, August 2006.
12. H. DÖRKSEN-REITER, I. DEBLED-RENNESON, *A Linear algorithm for polygonal representations of digital sets*, 11th International Workshop on Combinatorial Image Analysis (IWCIA'2006), Lecture Notes in Computer Science, Volume 4040, Berlin, Germany, Juny 2006.
13. I. DEBLED-RENNESON, F. FESCHET and J. ROUYER-DEGLI, *Optimal blurred segments decomposition in linear time*, 12th International Conference on Discrete Geometry for Computer Imagery (DGCI'05), Lecture Notes in Computer Science, Volume 3429, Poitiers, France, April 2005.
14. I. DEBLED-RENNESON, S. TABBONE, L. WENDLING, *Fast polygonal approximation of digital curves*, ICPR'04, Volume 1, pp 465-468, Cambridge, United Kingdom, August 2004.
15. I. DEBLED-RENNESON, *Estimation of tangents to a noisy discrete curve*, Vision Geometry XII, SPIE, Electronic Imaging, San José, USA, January 2004.
16. I. DEBLED-RENNESON, J.-L. REMY and J. ROUYER-DEGLI, *Segmentation of discrete curves into fuzzy segments*, 9th International Workshop on Combinatorial Image Analysis (IWCIA'03), Electronic Notes in Discrete Mathematics, Palerme, Italy, May 2003.
17. D. COEURJOLLY, I. DEBLED-RENNESON and O. TEYTAUD, *Segmentation and length estimation of 3D discrete curves*, Digital and Image Geometry, editors G. Bertrand, A. Imiya and R. Klette, Lecture Notes in Computer Science, Volume 2243, pp. 295-313, 2000.
18. I. DEBLED-RENNESON, J.-L. REMY and J. ROUYER-DEGLI, *Detection of the discrete convexity of polyominoes*, 9th International Conference DGCI 2000, Uppsala, Sweden, Lecture Notes in Computer Science, Volume 1953, pp. 491-504, December 2000.
19. P. TELLIER and I. DEBLED-RENNESON, *3D discrete normal vectors*, 8th Internatio-

nal Conference DGCI'99, Lecture Notes in Computer Science, Volume 1568, pp. 447-458, March 1999.

20. I. DEBLED-RENNESON and J.P. REVEILLÈS, *Incremental algorithm for recognizing pieces of digital planes*, Spie's International Symposium on Optical Science, Engineering, and Instrumentation, Vision Geometry V, Denver, USA, August 1996.
21. E. ANDRES, I. DEBLED-RENNESON, C. SIBATA, R-S. ACHARYA, *Application of linear contour segmentation in stereotactic radiosurgery*, poster, Spie's International Symposium on Medical Imaging, Image Processing Conference, California, USA, February 1996.
22. I. DEBLED-RENNESON and J.P. REVEILLÈS, *A new approach to digital planes*, Spie's International Symposium on Photonics for Industrial Applications, Vision Geometry III, Boston, USA, november 1994.
23. I. DEBLED-RENNESON and J.P. REVEILLÈS, *An incremental algorithm for digital plane recognition*, 4th International Conference DGCI'94, Grenoble, September 1994.
24. I. DEBLED-RENNESON and J.P. REVEILLÈS, *A linear algorithm for segmentation of digital curves*, Third International Workshop on Parallel Image Analysis, College Park, University of Maryland, USA, Juny 1994.

Conférences d'audience nationale avec comité de sélection

25. F. TOUZAIN, S. SCHBATH, I. DEBLED-RENNESON, B. AIGLE, P. LEBLOND and G. KUCHEROV, *SIGffRid : Programme de recherche des sites de fixation des facteurs de transcription par approche comparative*, Journées Ouvertes Biologie, Informatique et Mathématiques (JOBIM'05), Lyon, juillet 2005.
26. I. DEBLED-RENNESON, S. TABBONE, L. WENDLING, *Approximation polygonale à partir d'une analyse multi-ordre des points de contour*, CIFED'O4, La Rochelle, France, juin 2004.
27. Y. BOUKHATEM, M.M. MESMOUDI and I. DEBLED-RENNESON, *Sur la reconnaissance des plans discrets*, Rencontre Nationale en Analyse, Géométrie et Informatique (RNA-GI'99), Saïda, Algérie, novembre 1999.

Conférence invitée

28. I. DEBLED-RENNESON, *Discrete lines and segmentation of 2D and 3D curves*, Workshop on Mathematical Aspects of Visual Cognition, Hamburg, Germany, November 2002.

Participation à l'écriture d'un ouvrage collectif

29. G. LARGETEAU-SKAPIN and I. DEBLED-RENNESON, *Outils Arithmétiques pour la*

Géométrie Discrète, Editeurs D. Coeurjolly, A. Montanvert, et JM. Chassery, Géométrie discrète et images numériques, chapitre 2. Hermès, 2007.

30. I. SIVIGNON and I. DEBLED-RENNESON, *Droites et plans discrets*, Editeurs D. Coeurjolly, A. Montanvert, et JM. Chassery, Géométrie discrète et images numériques, chapitre 6. Hermès, 2007.

Posters présentés à des manifestations internationales sans comité de sélection

31. P. LAVIGNE, I. DEBLED-RENNESON, B. AIGLE, P. LEBLOND and G. KUCHEROV, *Identification of transcription factor binding sites in the genome of Streptomyces coelicolor A3(2)*, Journées Post Génomique de la Doua (JPGD2003), Lyon, May 2003.
32. F. TOUZAIN, P. LAVIGNE, I. DEBLED-RENNESON, B. AIGLE, P. LEBLOND and G. KUCHEROV, *Identification of transcription factor binding sites in Streptomyces coelicolor A3(2) by phylogenetic comparison*, European Conference for Computer Biology (ECCB'03), Paris, September 2003.

Communications effectuées à des manifestations sans comité de sélection

33. I. DEBLED-RENNESON, *Droites discrètes, reconnaissance et applications*, Ecole Jeunes Chercheurs Informatique Mathématique, Loria, Nancy, mars 2007.
34. I. DEBLED-RENNESON, *Estimation of tangents to a noisy discrete curve*, Workshop sur le Contenu Informatif des Images Numériques, ENS Cachan, November 2004.
35. H. DÖRKSEN-REITER, I. DEBLED-RENNESON, *Convex and concave parts of digital curves*, Dagstuhl Seminar "Geometric Properties from Incomplete Data", March 2004.
36. I. DEBLED-RENNESON, *Estimation of tangents to a noisy discrete curve*, Dagstuhl Seminar "Geometric Properties from Incomplete Data", March 2004.
37. I. DEBLED-RENNESON et J.P. REVEILLÈS, *Reconnaissance des droites et plans discrets*, 23ème Ecole de Printemps d'Informatique Théorique, Géométrie et Topologie Discrète, Le Lioran, Cantal, mai 1995.
38. I. DEBLED-RENNESON et J.P. REVEILLÈS, *Un algorithme linéaire de polygonalisation de courbes discrètes*, Deuxième colloque de géométrie discrète en imagerie, Grenoble, septembre 1992.

Séminaires invités

39. I. DEBLED-RENNESON, *Courbure discrète et ADN*, séminaire LIRIS, Lyon, octobre 2004.

40. I. DEBLED-RENNESON, *Un algorithme linéaire de reconnaissance de segments flous*, Intervention dans le groupe de travail national en géométrie discrète, ERI, Lyon, juin 2003.
41. I. DEBLED-RENNESON, *Sur la reconnaissance des droites discrètes*, Intervention au séminaire d'informatique fondamentale du LIAFA (<http://www.liafa.jussieu.fr/~semalgo/>), 29 février 2000.
42. I. DEBLED-RENNESON, *Reconnaissance de morceaux de plans discrets*, Intervention dans le groupe de travail national en géométrie discrète, ENS, Paris, 13 mars 2000.

Thèse

43. I. DEBLED-RENNESON, *Etude et reconnaissance des droites et plans discrets*, Thèse. Université Louis Pasteur, Strasbourg I, 1995.

Articles soumis

44. F. TOUZAIN, S. SCHBATH, I. DEBLED-RENNESON, B. AIGLE, P. LEBLOND and G. KUCHEROV, *SIGffRid : a tool to search for sigma factor binding sites in bacterial genomes using comparative approach and biologically driven statistics*, soumis à BMC Bioinformatics en juin 2007.
45. J.-P. SALMON, I. DEBLED-RENNESON, L. WENDLING, *Une nouvelle méthode pour détecter les arcs et les segments à partir de l'analyse du profil de rayons de courbure*, soumis à RFIA'08 en juin 2007.
46. L. PROVOT, I. DEBLED-RENNESON, *Geometric feature estimators for noisy discrete surfaces*, soumis à DGCI'08 en septembre 2007.
47. L. PROVOT, I. DEBLED-RENNESON, *Segmentation of noisy discrete surfaces*, soumis à IWCIA'08 en octobre 2007.

Troisième partie

Sélection d'articles de recherche

Articles de recherche sélectionnés

- A. I. DEBLED-RENNESSON, F. FESCHET and J. ROUYER-DEGLI, *Optimal blurred segments decomposition of noisy shapes in linear time*, Computers & Graphics, 30(1), 2006.
- B. I. DEBLED-RENNESSON, J.-L. REMY and J. ROUYER-DEGLI, *Linear segmentation of discrete curves into fuzzy segments*, Discrete Applied Mathematics, 151 :122-137, October 2005.
- C. I. DEBLED-RENNESSON, S. TABBONE, L. WENDLING, *Multiorder polygonal approximation of digital curves*, Electronic Letters on Computer Vision and Image Analysis, 5(2) :98-110, August 2005.
- D. H. DÖRKSEN-REITER, I. DEBLED-RENNESSON, *Convex and concave parts of digital curves*. In R. Klette, R. Kozera, L. Noakes, and J. Weickert, editors, Geometric Properties from Incomplete Data, volume 31 of Computational Imaging and Vision. Springer-Verlag, January 2005.
- E. I. DEBLED-RENNESSON, J.-L. REMY and J. ROUYER-DEGLI, *Detection of the discrete convexity of polyominoes*, Discrete Applied Mathematics, Volume 125, Issue 1, pp. 115-133, January 2003.

Optimal blurred segments decomposition of noisy shapes in linear time

I. Debled-Rennesson^b F. Feschet^a J. Rouyer-Degli^b

^a*LLAIC - IUT Clermont-Ferrand, BP 86, 63172 Aubière, France.*

^b*LORIA Nancy – Campus Scientifique - BP 239
54506 Vandœuvre-lès-Nancy Cedex, France*

Abstract

Blurred segments were introduced by Debled-Rennesson et al [1,2] as an extension of the arithmetical approach of Reveillès [3] on discrete lines, to take into account noise in digital images. A subclass of blurred discrete segments was introduced in [4] with an optimal linear time algorithm for the recognition of blurred segment in this class. This paper extends the previous algorithm to deal with very noisy curves and to propose a decomposition somewhat preserving the intuitive notion of corners.

Keywords: Blurred segments, convex hulls, noisy shapes decomposition.

1 Introduction

Discrete (also called digital) segments are well known objects which have been thoroughly studied for more than 30 years [5]. There are many definitions of discrete segments, all equivalent for 8-connected discrete sets and 4-connected discrete sets. Discrete segments serve as building blocks for representation [6], decomposition [7] or analysis of discrete curves and more generally shapes. For instance, polygonalizations of discrete curves are widely used in shape representation [6] and can be computed in linear time [8–10]. Moreover, the use of discrete segments permits a perfect representation or reconstruction of discrete curves. However, this might result in complicated representations when discrete curves include noise or have been distorted by an acquisition process.

Email addresses: debled@loria.fr (I. Debled-Rennesson),
feschet@llaic3.u-clermont1.fr (F. Feschet), rouyer@loria.fr (J. Rouyer-Degli).

Many polygonal approximation methods have been proposed throughout the years using different approaches [11–14]. To deal with noise and as an extension of the results presented in [9], the notion of fuzzy segments was introduced in [1,2]. From now on, we shall name these segments blurred segments rather than fuzzy segments in order to prevent any confusion with fuzzy logic and fuzzy geometry. The theorem of Debled-Renesson and Reveillès [9] provides an incremental algorithm with linear-time complexity for the recognition of discrete segments using arithmetical properties and has been extended to the case of blurred segments by Debled-Renesson et al [1]. However, blurred segments represent supersets of the original discrete data and thus, the result obtained with the previous theorem can not be guaranteed to be optimal, in the sense that the orders of the blurred segments are not necessarily minimal. We have presented in [4] a study of the order of blurred segments and point out the reason why it is difficult to minimize the order of a blurred segment in the recognition process. We have also presented theoretical arguments to justify a restriction in the class of blurred segments in order to guarantee optimality in the recognition process. Our new approach can deal with disconnected sets which was impossible with the theorem given in [1].

In the present paper, we recall the construction presented in [4] and propose some extensions to better handle the notion of corner in noisy shapes. The paper is organized as follows. In section 2 we recall definitions and properties used in [1]. We also provide the justification of [4] to motivate the introduction of a subclass of blurred segments by adding a geometric characterization based on convex hulls. To end section 2, we describe the algorithm presented in [4]. In section 3 and 4, we describe further improvements for managing shapes and to handle the notion of noisy corners with application to noisy shapes decomposition. The paper ends up with some conclusions and perspectives in section 5.

2 Blurred Segments

2.1 Definitions

The notion of blurred (also called fuzzy) segments relies on the arithmetical definition of discrete lines [3] where a line, with slope $\frac{a}{b}$, lower bound μ and thickness ω (with a , b , μ and ω being integer such that $\gcd(a, b) = 1$) is the set of integer points (x, y) verifying $\mu \leq ax - by < \mu + \omega$. Such a line is denoted by $\mathcal{D}(a, b, \mu, \omega)$. The real lines $ax - by = \mu + \omega - 1$ and $ax - by = \mu$ are respectively named the *upper and lower leaning lines* of $\mathcal{D}(a, b, \mu, \omega)$ [9]. The integer points (x_L, y_L) (resp. (x_U, y_U)) of the lower (resp. upper) leaning lines of $\mathcal{D}(a, b, \mu, \omega)$ are called the lower (resp. upper) leaning points of $\mathcal{D}(a, b, \mu, \omega)$.

We refer to Fig. 1 for a descriptive example of these notions.

In the following, we restrict our study to points of the first octant of \mathbb{Z}^2 , due to symmetries with respect to Ox , Oy and the real line $x = y$. We thus always have $0 \leq y \leq x$. This hypothesis can be done without loss of generality and simplifies notations, proofs and definitions.

Definition 1 [1] *A set \mathcal{S}_b of consecutive points ($|\mathcal{S}_b| \geq 2$) of an 8-connected curve is a **blurred segment with order d** in the sense of reference [1] if there exists a discrete line $\mathcal{D}(a, b, \mu, \omega)$, called **bounding**, such that all points of \mathcal{S}_b belong to \mathcal{D} and $\frac{\omega}{\max(|a|, |b|)} \leq d$.*

The notion of order of a blurred segment has been introduced to make a difference compared to the thickness of bounding lines of \mathcal{S} , since any sufficiently thick discrete line can contain \mathcal{S} . Thus, two discrete lines containing \mathcal{S} can be compared with respect to their orders and this leads to a classification of the lines containing \mathcal{S} . We will show in the following that this definition is too general to ensure that the parameters of such segments can be effectively computed. To be reasonably close to the points of \mathcal{S} , more restrictive conditions onto the discrete lines containing \mathcal{S} was introduced as follows.

Definition 2 [1] *Let \mathcal{S}_b be a blurred segment with order d whose abscissa interval is $[0, l - 1]$ and let $\mathcal{D}(a, b, \mu, \omega)$ be a bounding line of \mathcal{S}_b . \mathcal{D} is named **strictly bounding for \mathcal{S}_b** if \mathcal{D} possesses at least three leaning points in the interval $[0, l - 1]$ and, \mathcal{S}_b contains at least one lower leaning point and one upper leaning point of \mathcal{D} .*

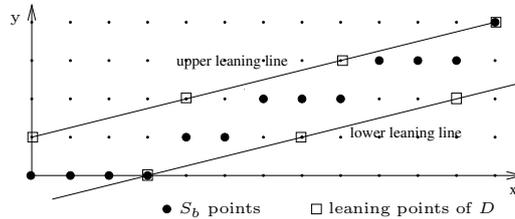


Fig. 1. A strictly bounding line D of a blurred segment \mathcal{S}_b

In Fig. 1, $\mathcal{D}(1, 4, -4, 8)$ is a strictly bounding line of the blurred segment \mathcal{S}_b . The leaning points of \mathcal{D} are the points $(4k + 3, k)$ and $(4k, k + 1)$, for $k \in [0, 3]$ and \mathcal{S}_b contains the lower leaning point $(3, 0)$ and the upper leaning point $(12, 4)$.

2.2 Restriction

A linear time incremental algorithm of segmentation into order d blurred segments was given in [1,2]. The algorithm was incremental and constructed a

strictly bounding line. However the constructions did not guarantee that the order of the built bounding line was minimal. Hence, the algorithm in [2] could lead to over-segmented shapes. This problem was identified in [4]. It is a consequence of the choice of the measure of thickness, that is $\omega / \max(|a|, |b|)$. More precisely, the following theorem was proved in [4].

Theorem 3 *For any finite set \mathcal{S}_b , let us denote by \mathcal{W} the series $(\omega(a_k, b_k)/b_k)$ where k is a positive integer, $a_k = k$ and $b_k = kb_0 + \lambda$, with b_0 and λ positive integers and $w(\cdot)$ the arithmetical width of a line of parameters (a_k, b_k) containing \mathcal{S}_b . Then \mathcal{W} is decreasing and has a limit equals to $\frac{\omega(1, b_0) - 1}{b_0}$.*

Our goal is to construct the longest possible blurred segments whose order is below a given order d . To do so, we must guarantee that the computed order be minimal among all possible parameters for a bounding line. Thanks to the previous theorem, we slightly modified the subclass of considered blurred segments by taking the limit measure of the previous theorem as a measure for comparing blurred segments. This choice directly furnishes the minimal reachable value.

We start by giving a geometric description of the limit measure. The **vertical distance of a discrete line** $\mathcal{D}(a, b, \mu, \omega)$ is the vertical distance (ordinate difference) between the leaning lines of \mathcal{D} and is equal to $\frac{\omega - 1}{b}$. We now recall that a **supporting line** of a convex set C is a line l such that the intersection of l with C is not empty, and such that C is entirely either below or above l [15]. The **vertical distance of a convex set** C , is the minimal vertical distance of any pair of parallel supporting lines. This could also be defined as the maximal length of the intersection of C with a vertical line.

Definition 4 *Let us consider a set of 8-connected points \mathcal{S}_b . A bounding line of \mathcal{S}_b is said **optimal** if its vertical distance is minimal, i.e. if the vertical distance of \mathcal{S}_b equals to the vertical distance of its convex hull $\text{conv}(\mathcal{S}_b)$.*

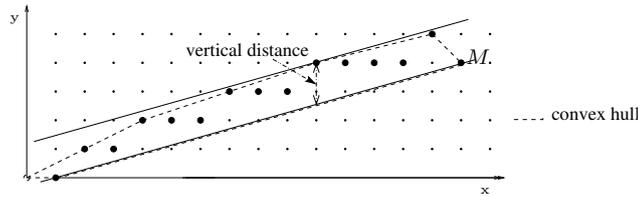


Fig. 2. An optimal bounding line

This definition is illustrated in Fig. 2 and leads to the following new definition concerning blurred segments.

Definition 5 *A set \mathcal{S}_b is a **blurred segment of width** ν if and only if its optimal bounding line has a vertical distance less or equal to ν .*

The recognition of blurred segments with width ν is thus equivalent to the computation of the vertical distance of the convex set $conv(\mathcal{S}_b)$. This problem was identified in [4] as a problem closely related to the the computation of the *width* of a convex set [16]. As it is used in [15] rotating calipers algorithm, three positions of optimality were identified: (edge,vertex), (vertex,edge) or (vertex,vertex) where the first element represents the lower part of the convex set and the second element represents the upper part. The edges corresponding to the position of a maximum of $\mathcal{V}(\cdot)$ are supporting lines, taking horizontal lines for a couple (vertex,vertex). Moreover, the edge and its parallel passing through the vertex define exactly the lower and upper leaning lines of an optimal bounding line for C . Hence, optimal lines are deduced from the positions of maxima of the vertical width of the convex sets of points of the boundary of the original shapes.

To obtain an incremental linear time algorithm, Melkman's algorithm [17] was used to maintain the convex hull. Moreover, points were added with increasing x to guarantee that computations were done on a simple polyline. We proved in [4] that the new optimal position could be found by moving along a part of the convex hull. This part corresponded to the portion of the convex hull at the opposite of the added new point. Moreover, the computations were always performed on disjoint parts of the dynamic convex hull such that the total complexity of the algorithm was linear in time and space. A sketch of the algorithm is given below where L and U represent point of the convex hull defining the current position of extremality. U is always on the upper part and L is always on the lower part of the convex hull. Moreover when using the notation X_L (X being U or L), L means *last* which corresponds to the X point the closer to the new added point M . It should be noticed that due to symmetry, the vertical width of the convex hull corresponds to the horizontal width depending on the octant of the computation. More generally, we can maintain the vertical and horizontal width simultaneously and always choose the minimal value as the width of the convex set. For clarity, the algorithm is presented in the first octant only.

Input: S an 8-connected sequence of integer points, ν a real value

Output: $isSegment$ a boolean value, a, b, μ, ω integers

Initialization: $isSegment = true$, $a = 0$, $b = 1$, $\omega = b$, $\mu = 0$, $M = (x_0, y_0)$

while (S is not entirely scanned and $isSegment$)

$M =$ next point of S ;

add M to the upper and lower convex hulls of the scanned part of S ;

$r = ax_M - by_M$;

if ($r = \mu$) **then** $U_L = M$;

if ($r = \mu + \omega - 1$) **then** $L_L = M$;

if ($r \leq \mu - 1$) **then**

$U_L = M$;

Let N the point before M in the upper convex hull,
 $a_0 = y_M - y_N$, $b_0 = x_M - x_N$,
 $a = \frac{a_0}{\gcd(a_0, b_0)}$, $b = \frac{b_0}{\gcd(a_0, b_0)}$, $\mu = ax_M - by_M$;
Find the first point C in the lower part of the convex hull starting at L_L ,
such that : slope of $[C, Cnext] > \frac{a}{b}$;
 $L_L = C$;
 $\omega = ax_{L_L} - by_{L_L} - \mu + 1$;
else
if ($r \geq \mu + \omega - 1$) **then** *symmetrical case*
end if
 $isSegment = \frac{\omega - 1}{b} \leq \nu$;
end

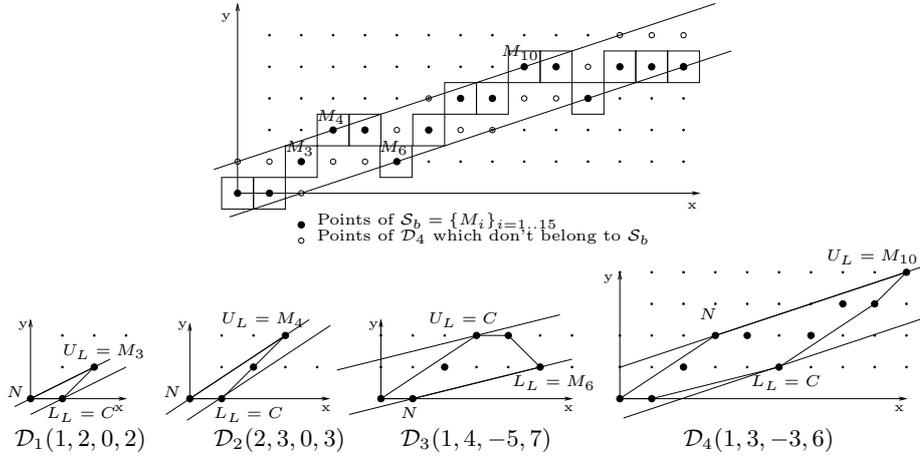


Fig. 3. An example of width 2 blurred segment incremental recognition

An example of the algorithm processing is depicted in Fig. 3. Part of a discrete curve, \mathcal{S}_b , is drawn at the top of the figure and the different optimal bounding lines, obtained during the incremental recognition, are given. As it can be seen, there are only four steps in the recognition process and the slopes of the supporting lines decrease or increase with respect to the added points. \mathcal{S}_b is a blurred segment of width 2 with $\mathcal{D}_4(1, 3, -3, 6)$ as optimal bounding line.

2.3 Blurred polygonal decomposition

From the previous recognition algorithm, a blurred polygonal decomposition algorithm can be built where the width ν is fixed in advance. The principle is as follows: start at some point of the curve and perform a recognition with width ν . The last point of the recognition is taken to be the first point of computation of the next recognition until all points of the curve have been examined. The resulting algorithm is also an incremental linear time algorithm (see [2] for full details about the algorithm).

In Fig. 4, we show how the curves on the left are segmented into blurred segments with width 2. The curve in Fig. 4 (top) is segmented into 18 segments. The curve in Fig. 4 (bottom) is a noisy version of the top one; the segmentation gives more segments: 21. If we increase the width of the blurred segments, the number of segments decreases: 19 segments for the width 2.5, 18 segments for the width 3.

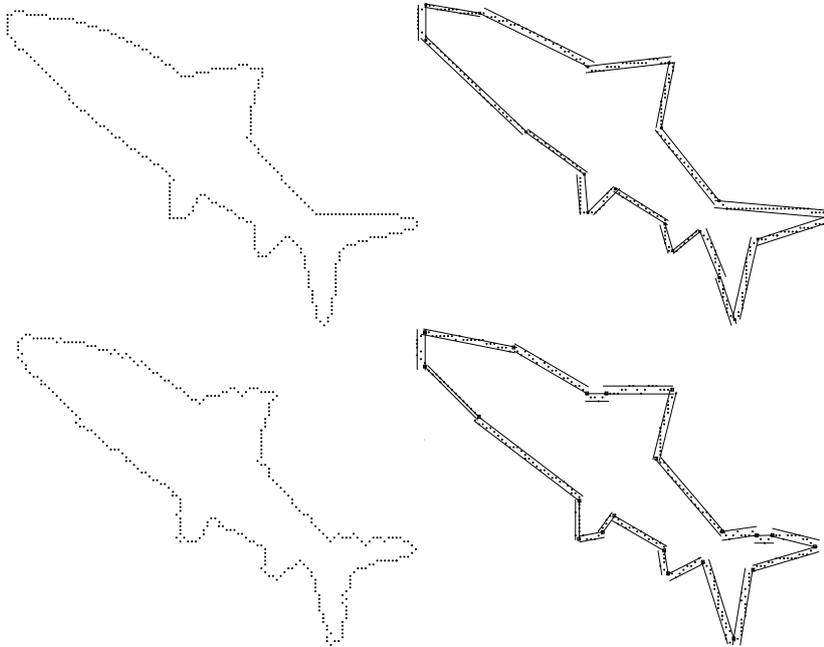


Fig. 4. Shark curves (top without noise, bottom with noise) decomposed into blurred segments of width 2

It should be noticed that in the previous experiments, the curve being 8-connected and one pixel-wide, we have worked with the Freeman codes as input to our algorithm.

3 Improvement

3.1 The problem

When the curves to be segmented are too noisy, the maximum width to set in the blurred polygonal decomposition algorithm described previously must be high enough to get a significant number of segments. Then we can see shifts at the extremity points between the obtained segments and the expected ones (c.f. figure 5). The expected corners are not well located. We give in this section a proposal to overcome this problem. We refer to the thesis of Marji [18] for the comparison of various algorithm for corner detection.

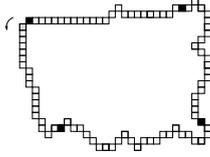


Fig. 5. A blurred segmentation with width 4. The grey point is the starting point and the black points are the extremity points of the blurred segments.

3.2 Post-processing proposal

After using the blurred polygonal decomposition algorithm on a discrete curve, we propose a post-processing which permits to shift the obtained extremities in order to improve the localization of the corners.

Our algorithm provides as results not only the extremity points of the blurred segments but also the characteristics of the optimal bounding lines. We can use these parameters to modify the positions of the detected extremities of the blurred segments.

The idea is as follows: for two consecutive blurred segments \mathcal{S}_i and \mathcal{S}_{i+1} , with optimal bounding lines $\mathcal{D}_i(a_i, b_i, \mu_i, \omega_i)$ and $\mathcal{D}_{i+1}(a_{i+1}, b_{i+1}, \mu_{i+1}, \omega_{i+1})$, and with the corresponding extremity point E_{i+1} , we consider the real lines d_i and d_{i+1} located between (at the middle of) the leaning lines of \mathcal{D}_i and \mathcal{D}_{i+1} and with respectively the same slope. It should be noticed that d_i and d_{i+1} are in fact rational lines since their slope coefficients are integers and their shifts are half-integers. Due to this property, no error are introduced and the computation can be all be made with rational numbers only. The intersection point, I_{i+1} , between these lines d_i and d_{i+1} is then considered. If the two lines d_i and d_{i+1} do not belong to the same octant, then the closer point of the curve to I_{i+1} becomes the new extremity point and replaces the point E_{i+1} . When the variation of the slopes between d_i and d_{i+1} is low, care must be taken with their intersection. Indeed, it is not obvious to see where will be the intersection point I_{i+1} . To overcome this problem, we do not apply the post-processing when the variation of the slopes is below a given threshold. This is why some corners are not modified by our post-processing.

An illustration of the used technique is proposed at the figure 6. The extremity points obtained after a blurred polygonal decomposition with width 4 are : $E_1(0, 0)$, $E_2(6, 16)$, $E_3(28, 15)$, $E_4(25, -3)$ and $E_5(1, -1)$. The obtained optimal bounding lines are $\mathcal{D}_1(-8, -3, 0, 27)$, $\mathcal{D}_2(0, -1, 15, 5)$, $\mathcal{D}_3(17, 2, 431, 67)$, $\mathcal{D}_4(-1, 10, -3, 19)$. We consider the real lines $d_1 : -8x + 3y = 13$, $d_2 : y = 17$, $d_3 : 17x - 2y = 464$ and $d_4 : -x - 10y = 6$. The computed intersection points are $I_2(4.75, 17)$, $I_3(29.29, 17)$ and $I_4(26.9, -3.29)$. Therefore, the new extremity points are the points of the curve $E_2(4, 17)$, $E_3(29, 17)$ and $E_4(27, -4)$.

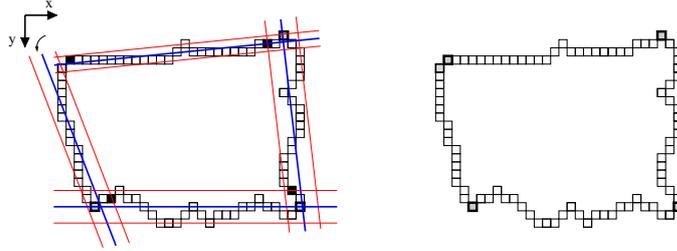


Fig. 6. Example of post-processing: in black the extremity points obtained after a blurred decomposition with width 4. On the right hand side, the new extremity points.

This technique is used in the figure 7 after a blurred polygonal decomposition with width 3.9. We can see improvements for the detection of corners of the shark tail where the curve inflections are rather strongly. However the areas, where the slope of the optimal bounding lines of detected blurred segments are close, are not modified.

To ensure better improvement, we propose to use this post-processing within a multi-scale analysis presented in [19]. Since this analysis is based on blurred segments, it applies straightforwardly to our new recognition algorithm. It was already mentioned in [19] that the extremities of the blurred segments might move when the width increases. With the post-processing, we can regularize those shifts of the extremities and obtain a better multi-scale analysis of the extremities. A validation of this process is currently under progress.

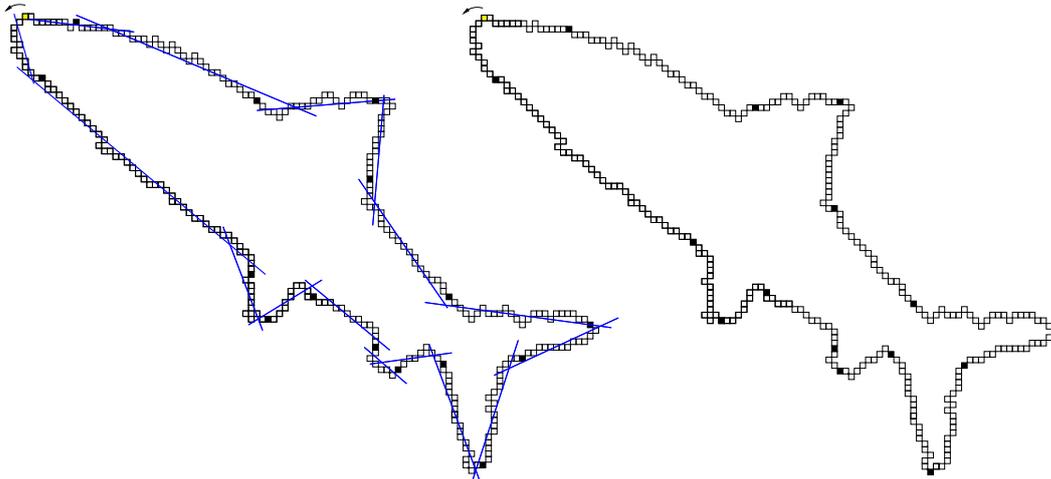


Fig. 7. Example of post-processing: in black the extremities points obtained after blurred decomposition with width 3.9. On the right hand side, the new extremity points.

4 Extensions

Several extensions might be provided to the original algorithm. First, it should be noticed that it never supposes that the points are connected. Hence, without any modification, the given algorithm works for disconnected sets. The main problem to solve for disconnected sets is the ordering of the points if the digital set is only known as a unorganized set of points. One strategy [20] consists in computing a complete graph between the points and to label each edge with the distance (Euclidean or discrete) between the two vertices of the edge. A minimum spanning tree is then computed on the graph to build the ordering of the points.

Second is the problem of the dynamic maintenance of the convex hull. We add the hypothesis that the points were added with increasing x , which restrict the application of the algorithm. Several solutions exist in that case. The most complete one is the full dynamic algorithm of Brodal and Jacob [21]. The maintenance of the width could be deduced from their study. A naive algorithm as given in [22] has a $O(\log^2 n)$ update complexity. Another approach, since only the width is of interest, is to use the fully dynamic approach of Chan [23]. The only requirement is that the number of points of the shapes must be known in advance. This point is not a crucial point in practice since usually boundaries of shapes are obtained via a contour tracing algorithm. The complexity of Chan's algorithm is $O(\sqrt{n} \log^3 n)$ per update. Since our problem is restricted to insertion only, a complexity of $O(\log n)$ per update is achievable [22].

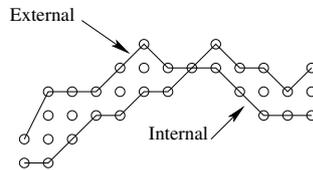


Fig. 8. The external and internal contours associated to a thick digital shape

The third extension is to better manage thick curves. Indeed, usually thick curves are described by a list of their points and so this imposes an arbitrary ordering between points with same x or same y coordinates. Our algorithm takes a list as input. Hence, the algorithm stops as soon as the threshold order cannot be satisfied anymore. If we now consider the external and internal boundary associated to a thick curve (see Fig. 8), it appears that our algorithm might stop its recognition process inside points with the same x coordinates. This could be undesirable since all those points can be considered as indiscernable. Indeed, in the case of thick curves, applications might require that the section of the curves are processed in a whole and not sequentially pixel by pixel. Since our algorithm only used convex hulls, all internal points of the contour can be avoided. In such case, we only add the points of the external and internal boundaries. The recognition process then stops if both

points cannot be added. A prerequisite of this modification is to fix the width ν at a higher value than the thickness of the curve in order to ensure a global decomposition.

5 Conclusion

We have pursued in this paper the study of blurred segments and their applications to discrete noisy curves. Previous works of Debled-Rennesson et al [1] were not able to decompose discrete noisy curves with minimally thin blurred segments. We identified the origin of this drawback and proposed to restrict the class of blurred segments by adding a geometrical bound on their thickness in [4]. We have presented an incremental linear time algorithm which solves this problem with minimality in the thickness of constructed blurred segments. In this paper, we propose a post-processing to manage the localization of corners in noisy shapes. We are currently investigating the multi-scale consequences of our post-processing with applications in polygonal simplification. The various extensions of our algorithm are also under progress and results will be reported in a forthcoming paper.

References

- [1] I. Debled-Rennesson, J.-L. Rémy, J. Rouyer, Segmentation of discrete curves into fuzzy segments, in: 9th IWCIA, Vol. 12 of Electronic Notes in Discrete Mathematics, 2003.
- [2] I. Debled-Rennesson, J.-L. Rémy, J. Rouyer, Segmentation of discrete curves into fuzzy segments, extended version, Tech. rep., INRIA Report RR-4989, <http://www.inria.fr/rrrt/rr-4989.html> (2003).
- [3] J. Reveillès, Géométrie discrète, calculs en nombres entiers et algorithmique, Thèse d'Etat – Université Louis Pasteur (1991).
- [4] I. Debled-Rennesson, F. Feschet, J. Rouyer, Optimal blurred segments decomposition in linear time, in: E. Andres, G. Damiand, P. Lienhardt (Eds.), 12th International Conference DGCI, no. 3429 in LNCS, Springer-Verlag, 2005, pp. 371–382.
- [5] A. Rosenfeld, R. Klette, Digital Straightness – a review, *Discrete Applied Mathematics* 139 (1–3) (2004) 197–230.
- [6] D. Zhang, G. Lu, Review of shape representation and description techniques, *Pattern Recognition* 37 (1) (2004) 1–19.

- [7] F. Feschet, L. Tougne, Optimal time computation of the tangent of a discrete curve: application to the curvature, in: 8th International Conference DGCI, no. 3429 in LNCS, Springer Verlag, 1999, pp. 31–40.
- [8] A. Smeulders, L. Dorst, Decomposition of discrete curves into piecewise straight segments in linear time, *Contemporary Mathematics* 119 (1991) 169–195.
- [9] I. Debled-Renesson, J. Reveillès, A linear algorithm for segmentation of digital curves, *IJPRAI* 9 (6).
- [10] M. Lindenbaum, A. Bruckstein, On Recursive, $O(N)$ Partitioning of a Digitized Curve into Digital Straight Segments, *IEEE Transactions on PAMI* 15 (9) (1993) 949–953.
- [11] J. Perez, E. Vidal, Optimum polygonal approximation of digitized curves, *Pattern Recognition Letter* 15 (1994) 743–750.
- [12] P. L. Rosin, Techniques for assessing polygonal approximations of curves, *IEEE Transactions on PAMI* 19 (6) (1997) 659–666.
- [13] P. Yin, A tabu search approach to polygonal approximation of digital curves, *IJPRAI* 14 (2) (2000) 243–255.
- [14] A. Kolesnikov, P. Fränti, Reduced-search dynamic programming for approximation of polygonal curves, *Pattern Recognition Letter* 24 (14) (2003) 2243–2254.
- [15] M. Houle, G. Toussaint, Computing the width of a set, *IEEE Trans. on PAMI* 10 (5) (1988) 761–765.
- [16] F. Preparata, M. Shamos, *Computational Geometry: an Introduction*, Springer-Verlag, 1985.
- [17] A. Melkman, On-line Construction of the Convex Hull of a Simple Polygon, *Information Processing Letters* 25 (1987) 11–12.
- [18] M. Marji, On the detection of dominant point on digital planar curves, Ph.D. thesis, Wayne State University (2003).
- [19] I. Debled-Renesson, S. Tabbone, L. Wendling, Fast Polygonal Approximation of Digital Curves, in: *ICPR*, Vol. 1, 2004, pp. 456–468.
- [20] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, W. Stuetzle, Surface reconstruction from unorganized points, in: *Computer Graphics Proceedings, Annual Conference Series*, 1992, pp. 295–302.
- [21] G. Brodal, R. Jacob, Dynamic planar convex hull, in: *43rd Annual Symposium on Foundations of Computer Science*, 2002.
- [22] L. Buzer, An elementary algorithm for digital line recognition in the general case, in: E. Andres, G. Damiand, P. Lienhardt (Eds.), 12th International Conference DGCI, no. 3429 in LNCS, Springer-Verlag, 2005, pp. 299–310.
- [23] T. Chan, A fully dynamic algorithm for planar width, *Discrete and Computational Geometry* 30 (2003) 17–24.

Linear Segmentation of Discrete Curves into Blurred Segments

Isabelle Debled-Rennesson ^{*,1}, Jean-Luc Rémy ²,
Jocelyne Rouyer-Degli ³

LORIA

(Laboratoire Lorrain de Recherche en Informatique et ses Applications)
Campus Scientifique, B.P. 239, F54506 Vandœuvre-lès-Nancy, FRANCE

Abstract

The concept of blurred segment is introduced, it allows the flexible segmentation of discrete curves, taking into account noise. Relying on an arithmetic definition of discrete lines [7], it generalizes such lines, admitting that some points are missing. Thus, a larger class of objects is considered. A very efficient linear detection algorithm for blurred segments and its application to curve segmentation are presented.

Key words: Segmentation, Blurred Segment, Discrete Line, Noisy Curve

1 Introduction

Numerous techniques for the segmentation of planar discrete curves have been proposed for the last thirty years (such as [2,8,10]). Some of them are based on the principle of polygonal approximation where the curve is split into a sequence of straight line segments. Several authors [5,3] proposed linear algorithms for the curve segmentation into exact discrete straight lines, based on precise mathematical definitions of discrete straight lines.

However, in order to fulfil the need of an approximate segmentation of discrete curve taking into account noises due to data processing operations, such as skeletonization in the case of image data, we present in this paper the notion of

* Corresponding author.

Email address: debled@loria.fr.

¹ Institut National de Recherche en Informatique et en Automatique

² Centre National de la Recherche Scientifique

³ Université Henri Poincaré, Nancy 1

blurred segment, which relies on an arithmetical definition of discrete straight lines [7] where thickness may be parameterized.

The discrete arithmetical geometry has been introduced by J.-P. Reveillès [7], it enables a better formalisation of discrete geometry problems. Indeed discrete objects such as straight lines, circles, planes are defined thanks to analytical expressions with integer numbers only. Arithmetic and the theory of numbers allow to obtain some properties of these objects then to deduce very performing incremental algorithms for the construction, the localisation and the recognition [5,3,9,1]. The work presented in this paper belongs to this framework.

A blurred segment is an 8-connected sequence of points which belong to a discrete straight line with a given thickness. A parameter, the order of a blurred segment, permits to control the amplitude of the authorized noise by fixing the thickness of the straight line bounding the blurred segment. Adding a point to a blurred segment is translated into the calculation of the slope and thickness of a new bounding straight line. It corresponds to very easy calculations. It leads to an incremental and very efficient linear algorithm for splitting discrete curves into blurred segments with fixed order.

In Section 2, after recalling definitions and basic properties of discrete straight lines, we define the related notion of blurred segment and bounding straight line. Then, in Section 3, a fundamental theorem on the growing of a blurred segment is proved. It leads to the incremental and linear algorithm for the recognition of a blurred segment with a fixed order, as detailed in Section 4. At last, the linear algorithm for the segmentation of a curve into blurred segments with fixed order is presented and illustrated by a few examples.

2 Definitions and first properties

2.1 Discrete lines

In this section, we briefly recall some results of [7] and [3] that we shall need.

Definition 1 A discrete line [7], named $\mathcal{D}(a, b, \mu, \omega)$, is the set of integer points (x, y) verifying the inequalities $\mu \leq ax - by < \mu + \omega$ where a, b, μ, ω are integers. $\frac{a}{b}$ with $b \neq 0$ and $\gcd(a, b) = 1$ is the slope of the discrete line, μ is named lower bound and ω arithmetical thickness.

Among the discrete lines we shall distinguish, according to their topology [7]:

- the *naive lines* which are 8-connected and for which the thickness ω verifies $\omega = \max(|a|, |b|)$,
- the **-connected lines* for which the thickness ω verifies $\max(|a|, |b|) < \omega <$

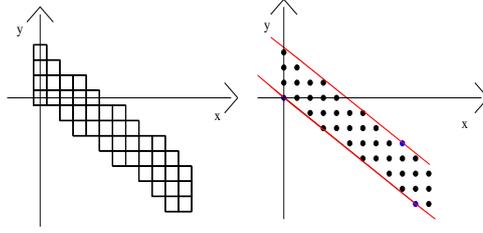


Fig. 1. On the left hand side a representation by pixels (each integer point is represented by a square centered at the point) of a segment of the thick line $\mathcal{D}(7, -10, 0, 34)$ whose equation is $0 \leq 7x + 10y < 34$, for $x \in [0, 10]$, on the right hand side the points of this line are represented by disks to get a better visualisation of the leaning lines.

- $|a| + |b|$,
- the discrete lines said *standard* where $\omega = |a| + |b|$, this thickness is the smallest one for which the discrete line is 4-connected,
- the *thick lines* where $\omega > |a| + |b|$, they are 4-connected

Definition 2 Real straight lines $ax - by = \mu$ and $ax - by = \mu + \omega - 1$ are named the leaning lines of the discrete line $\mathcal{D}(a, b, \mu, \omega)$. An integer point of these lines is named a leaning point (see Fig. 1).

The leaning line located above (resp. under) \mathcal{D} in the first quadrant ($0 \leq a$ and $0 \leq b$) respects the following equation $ax - by = \mu$ (resp. $ax - by = \mu + \omega - 1$), it is named *upper leaning line* (resp. *lower leaning line*) of \mathcal{D} , noted d_U (resp. d_L).

Definition 3 Let $M(x_M, y_M)$ be an integer point, the remainder at the point M as a function of $\mathcal{D}(a, b, \mu, \omega)$, noted $r(M)$, is defined by:

$$r(M) = ax_M - by_M$$

To simplify the writing, we shall suppose hereafter that *the slope coefficients verify* $0 \leq a \leq b$ which corresponds to the first octant.

Proposition 4 Let $\mathcal{D}(a, b, \mu, \omega)$ be a discrete straight line. For each integer k , there exists exactly one integer point P_k whose coordinates (x_{P_k}, y_{P_k}) satisfy both conditions:

$$\begin{cases} r(P_k) = k \\ 0 \leq x_{P_k} \leq b - 1 \end{cases}$$

Proof (Existence) As a and b are relatively prime between them, according to Bezout's theorem, there exist x, y such that $ax - by = 1$. Therefore, the integers kx (noted x') and ky (noted y') satisfy: $ax' - by' = k$. Let x'' be the

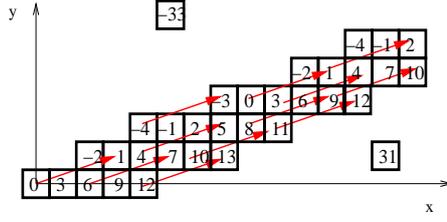


Fig. 2. Representations of the positive shift vector of the line $\mathcal{D}(3, 8, -4, 18)$, the remainder value is indicated on each pixel of the segment. Moreover two exterior points are represented, the point $(5, 6)$ is 29-exterior to \mathcal{D} and the point $(13, 1)$ is 18-exterior to \mathcal{D} .

remainder of the Euclidian division of x' by b . By definition, x'' satisfies the inequalities $0 \leq x'' \leq b - 1$ and there is an integer q such that $x' - x''$ is equal to qb . Let $y'' = y' - aq$. The point P , with coordinates (x'', y'') , satisfies the required conditions.

(Uniqueness) Let $P' = (x', y')$ and $P'' = (x'', y'')$ be two points satisfying the required conditions. Let us prove that $P' = P''$ or, equivalently, that $x' = x''$. Let $x = x' - x''$ and $y = y' - y''$. By hypothesis, x satisfies $-b < x < b$. Moreover, by subtraction, $ax - by = 0$. So b divides ax and, as a and b are relatively prime between them, b divides x and therefore, necessarily, $x = 0$. \square

Definition 5 We call positive shift vector (resp. negative shift vector) of $\mathcal{D}(a, b, \mu, \omega)$, noted V_{+1} (resp. V_{-1}), a vector between two points of \mathcal{D} for which remainder values are different from $+1$ (resp. -1) and such that $0 \leq x_{V_{+1}} \leq b$ (resp. $0 \leq x_{V_{-1}} \leq b$) (see Fig. 2).

Definition 6 An integer point M is k -exterior to \mathcal{D} if $r(M) = \mu - k$ or $r(M) = \mu + \omega + k - 1$ with k being a strictly positive integer. If $k > 1$, this point is named strongly exterior to \mathcal{D} and if $k = 1$, it is named weakly exterior to \mathcal{D} (see Fig. 2).

Proposition 7 Let us suppose that $a > 0$, the vertical distance (ordinate difference) between the leaning lines of a line $\mathcal{D}(a, b, \mu, \omega)$ is equal to $\frac{\omega-1}{b}$ and the horizontal distance (abscissa difference) between the leaning lines is equal to $\frac{\omega-1}{a}$.

Proof Let us consider a point $M(x, y_M)$ on the upper leaning line and the point $N(x, y_N)$ on the lower leaning line. By definition $ax - by_M = \mu$ and $ax - by_N = \mu + \omega - 1$, and therefore $y_M - y_N = \frac{\omega-1}{b}$. The reasoning is similar for the horizontal distance. \square

Definition 8 [7] The vertical (resp. horizontal) steps of $\mathcal{D}(a, b, \mu, \omega)$ are the segments obtained by intersecting \mathcal{D} and the vertical line whose equation is $x = k$ (resp. horizontal one whose equation is $y = k$), where k is integer.

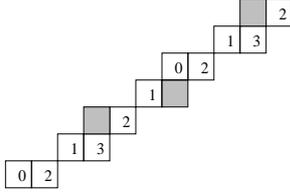


Fig. 3. In white, a blurred segment with order $\frac{4}{3}$, in gray the pixels which do not belong to the segment but which are contained in the bounding line $\mathcal{D}(2, 3, 0, 4)$, the value $2x - 3y$ is indicated on the pixels of the blurred segment.

Proposition 9 [7] *The lengths of the vertical steps are the consecutive integer values $\lceil \frac{\omega}{b} \rceil$ and $\lceil \frac{\omega}{b} \rceil + 1$ if b does not divide ω and the integer $\frac{\omega}{b}$ if b divides ω .*

As well, the lengths of the horizontal steps are the consecutive integer values $\lceil \frac{\omega}{a} \rceil$ and $\lceil \frac{\omega}{a} \rceil + 1$ if a does not divide ω and the integer $\frac{\omega}{a}$ if a divides ω .

The value $\frac{\omega}{b}$ will allow us to define the notion of blurred segment introduced in the next paragraph.

2.2 The blurred segments

Definition 10 *An 8-connected digital set $\mathcal{C} \subseteq \mathbb{Z}^2$ is called a digital 8-curve if each point $P \in \mathcal{C}$ has exactly two 8-neighbors in \mathcal{C} with the possible exception of at most two points, the so-called end points of the curve, having exactly one neighbor in \mathcal{C} .*

Definition 11 *A set $\mathcal{S}f$ of consecutive points ($|\mathcal{S}f| \geq 2$) of a digital 8-curve is a blurred segment with order d if there is a discrete line $\mathcal{D}(a, b, \mu, \omega)$ such that all points of $\mathcal{S}f$ belong to \mathcal{D} and $\frac{\omega}{\max(|a|, |b|)} \leq d$. The line \mathcal{D} is said bounding for $\mathcal{S}f$.*

The order of a blurred segment allows to limit the thickness of the discrete line framing the 8-connected sequence of points of the blurred segment and, so doing, to control the length of vertical steps of the bounding line. In order to be reasonably close to the points of the blurred segment, we introduce more restrictive conditions to the discrete line with the notion of strictly bounding line as defined hereafter.

Definition 12 *Let $\mathcal{S}f$ be a blurred segment whose order is d , and whose the abscissa interval is $[0, l - 1]$, $\mathcal{D}(a, b, \mu, \omega)$ a bounding line of $\mathcal{S}f$, \mathcal{D} is named strictly bounding for $\mathcal{S}f$ if:*

- \mathcal{D} possesses at least three leaning points in the interval $[0, l - 1]$,

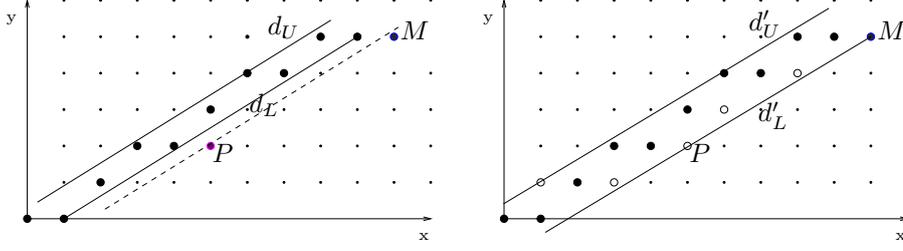


Fig. 4. On the left hand side, a blurred segment $\mathcal{S}f$ whose order is 1, the line $\mathcal{D}(5, 8, -2, 8)$ is strictly bounding and we add the point $M(10, 5)$ to it, $r_{\mathcal{D}}(M) = 10$. P is the point from the interval $[0, 7]$ such that $r_{\mathcal{D}}(P) = 9$. On the right hand side, the line $\mathcal{D}'(3, 5, -2, 7)$ is strictly bounding for $\mathcal{S}f \cup \{M\}$ whose slope is calculated from the vector PM , in black the points of $\mathcal{S}f \cup \{M\}$ and in white the points of \mathcal{D}' which do not belong to $\mathcal{S}f \cup \{M\}$.

- $\mathcal{S}f$ contains at least one lower leaning point and one upper leaning point of \mathcal{D} .

3 Growth of a blurred segment

We present in this section a result which allows to control the growth of a blurred segment. The problem is as follows: let $\mathcal{S}f$ be a blurred segment whose order is d , whose interval on the x axis is $[0, l - 1]$, and let $\mathcal{D}(a, b, \mu, \omega)$ be a strictly bounding line for $\mathcal{S}f$. The point M , connected to $\mathcal{S}f$, whose abscissa x_M is equal to l or $l - 1$, is added to $\mathcal{S}f$. Is the line $\mathcal{D}(a, b, \mu, \omega)$ strictly bounding for $\mathcal{S}f \cup M$ and if not, how can we determine a strictly bounding line? A solution is given by the theorem hereafter whose principle is the following. Two cases are possible:

- $M \in \mathcal{D}$, in that case \mathcal{D} is strictly bounding for $\mathcal{S}f \cup M$,
- $M \notin \mathcal{D}$, let us suppose that $r(M) \geq \mu + \omega$ (symmetrical case if $r(M) < \mu$), M is then located under the lower leaning line of \mathcal{D} , the idea consists in thickening if necessary the line \mathcal{D} so that the lower leaning line is the line whose equation is $ax - by = r(M) - 1$, then to take as new slope the one obtained from the vector \overrightarrow{PM} with P the point of this line whose abscissa verifies $x_P \in [0, b - 1]$, the thickness of this new line will be calculated from the last upper leaning point of \mathcal{D} present in $\mathcal{S}f$ (see illustrated Figure 4). The vector \overrightarrow{PM} is calculated from the shift vectors of \mathcal{D} (see details in Remark 14).

The principle used here is different from the one of naive lines recognition [3], indeed, at each step, the slope of the bounding line may change but its thickness too. Moreover, all the points of the line in the considered interval do not necessarily belong to the blurred segment. When the added point M is

1-exterior to \mathcal{D} , the thickening step does not exist, the point of the first period to be considered for the calculation of the new slope is on the lower leaning line of \mathcal{D} .

Theorem 13 *Let us consider a blurred segment $\mathcal{S}f$ whose interval on the x axis is $[0, l - 1]$, $\mathcal{D}(a, b, \mu, \omega)$ a strictly bounding line. In this case the order of $\mathcal{S}f$ is $\frac{\omega}{b}$. Let $M(x_M, y_M)$ be an integer point connected to $\mathcal{S}f$ whose abscissa is equal to l or $l - 1$.*

- (i) *If $\mu \leq r(M) < \mu + \omega$, then $M \in \mathcal{D}$.
 $\mathcal{S}f \cup M$ is a blurred segment whose order is $\frac{\omega}{b}$ with \mathcal{D} as strictly bounding line.*
- (ii) *If $r(M) \leq \mu - 1$, then M is exterior to \mathcal{D} .
 $\mathcal{S}f \cup M$ is a blurred segment whose order is $\frac{\omega'}{b'}$ and the line $\mathcal{D}'(a', b', \mu', \omega')$ is strictly bounding, with*
- b' and a' coordinates of the vector $\overrightarrow{P_{r(M)+1}M}$, $P_{r(M)+1}$ being the point whose remainder is $r(M) + 1$ with regard to \mathcal{D} and $x_{P_{r(M)+1}} \in [0, b - 1]$,
 - $\mu' = a'x_M - b'y_M$
 - $\omega' = a'x_{L_L} - b'y_{L_L} - \mu' + 1$, with $L_L(x_{L_L}, y_{L_L})$ last lower leaning point of the line \mathcal{D} present in $\mathcal{S}f$.
- Moreover, the shift vectors of \mathcal{D}' are:*
- $$\overrightarrow{V'_{-1}} = (b' - b, a' - a) + \left[\frac{b}{b'}\right](b', a') \text{ and } \overrightarrow{V'_{+1}} = (b, a) - \left[\frac{b}{b'}\right](b', a')$$

- (iii) *If $r(M) \geq \mu + \omega$, then M is exterior to \mathcal{D} .
 $\mathcal{S}f \cup \{M\}$ is a blurred segment whose order is $\frac{\omega'}{b'}$ and the line $\mathcal{D}'(a', b', \mu', \omega')$ is strictly bounding with*
- b' and a' coordinates of the vector $\overrightarrow{P_{r(M)-1}M}$, $P_{r(M)-1}$ being the point whose remainder is $r(M) - 1$ with regard to \mathcal{D} and $x_{P_{r(M)-1}} \in [0, b - 1]$,
 - $\mu' = a'x_{U_L} - b'y_{U_L}$ with $U_L(x_{U_L}, y_{U_L})$ last upper leaning point of the line \mathcal{D} present in $\mathcal{S}f$,
 - $\omega' = a'x_M - b'y_M - \mu' + 1$.
- Moreover, the shift vectors of \mathcal{D}' are:*
- $$\overrightarrow{V'_{-1}} = (b, a) - \left[\frac{b}{b'}\right](b', a') \text{ and } \overrightarrow{V'_{+1}} = (b' - b, a' - a) + \left[\frac{b}{b'}\right](b', a')$$

Remark 14

1. In the cases (ii) and (iii) of the theorem, the new slope is calculated from vectors $\overrightarrow{P_{r(M)+1}M}$ and $\overrightarrow{P_{r(M)-1}M}$. Thanks to the shift vectors of \mathcal{D} , they can be very easily calculated as follow:

$$\overrightarrow{P_{r(M)-1}M} = \left[\frac{x_M - x_{\overrightarrow{V_{+1}}}}{b}\right](b, a) + \overrightarrow{V_{+1}} \text{ and } \overrightarrow{P_{r(M)+1}M} = \left[\frac{x_M - x_{\overrightarrow{V_{-1}}}}{b}\right](b, a) + \overrightarrow{V_{-1}}.$$

2. When $r(M) = \mu - 1$, the point M is weakly exterior to \mathcal{D} and the point

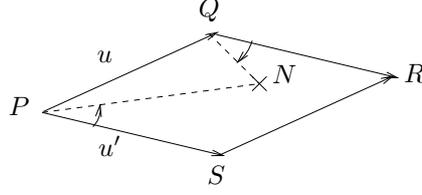


Fig. 5.

$P_{r(M)+1}$ is the first upper leaning point of the line \mathcal{D} in the interval $[0, l - 1]$. As well, when $r(M) = \mu + \omega$, the point M is weakly exterior to \mathcal{D} and the point $P_{r(M)-1}$ is the first lower leaning point of the line \mathcal{D} in the interval $[0, l - 1]$.

3. Adding a point M to a blurred segment $\mathcal{S}f$ thanks to the theorem 13 permits to obtain a strictly bounding line but this one is not always the closest to the points of $\mathcal{S}f \cup \{M\}$. Indeed, for an order which is strictly greater than 1, the characteristics of the strictly bounding line of the obtained blurred segment are not always such that $\frac{\omega'}{\nu}$ is minimal. It is due to the choice of the point P which, in theorem 13, assures that we get a strictly bounding line for the set of considered points but it is not always the "best" one. This situation could happen when a large number of points of the strictly bounding line does not belong to the current blurred segment, i.e. when the analysed curve is highly noisy.

Proof of the theorem we shall only demonstrate point (iii) as point (i) is obvious and point (ii) symmetrical of case (iii).

The demonstration requires the result described hereafter (pp. 27 of [4]).

Lemma 15 Let P be a point with integer coordinates, $\vec{u}(b, a)$ and $\vec{u}'(b', a')$ two vectors with integer coordinates. If $ab' - a'b = 1$, the interior of the parallelogram $PQRS$ where $\vec{PQ} = \vec{u}$ and $\vec{PS} = \vec{u}'$ contains no point with integer coordinates.

Proof of the lemma A point $N(x, y)$ inside the parallelogram is located between both lines (PS) and (QR) (see Figure 5), it implies that the angles (\vec{PS}, \vec{PN}) and (\vec{QR}, \vec{QN}) , and therefore their sinus, are with opposite signs. It is equivalent to

$$\left| \begin{array}{c} b' \\ a' \end{array} \begin{array}{c} x - x_P \\ y - y_P \end{array} \right| \left| \begin{array}{c} b' \\ a' \end{array} \begin{array}{c} x - x_P - b \\ y - y_P - a \end{array} \right| < 0,$$

thus $(b'(y - y_P) - a'(x - x_P))(b'(y - y_P - a) - a'(x - x_P - b)) < 0$.

If $ab' - a'b = 1$, we obtain, by calling Δ the expression $b'(y - y_P) - a'(x - x_P)$, $\Delta(\Delta - 1) < 0$, which implies $\Delta > 0$ and $\Delta - 1 < 0$. But it is impossible to

have $0 < b'(y - y_P) - a'(x - x_P) < 1$ with x and y being integer as a' , b' , x_P , y_P are integer by hypothesis. \square

(a) Firstly let us prove that all points of $\mathcal{S}f \cup \{M\}$ belong to \mathcal{D}' . For this, we only have to prove that all points of $\mathcal{S}f \cup \{M\}$ are located between both leaning lines of \mathcal{D}' , i.e. between the line through M whose main vector is $\overrightarrow{P_{r(M)-1}M} = (b', a')$ that we'll call d'_L and the line through U_L with the same main vector named d'_U .

By looking at Figure 6, we can see that two triangles must be studied more precisely, the one coming from the point U_L and the other coming from the point Q , intersection of d'_L with the lower leaning line of \mathcal{D} , named d_L .

Let us consider the triangle U_LSS' , S being the point whose abscissa is x_M on the upper leaning line of \mathcal{D} (named d_U) and S' the point whose abscissa is x_M on the line d'_U which is the upper leaning line of \mathcal{D}' . We must verify that there is no point of $\mathcal{S}f$ contained inside the triangle U_LSS' other than those located on the line d'_U . There is no point of $\mathcal{S}f$ on the line d_U other than U_L because U_L is, by hypothesis, the last upper leaning point of \mathcal{D} .

Let us show that $ab' - a'b = 1$ (*):

$$\begin{aligned} ab' - a'b &= a(x_M - x_{P_{r(M)-1}}) - (y_M - y_{P_{r(M)-1}})b \\ &= (ax_M - by_M) - (ax_{P_{r(M)-1}} - by_{P_{r(M)-1}}) \\ &= r(M) - (r(M) - 1) = 1. \end{aligned}$$

According to lemma, there is no point whose coordinates are integer inside the parallelogram whose origin is U_L generated by both vectors u and u' . The abscissa of the vertex of this parallelogram which does not belong to d_U and d'_U is $x_{U_L} + b + b'$. However we have, by definition of b' , $b' = x_M - x_{P_{r(M)-1}} > x_M - b$, and therefore $x_{U_L} + b + b' > x_M$. This vertex does not belong to the segment $\mathcal{S}f$. It is a fortiori true for the vertices, which do not belong to the lines d_U and d'_U , of the parallelograms deduced from the previous one by successive translations by vector u or u' , possibly necessary to cover the triangle U_LSS' . For the points of $\mathcal{S}f$ which are close to the origin, it is clear that in the case where the abscissa of Q is negative, the points of $\mathcal{S}f$ are above d_L as they belong to \mathcal{D} , and therefore above d'_L . In the other case, we consider the triangle QTT' where T and T' are the points intersection of the axis Oy with d_L and d'_L . This triangle is contained inside the triangle delimited by the line d'_L , the axis Oy and the parallel to d_L through $P_{r(M)-1}$ (in gray on Figure 6). As we did for the triangle U_LSS' , we can cover this triangle by the parallelogram whose origin is $P_{r(M)-1}$ generated by both vectors $-u$ and $-u'$ and one or several parallelograms obtained by successive translations by vector $-u'$. The points of $\mathcal{S}f$ are necessarily some vertices of these parallelograms. Those ones which belong to d'_L also belong to \mathcal{D}' . The other ones have a negative abscissa because $x_{P_{r(M)-1}}$ is less than b .

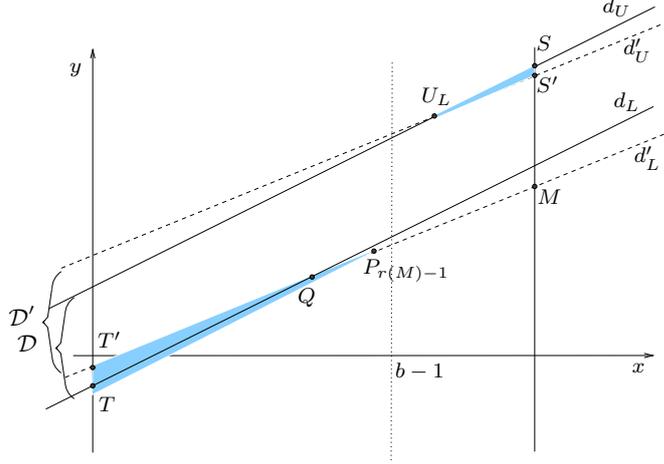


Fig. 6.

(b) Secondly, let us prove that \mathcal{D}' is strictly bounding for $\mathcal{S}f \cup \{M\}$. By construction \mathcal{D}' has in $\mathcal{S}f \cup \{M\}$ a lower leaning point, the point M , and an upper leaning point, the point U_L . The third leaning point is the point $P_{r(M)-1}$ which does not necessarily belong to $\mathcal{S}f \cup \{M\}$.

(c) Finally, let us prove that $\overrightarrow{V'_{+1}} = (b' - b, a' - a) + [\frac{b}{b'}](b', a')$. Let Q and S be two points of \mathcal{D}' such that $\overrightarrow{QS} = (b' - b, a' - a) + [\frac{b}{b'}](b', a')$. Then,

$$\begin{aligned} r'(S) - r'(Q) &= a'x_S - b'y_S - (a'x_Q - b'y_Q) \\ &= a'(b' - b + [\frac{b}{b'}]b') - b'(a' - a + [\frac{b}{b'}]a') \\ &= -a'b + ab' = 1, \text{ thanks to } (*). \end{aligned}$$

Moreover, $x_{\overrightarrow{QS}} = b' - b + [\frac{b}{b'}]b'$. As b and b' are greater than 0 and $[\frac{b}{b'}] \leq \frac{b}{b'}$, $0 \leq x_{\overrightarrow{QS}} \leq b'$. Consequently, $\overrightarrow{QS} = \overrightarrow{V'_{+1}}$.

The proof of formula for $\overrightarrow{V'_{-1}}$ is similar. \square

This theorem will allow us to determine an incremental and linear algorithm for the recognition of a given blurred segment and to deduce a linear algorithm for the segmentation of 8-connected curves into blurred segments. These algorithms are presented in the next section.

4 A linear segmentation algorithm of 8-curves into blurred segments

4.1 Incremental recognition of a blurred segment with order d in the first octant

Let d be a real number, the algorithm hereafter analyses an 8-connected sequence E of pixels located in the first octant and determines if E is a blurred segment with order d . Moreover, in that case, the characteristics a , b , μ and ω of a strictly bounding line are calculated.

Each point M of E is analyzed and added to the current segment and the characteristics a , b , μ and ω of a strictly bounding line of this segment can possibly change according to the theorem of the previous section.

At each step, the value $\frac{\omega}{b}$ is evaluated and if it is greater than d , the recognition stops. E is not a blurred segment whose we may calculate the strictly bounding line according to the theorem.

Algorithm 1 Blurred Segment Recognition Algorithm

Input: E an 8-connected sequence of points and d the order authorized for the blurred segment

Output: a boolean value $isSegment$,

- false if E is not a blurred segment with order d according to the theorem
- true else, in this case a, b, μ and ω are the characteristics of the blurred segment

Initialisation: $a = 0$, $b = 1$, $\omega = b$, $\mu = 0$, $isSegment = true$, $M =$ the first point of E , $L_L = U_L = M = (0, 0)$, $\vec{V}_{-1} = (0, 1)$, $\vec{V}_{+1} = (0, -1)$

while E is not entirely scanned and $isSegment$ **do**

$M =$ next point of E ;

$r = ax_M - by_M$;

if $r = \mu$ **then** $U_L = M$; **endif**

if $r = \mu + \omega - 1$ **then** $L_L = M$; **endif**

if $r \leq \mu - 1$ **then**

$a_{last} = a$; $b_{last} = b$; $\mu_{last} = \mu$; $\omega_{last} = \omega$;

$a = [\frac{x_M - x_{\vec{V}_{-1}}}{b_{last}}]a_{last} + y_{\vec{V}_{-1}}$;

$b = [\frac{x_M - x_{\vec{V}_{-1}}}{b_{last}}]b_{last} + x_{\vec{V}_{-1}}$;

$\mu = ax_M - by_M$;

$\omega = ax_{L_L} - by_{L_L} - \mu + 1$;

$U_L = M$; possible update of L_L ;

$\vec{V}_{-1} = (b - b_{last} + [\frac{b_{last}}{b}]b, a - a_{last} + [\frac{b_{last}}{b}]a)$;

$\vec{V}_{+1} = (b_{last} - [\frac{b_{last}}{b}]b, a_{last} - [\frac{b_{last}}{b}]a)$;

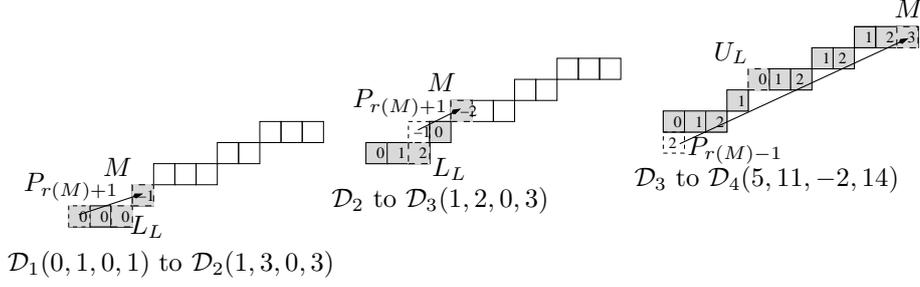


Fig. 7.

```

endif
if  $r \geq \mu + \omega$  then
   $a_{last} = a$  ;  $b_{last} = b$  ;  $\mu_{last} = \mu$  ;  $\omega_{last} = \omega$  ;
   $a = \left[ \frac{x_M - x_{\overrightarrow{V_{+1}}}}{b_{last}} \right] a_{last} + y_{\overrightarrow{V_{+1}}}$  ;
   $b = \left[ \frac{x_M - x_{\overrightarrow{V_{+1}}}}{b_{last}} \right] b_{last} + x_{\overrightarrow{V_{+1}}}$  ;
   $\mu = ax_{U_L} - by_{U_L}$  ;
   $\omega = ax_M - by_M - \mu + 1$  ;
   $L_L = M$  ; possible update of  $U_L$ 
   $\overrightarrow{V_{-1}} = (b_{last} - \left[ \frac{b_{last}}{b} \right] b, a_{last} - \left[ \frac{b_{last}}{b} \right] a)$  ;
   $\overrightarrow{V_{+1}} = (b - b_{last} + \left[ \frac{b_{last}}{b} \right] b, a - a_{last} + \left[ \frac{b_{last}}{b} \right] a)$  ;
endif
   $isSegment = \frac{\omega}{b} \leq d$  ;
endwhile

```

An example of the algorithm processing is presented at Figure 7. The recognition of the blurred segment whose order is 1.5 and whose strictly bounding line is $\mathcal{D}(5, 11, -2, 14)$ required 3 steps during which the characteristics of a strictly bounding line have been calculated. On the figure, the pixels are weighted by the remainder value as a function of the strictly bounding line of the current segment.

Comments about the Algorithm 1:

1. At the beginning of the recognition, the segment is an exact segment of a naive line which is a strictly bounding line of this segment. The introduction of a strongly exterior point changes that state, the segment is no longer exact, it does no longer contain all points of the discrete bounding line on the considered interval.
2. **Update of the leaning points**
 In the case $r \leq \mu - 1$, an update of L_L is mandatory if a point of the new lower leaning line, whose abscissa is between x_{L_L} and x_M , belongs to the segment; indeed L_L must always be the last lower leaning point of the segment. When it exists, this point is necessarily the vertex C following L_L

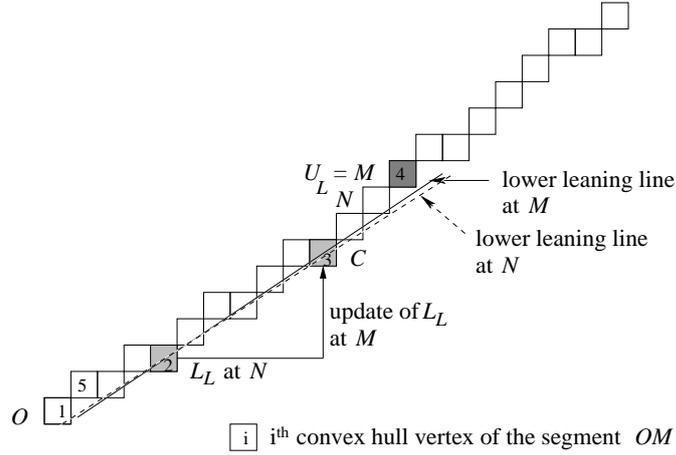


Fig. 8. Example of a lower leaning point update

in the convex hull of the segment, with a counterclockwise orientation. To keep the linearity of the algorithm, we incrementally compute the convex hull of the segment by the Melkman algorithm [6], and we memorize the indices of U_L and L_L . L_L needs to be updated if and only if the slope of the straight line ($L_L C$) is the same as the slope of the new strictly bounding line of the segment. In the example of Fig. 8, it is the case : the slope of ($L_L C$) is $\frac{2}{3}$ and the new strictly bounding line at M is $\mathcal{D}(2, 3, -1, 2)$. At the point N , the strictly bounding line of the segment is $\mathcal{D}(7, 11, -4, 11)$. It is similar for U_L in the case $r \geq \mu + \omega$, with the vertex before U_L .

3. Complexity

The operations to make the characteristics evolve are not costly (memorizing the last scanned leaning points, additions, subtractions between coordinates in $O(1)$). The incremental algorithm of Melkman is in $O(N)$, with N the number of points of E . The possible update only needs one test to compare the slopes. So, the algorithm complexity is $O(N)$.

4.2 A linear algorithm for splitting an 8-curve into blurred segments with order d

The theorem of Section 3 and the algorithm of the previous paragraph are used to split a digital 8-curve \mathcal{C} into 8-connected blurred segments with a fixed order d . The curve \mathcal{C} is incrementally scanned, each point is watched. Let $\mathcal{S}f$ be the current blurred segment, the point M of \mathcal{C} is added to $\mathcal{S}f$ by a procedure `addPointSf` which possibly changes the characteristics of the strictly bounding line of $\mathcal{S}f \cup M$ (according to the theorem of the previous section). According to the obtained ratio $\frac{\omega}{\max(|a|, |b|)}$, the current segment includes or not the point M .

- If $\frac{\omega}{\max(|a|,|b|)} > d$, M is not included in the current blurred segment $\mathcal{S}f$, this one ends at the predecessor M_p of M in the curve \mathcal{C} , the strictly bounding line of $\mathcal{S}f$ has the same characteristics as the ones obtained before the point M was added. A new blurred segment then starts, consisting of points M_p and M .
- If $\frac{\omega}{\max(|a|,|b|)} \leq d$, M is included into $\mathcal{S}f$, $\mathcal{S}f$ becomes $\mathcal{S}f \cup \{M\}$ and the characteristics of its strictly bounding line are the last calculated ones.

Any scan of the curve must take into account the possible changes of octants, it is therefore mandatory to include in the algorithm the detection and the management of octant changes. Several solutions are possible, we chose to do all calculations in the first octant by considering that the first point of each segment is $(0, 0)$. For each added point M , we work with its *transformed point* M_c in the first octant after having checked that M belongs to the octant of the current segment. In the algorithm given below, the `testOctant` procedure:

- tests the validity of the point M according to the octant of the current segment, and sets *isSameOctant* to the according value,
- possibly updates the number of the octant of the current segment (variable *octantNumber* of the algorithm).

The way the boolean value *isSameOctant* is updated in the procedure `testOctant` depends on the *accepted directions* of a segment in a given octant.

When the current segment has entirely been scanned, we compute the transformed characteristics of those obtained in the first octant, according to the current octant, by using compositions of symmetries of discrete lines [7].

Algorithm 2 Splitting a curve into blurred segments

Input: \mathcal{C} an 8-connected sequence of points and d the order authorized for the blurred segments

Output: a list \mathcal{L} of blurred segments, each of them being defined by its number of points *nbPoint* and the characteristics a, b, μ, ω of a strictly bounding line

Initialisation: $\mathcal{L} = \emptyset$, $a = 0$, $b = 1$, $\mu = 0$, $\omega = b$, $nbPoint = 1$, $end = false$,
 $octantNumber = 0$, $isSegment = true$, $isSameOctant = true$,
 $M = \text{the first point of } \mathcal{C}$, $L_L = U_L = M_c = (0, 0)$, $\overrightarrow{V_{-1}} = (0, 1)$,
 $\overrightarrow{V_{+1}} = (0, -1)$

while *!end* **do**

while *isSegment* and *isSameOctant* and *!end* **do**

 // Loop of determination of a blurred segment

$M_{last} = M$;

$M = \text{next point of } \mathcal{C}$; $M_c = \text{image of } M \text{ in the first octant}$;

$testOctant(M)$;

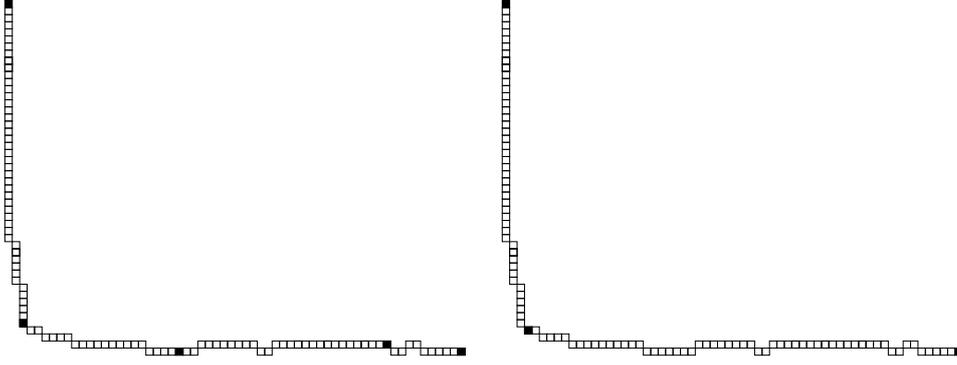


Fig. 9.

```

 $a_{last} = a ; b_{last} = b ; \mu_{last} = \mu ; \omega_{last} = \omega ;$ 
if isSameOctant then
    addPointSf( $M_c$ ) ; //See 4.1
     $isSegment = \frac{\omega}{b} \leq d ;$ 
    if isSegment then  $nbPoint = nbPoint + 1 ;$  endif
endif
end = C is entirely scanned ;
endwhile
if isSegment and isSameOctant then
    // The last segment integrates  $M$ 
    Add to  $\mathcal{L}$  the blurred segment characterized by  $nbPoint$  and, according to
    the current octant, the transformed characteristics of  $a, b, \mu, \omega ;$ 
else
    // We add a segment which does not integrate  $M$ 
    Add to  $\mathcal{L}$  the blurred segment characterized by  $nbPoint$  and, according to
    the current octant, the transformed characteristics of  $a_{last}, b_{last}, \mu_{last},$ 
 $\omega_{last} ;$ 
    if end then
        // The last point of the curve is isolated
        Add to  $\mathcal{L}$  the blurred segment characterized by  $nbPoint = 2$  and the
        transformed characteristics of  $a = 0, b = 1, \mu = 0, \omega = 1 ;$ 
    endif
endif
    // Initialisations for the next segment
     $a = 0 ; b = 1 ; \mu = 0 ; \omega = b ; nbPoint = 1 ; octantNumber = 0 ;$ 
     $isSegment = true ; isSameOctant = true ; M = M_{last} ;$ 
     $L_L = U_L = M_c = (0, 0) ; \vec{V}_{-1} = (0, 1) ; \vec{V}_{+1} = (0, -1) ;$ 
endwhile

```

Remark 16 *The first point of a new blurred segment is the last point of the previous blurred segment.*

Example 17 *Let us consider the curve given in Figure 9, on the left hand*

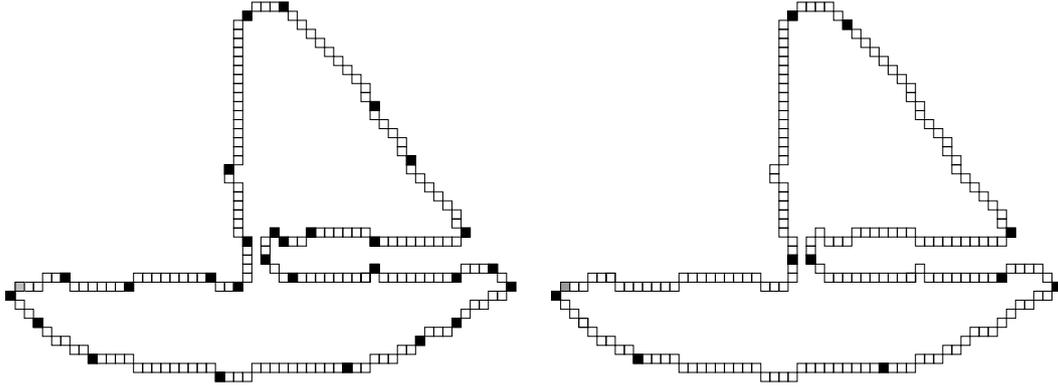


Fig. 10.

side, we can see the segmentation of the curve with an order equal to 2, the obtained blurred segments have the following characteristics:

- First blurred segment located in the octant 6 with a length of 46 and with $\mathcal{D}(-33, 1, -26, 60)$ as strictly bounding line
- Second blurred segment located in the octant 7 with a length of 22 and with $\mathcal{D}(4, -19, -29, 38)$ as strictly bounding line
- Third blurred segment located in the octant 0 with a length of 29 and with $\mathcal{D}(1, 13, -10, 26)$ as strictly bounding line
- Fourth blurred segment located in the octant 7 with a length of 11 and with $\mathcal{D}(1, -5, -4, 10)$ as strictly bounding line.

On the right hand side of Figure 9, the curve segmentation is done with the order 4, there are only two blurred segments.

By using a variant of the algorithm, in some cases, the points whose remainders are $r(M) - 1$ or $r(M) + 1$ can be seeked beyond the first period, it allows to obtain a strictly bounding straight line which is closer to the points of the segment. In particular, in the above case, an horizontal segment can be detected and we obtain, from the order 2, three segments for the above curve, the first two ones are identical and the third blurred segment is located in the octant 0 with a length of 39 and with $\mathcal{D}(0, 1, -1, 2)$ as strictly bounding straight line.

An other example of curve segmentation by our algorithm is presented in Fig. 10; on the left hand side, the order of segmentation is 1 and on the right side, the order is 3. The grey point is the start point of segmentation which is processed clockwise. There is an example with a large number of points, too big to be included in this paper, available on the web page <http://www.loria.fr/~debled/example>.

5 Conclusion and perspectives

We have presented in this paper a new notion of discrete segment named blurred segment, which enables the splitting of discrete curves in a less strict way than with the techniques proposed in [3], by taking into account possible noise. A fast and linear segmentation algorithm was presented, it relies on very simple operations between integers. It is very efficient to segment weakly noisy curves.

However the algorithm will have to be adaptive to obtain performing applications as, for example, in the document analysis domain. An improvement might be the search of “best” joining point with the study of the intersection between two consecutive obtained blurred segments.

Moreover, this notion opens new perspectives ; it might be used to define a notion of blurred tangent by extending the definition of discrete tangents given by A. Vialard [9]. The notion of discrete blurred arc might be as well deduced from the notion of blurred segment and from the work undertaken on the discrete arcs in [1].

Acknowledgments to Xavier Hilaire (QGAR project, LORIA) who asked us to segment a set of noisy curves by using a digital arithmetical geometry approach.

References

- [1] D. Coeurjolly, L. Tougne, Y. Gérard and J.P Reveillès. An elementary algorithm for digital arc segmentation. *Electronic Notes in Theoretical Computer Science*, vol. 46, 2001.
- [2] T.J. Davis. Fast decomposition of digital curves into polygons using the Haar transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, n. 8, august, 1999.
- [3] I. Debled-Rennesson and J.P. Reveillès. A linear algorithm for segmentation of digital curves. *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 9, pp. 635–662, december, 1995.
- [4] G.H. Hardy and E.M. Wright. An introduction to the theory of numbers. Oxford Press, 1989.
- [5] M. Lindenbaum and A. Bruckstein. On recursive, $O(n)$ partitioning of a digitized curve into digital straight segments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, n. 9, september, 1993.
- [6] A. Melkman. On-line construction of the convex hull of a simple polygon. *Info. Proc. Letters*, 25, pp. 11-12, 1987.

- [7] J.-P. Reveillès. Géométrie discrète, calculs en nombre entiers et algorithmique. Thèse d'état. Université Louis Pasteur, Strasbourg, 1991.
- [8] P.L. Rosin. Techniques for assessing polygonal approximations of curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.19, n. 6, june, 1997.
- [9] A. Vialard. Geometrical parameters extraction from discrete path. *6th International workshop DGCI*, Springer, 1996.
- [10] W. Wan and J.A. Ventura. Segmentation of planar curves into straight-line segments and elliptical arcs. *Graphical models and image processing*, vol. 59, n. 6, pp. 484-494, november, 1997.

Multiorder polygonal approximation of digital curves

I. Debled-Rennesson, S. Tabbone and L. Wendling

LORIA-INRIA

Campus Scientifique, BP 239

54506 Vandœuvre-les-Nancy Cedex

France

email: {debled,tabbone,wendling}@loria.fr

Received 16 July 2004; accepted 6 June 2005

Abstract

In this paper, we propose a quick threshold-free algorithm, which computes the angular shape of a 2D object from the points of its contour. For that, we have extended the method defined in [4, 5] to a multiorder analysis. It is based on the arithmetical definition of discrete lines [11] with variable thickness. We provide a framework to analyse a digital curve at different levels of thickness. The extremities of a segment provided at a high resolution are tracked at lower resolution in order to refine their location. The method is threshold-free and automatically provides a partitioning of a digital curve into its meaningful parts.

Key Words: Polygonal Approximation, Scale Space, Discrete Lines

1 Introduction

Generally, the goal of a polygonal approximation method is to provide a cut-out of a sequence of points into segments in order to minimise a global error criterion or to not locally overtake a predefined error. The interest is to give a description which is more compact and more adapted for further interpretation or shape recognition processing. The sequences of points are mainly obtained by an outline detection process in gray-scale images or by a skeleton in graphical documents.

Many polygonal approximation methods have been designed throughout the years [2, 13, 19]. The aim is to approximate a given digital curve by another polygonal curve with a number of line segments considering a local or global approximation error. Some approaches [9, 15, 16] use a linear scan of the digital curve. The aim is to find the longest segments such that the error with the digital curves is lower than a predefined tolerance error. Perez and Vidal [8] have proposed an interesting algorithm, based on a dynamic programming, to minimise a global error of segmentation. However the complexity of the approach is high because the whole space of solutions is checked. Recent works [7, 14] have been proposed to improve it.

Split-and-merge methods [10, 12] start from an initial segmentation and then iteratively split a line if the error is too big and otherwise merge two lines if the error is too small. In some methods [1, 12, 18] anchor points which correspond to high curvature points, are injected in the segmentation process.

Despite the important number of approximation methods, there are still major problems of robustness, stability and complexity when applying geometrical transformations. The algorithms rely on error tolerance thresholds which are manually defined without any knowledge on the most pertinent value of the threshold. Moreover they can be different from a sequence of points to another.

Correspondence to: debled@loria.fr

Recommended for acceptance by J.M. Ogier, T. Paquet, G. Sanchez

In this paper, we present a quick threshold-free algorithm, which computes the angular shape of a 2D object from the points of its contour. This rough representation shall be used, for example, in shape recognition system for a quick first classification of a database of objects. In this perspective, we use the linear method defined in [4, 5] which consists in the segmentation of a given curve into blurred segments. This algorithm depends on a parameter, called order, which allows to control the amplitude of the authorised noise by setting the thickness of the discrete line bounding the blurred segment.

The study of the behaviour of this algorithm lead us to use it on a multioorder analysis framework and we obtain a threshold-free method which permits to split a given curve into meaningful segments.

In the next section, we recall the theoretical concepts related to the blurred segment definition. Then, in section 3, we present the algorithm used to split a given curve into several blurred segments at a given order. At last, we present in sections 4 and 5 our approach and the threshold-free algorithm to analyse a sequence of points in a multioorder space as well as the obtained results.

2 Blurred segment

In the following, we refer to the first octant (8^{th}) of the plane such that $x \geq 0$ and $0 \leq y \leq x$. The notion of blurred (or fuzzy) segments relies on the *arithmetical definition of discrete lines* [11] where a line, whose slope is $\frac{a}{b}$, lower bound μ and thickness ω (with a, b, μ and ω being integer) is the set of integer points (x, y) verifying $\mu \leq ax - by < \mu + \omega$. Such a line is noted $\mathcal{D}(a, b, \mu, \omega)$. From now on, we shall name these segments blurred segments rather than fuzzy segments in order to prevent any confusion with fuzzy logic concepts.

The real lines $ax - by = \mu + \omega - 1$ and $ax - by = \mu$ are named the leaning lines of $\mathcal{D}(a, b, \mu, \omega)$. Moreover, the points (x_L, y_L) (resp. (x_U, y_U)) of a discrete line $\mathcal{D}(a, b, \mu, \omega)$ which verify $ax_L - by_L = \mu + \omega - 1$ (resp. $ax_U - by_U = \mu$) are called **lower (resp. upper) leaning points** and located under (resp. above) all other points of \mathcal{D} (see dashed pixels in figure 1). We have the following definitions [4]:

Definition 1: A set $\mathcal{S}f$ of consecutive points ($|\mathcal{S}f| \geq 2$) of an 8-connected curve is a **blurred segment with order d** if there is a discrete line $\mathcal{D}(a, b, \mu, \omega)$ such that all points of $\mathcal{S}f$ belong to \mathcal{D} and $\frac{\omega}{\max(|a|, |b|)} \leq d$. The line \mathcal{D} is said **bounding** for $\mathcal{S}f$.

The order of a blurred segment allows to limit the thickness of the discrete line bounding the 8-connected sequence of points of the blurred segment and, so doing, to control the length of vertical steps of the bounding line. In order to be reasonably close to the points of the blurred segment, we introduce more restrictive conditions to the discrete line with the notion of strictly bounding line as defined hereafter.

Definition 2: Let $\mathcal{S}f$ be a blurred segment with order d whose abscissa interval is $[0, l - 1]$, and let $\mathcal{D}(a, b, \mu, \omega)$ be a bounding line of $\mathcal{S}f$. \mathcal{D} is named **strictly bounding for $\mathcal{S}f$** if, \mathcal{D} possesses at least three leaning points in the interval $[0, l - 1]$ and, $\mathcal{S}f$ contains at least one lower leaning point and one upper leaning point of \mathcal{D} .

The following theorem, proven in [4], studies the different possible cases of the growth of a blurred segment.

Theorem 1: Let us consider a blurred segment $\mathcal{S}f$ in the first octant whose abscissa interval is $[0, l - 1]$ and $\mathcal{D}(a, b, \mu, \omega)$, a strictly bounding line. In this case, the order of $\mathcal{S}f$ is $\frac{\omega}{b}$. Let $M(x_M, y_M)$ be an integer point connected to $\mathcal{S}f$ whose abscissa is equal to l or $l - 1$. We call **remainder** at M , as a function of \mathcal{D} , noted $r(M)$ and defined by $r(M) = ax_M - by_M$.

- (i) If $\mu \leq r(M) < \mu + \omega$, then $M \in \mathcal{D}$;
 $\mathcal{S}f \cup M$ is a blurred segment whose order is $\frac{\omega}{b}$ with \mathcal{D} as strictly bounding line.

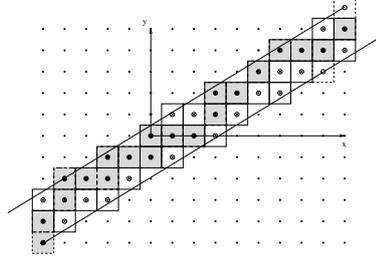


Figure 1: Points of $\mathcal{D}(8, 13, -6, 32)$ in the abscissa interval $[-5, 9]$, in grey a blurred segment with order 3 and \mathcal{D} as strictly bounding line.

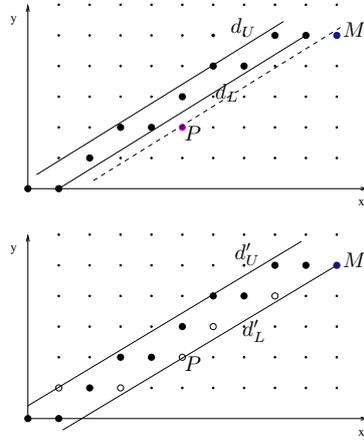


Figure 2: An example of blurred segment growth relying on the theorem 1.

- (ii) If $r(M) \leq \mu - 1$, then M is exterior to \mathcal{D} ;
 $\mathcal{S}f \cup M$ is a blurred segment whose order is $\frac{\omega'}{b'}$ and the line $\mathcal{D}'(a', b', \mu', \omega')$ is strictly bounding, with
- b' and a' coordinates of the vector $\overrightarrow{P_{r(M)+1}M}$, $P_{r(M)+1}$ being the point whose remainder is $r(M) + 1$ with regard to \mathcal{D} and $x_{P_{r(M)+1}} \in [0, b - 1]$,
 - $\mu' = a'x_M - b'y_M$
 - $\omega' = a'x_{L_L} - b'y_{L_L} - \mu' + 1$, with $L_L(x_{L_L}, y_{L_L})$ last lower leaning point of the line \mathcal{D} present in $\mathcal{S}f$.
- (iii) If $r(M) \geq \mu + \omega$, then M is exterior to \mathcal{D} ;
 $\mathcal{S}f \cup \{M\}$ is a blurred segment whose order is $\frac{\omega'}{b'}$ and the line $\mathcal{D}'(a', b', \mu', \omega')$ is strictly bounding with
- b' and a' coordinates of the vector $\overrightarrow{P_{r(M)-1}M}$, $P_{r(M)-1}$ being the point whose remainder is $r(M) - 1$ with regard to \mathcal{D} and $x_{P_{r(M)-1}} \in [0, b - 1]$,
 - $\mu' = a'x_{U_L} - b'y_{U_L}$ with $U_L(x_{U_L}, y_{U_L})$ last upper leaning point of the line \mathcal{D} present in $\mathcal{S}f$,
 - $\omega' = a'x_M - b'y_M - \mu' + 1$.

An example of application of this theorem is given in the figure 2. On the top of it, a blurred segment Sf of order 1, $\mathcal{D}(5, 8, -2, 8)$ is strictly bounding for Sf , d_U and d_L are the leaning lines of \mathcal{D} . The point $M(10, 5)$ is added to Sf , as $r_{\mathcal{D}}(M) = 10$, adding M to Sf corresponds to the case (iii) of the theorem : P is the point in $[0, 7]$ such that $r_{\mathcal{D}}(P) = 9$, therefore, $Sf \cup \{M\}$ has $\mathcal{D}'(3, 5, -2, 7)$ as strictly bounding line whose slope is calculated with the vector PM . In the bottom of figure 2, a representation of \mathcal{D}' and $Sf \cup \{M\}$ (black points) is given. The points of \mathcal{D}' which do not belong to $Sf \cup \{M\}$ are in white, d'_U and d'_L are the leaning lines of \mathcal{D}' .

The algorithm presented in the following section is directly deduced from this theorem.

3 A segmentation algorithm of 8-connected curves into blurred segments

The curve \mathcal{C} is incrementally scanned, each point is watched. Let Sf be the current order d blurred segment, a point M of \mathcal{C} is added to Sf , the characteristics of a strictly bounding line of $Sf \cup M$ are calculated (according to the theorem 1). The current segment includes the point M if the value of the obtained ratio $\frac{\omega}{\max(|a|, |b|)}$ is lesser than or equal to the order d . In the algorithm given hereafter, two procedures are used:

- Each point M of E is analysed, transformed in the first octant and added to the current segment by the procedure `addPointSf` which possibly changes the characteristics a , b , μ and ω of a strictly bounding line of this segment according to the theorem 1.
- The `testOctant` procedure tests the validity of the point M according to the octant of the current segment, and sets the boolean `isSameOctant` to the right value, possibly updates the number of the octant of the current segment. The way the boolean value `isSameOctant` is updated in the procedure `testOctant` depends on the authorised directions (freeman code elements) of a segment in a given octant.

Algorithm 1: Blurred Segmentation

Input: \mathcal{C} an 8-connected sequence of points and d the authorised order for the blurred segments.

Output: the list L of blurred segments, each of them being defined by its number of points `nbPoint` and the characteristics a, b, μ, ω of a strictly bounding line.

Initialisation: $a = 0, b = 1, \mu = 0, \omega = b, nbPoint = 1, M_c = (0, 0)$,
`isSegment = true, end = false, isSameOctant = true`,
 M = the first point of \mathcal{C} .

```

while !end do
  while isSegment and isSameOctant and !end do
     $M_{last} = M$  ;
     $M$  = next point of  $\mathcal{C}$  ;
    testOctant( $M$ ) ;  $M_c$  = image of  $M$  in the first octant ;
     $a_{last} = a$  ;  $b_{last} = b$  ;  $\mu_{last} = \mu$  ;  $\omega_{last} = \omega$  ;
    if isSameOctant then
      addPointSf( $a, b, \mu, \omega, M_c$ ) ;
       $isSegment = \frac{\omega}{b} \leq d$  ;
      if isSegment then nbPoint ++ ; endif
    endif
    end =  $\mathcal{C}$  is entirely scanned ;
  endwhile
  if isSegment and isSameOctant then

```

```

    Add to  $L$  the blurred segment characterised by  $nbPoint$  and, according
    to the current octant, the transformed characteristics of  $a, b, \mu, \omega$  ;
else
    Add to  $L$  the blurred segment characterised by  $nbPoint$  and, according
    to the current octant, the transformed characteristics of  $a_{last}, b_{last},$ 
     $\mu_{last}, \omega_{last}$  ;
endif
 $a = 0 ; b = 1 ; \mu = 0 ; \omega = b ; nbPoint = 1 ; M = M_{last} ; M_c = (0, 0) ;$ 
 $isSegment = true ; isSameOctant = true ;$ 
endwhile

```

This algorithm is very fast; each point M of \mathcal{C} is analysed only once and added to the current segment by the procedure `addPointSf`. Furthermore the operations which are necessary to make the characteristics evolve are not costly (for details see [5]). The order of a blurred segment, allows to control the amplitude of the authorised noise by fixing the thickness of the discrete line bounding the segment.

Study of the behaviour of the Algorithm 1. At a low order, the extremities of the obtained segments are well localised. However, for noisy sets of points, too many small inconsistent segments are also provided. When the order increases, we can see shifts at the extremities points between the obtained segments and the expected ones (c.f. the segmentation at order 7 in the figure 3). Nevertheless the number of segments decreases and only significative segments remain *. For example, in the figure 3, we can see that the number of segments for an order 1 is 15 while it decreases to 3 for an order 7.

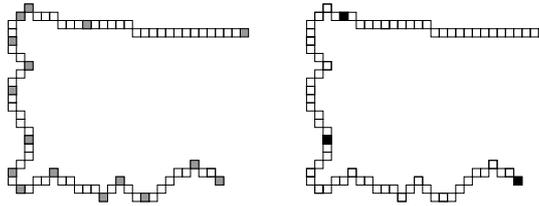


Figure 3: Examples of segmentation at two orders. Grey points (left): extremities points of obtained order 1 blurred segments. Black points (right): extremities points of obtained order 7 blurred segments.

From this statement, we present in the next section a multiorder approach which uses the algorithm 1 and the previous remarks.

4 Multiorder analysis

We propose to analyse a set of points in a multiorder space. Such a space looks like the scale space representation which was first introduced by A. Witkin [17] for the detection of edges at different scales.

Figure 4 shows a multiorder space determined from the segments obtained with algorithm 1 at different orders using the set of points of the figure 3. For each order d , represented on the Z -axis, we put the coordinates of the extremities points of the obtained blurred segments. We can see in this example that 4 extremities remain. They correspond to the description in three segments of the set of points (see figure 5). Moreover the position

*Significative qualifies segments which correspond to a rough polygonal contour defined from the main segments of the set of points.

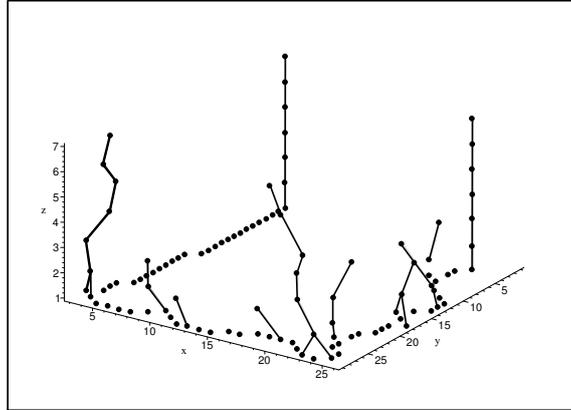


Figure 4: Example of a multiorder space.

of the points evolves except for the extremities of the set. When the number of segments from one order to another does no longer change, we decide to stop the description process. From that order, the extremities of the points are projected onto a lower order and the new position of each point is defined from the nearest segment extremity provided by the lower order. The size of the observation neighbourhood is equal to the thickness of the segment provided by the currently analysed level. This thickness is deduced from the order (cf. §3).

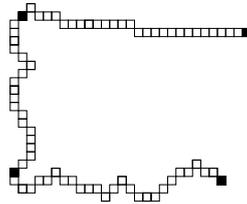


Figure 5: Multiorder Segmentation. Black points: multiorder points obtained with algorithm 2.

We present below the general algorithm which permits to extract the multiorder points. *Approx* is a function, coming from the algorithm of §3, enabling to segment a sequence \mathcal{C} of order d and returning the number of found points, denoted $nbpoint$, as well as a set of points named $L[d][nbpoint]$. The search of neighbourhood for a given order is determined by calculating a « chess board »-type distance [3], denoted $Dist$ with $Dist(a, b) = \max(|x_a - x_b|, |y_a - y_b|)$. This metric assumes that you can make moves on the pixel grid as if you were a King making moves in chess, i.e. a diagonal move counts the same as a horizontal move. This distance is much faster to compute than the Euclidean metric.

Algorithm 2: Multiorder Segmentation

Input: \mathcal{C} an 8-connected sequence

Output: the list L_{MO} of multiorder points

Initialisation : $d = 0, nbpoint = 0$

do $d = d + 1;$
 $nbprec = nbpoint;$

```

    nbpoint = Approx(C, d, L[d]);
while nbpoint  $\neq$  nbprec
    LMO  $\leftarrow$  L[d]
for dc = d - 1 downto 1
     $\forall u \in L_{MO}$ 
    % One scan is enough to consider any point
    if  $\exists p \in L[d_c] / \text{Dist}(p, u) < \text{Int}((d_c + 1)/2) + 1$  then
    % shifting in multi-order space (p becomes u)
    LMO  $\leftarrow$  LMO - {u}; LMO  $\leftarrow$  LMO  $\cup$  {p};
    else
    LMO  $\leftarrow$  LMO - {u}; % the multiorder path is not right
    endif
endfor

```

5 Experimental results

Figures 6, 7, 8 and the appendix section show some examples of polygonal approximation obtained using our approach.

By comparison, we print on the figures 6 and 7 the results found by applying the Wall and Danielsson's method [16]. We adjust at best their cut threshold so that the number of segments is similar to the one of our approach. Moreover we show in the figure 7, the results of the polygonal approximation obtained for each image at the highest order. The stop orders are respectively 11, 4 and 8. We can remark that the number of iterations to obtain the convergence is linked to the regularity of the shape. Moreover, in our approach, the number of segments is automatically obtained while we have to test several different thresholds in the other method to vectorize the image with the same number of segments.

In some cases, some improvements of the algorithm shall have to be done regarding the tracing of points in the multiorder space. For example, on the figure 7.h some points provided by the last order are merged by moving to a lower order. Another problem concerns the tracing of points for which two choices are possible (points at an equal distance) at lower orders (cf. multiorder space of the figure 4). Currently a move by default is chosen. At last the first point of the sequence is considered at all orders even if it does not correspond to the extremity of an expected segment. We foresee to make this cut-out from the estimation of the tangent at some points [6].

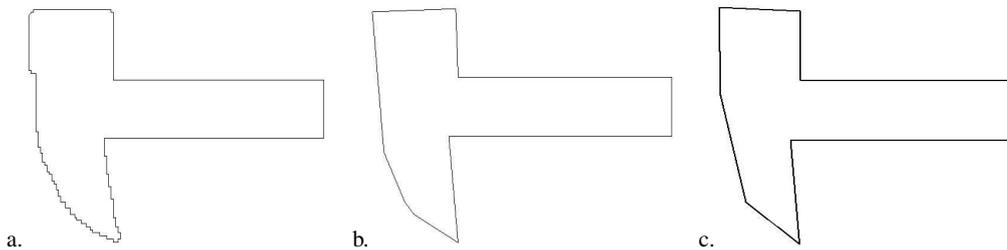


Figure 6: Experimental results.

a) Curve representing a hammer. b) Results automatically obtained with our approach. c) Results obtained with the Wall and Danielsson's method [16] with a cut threshold of 30.

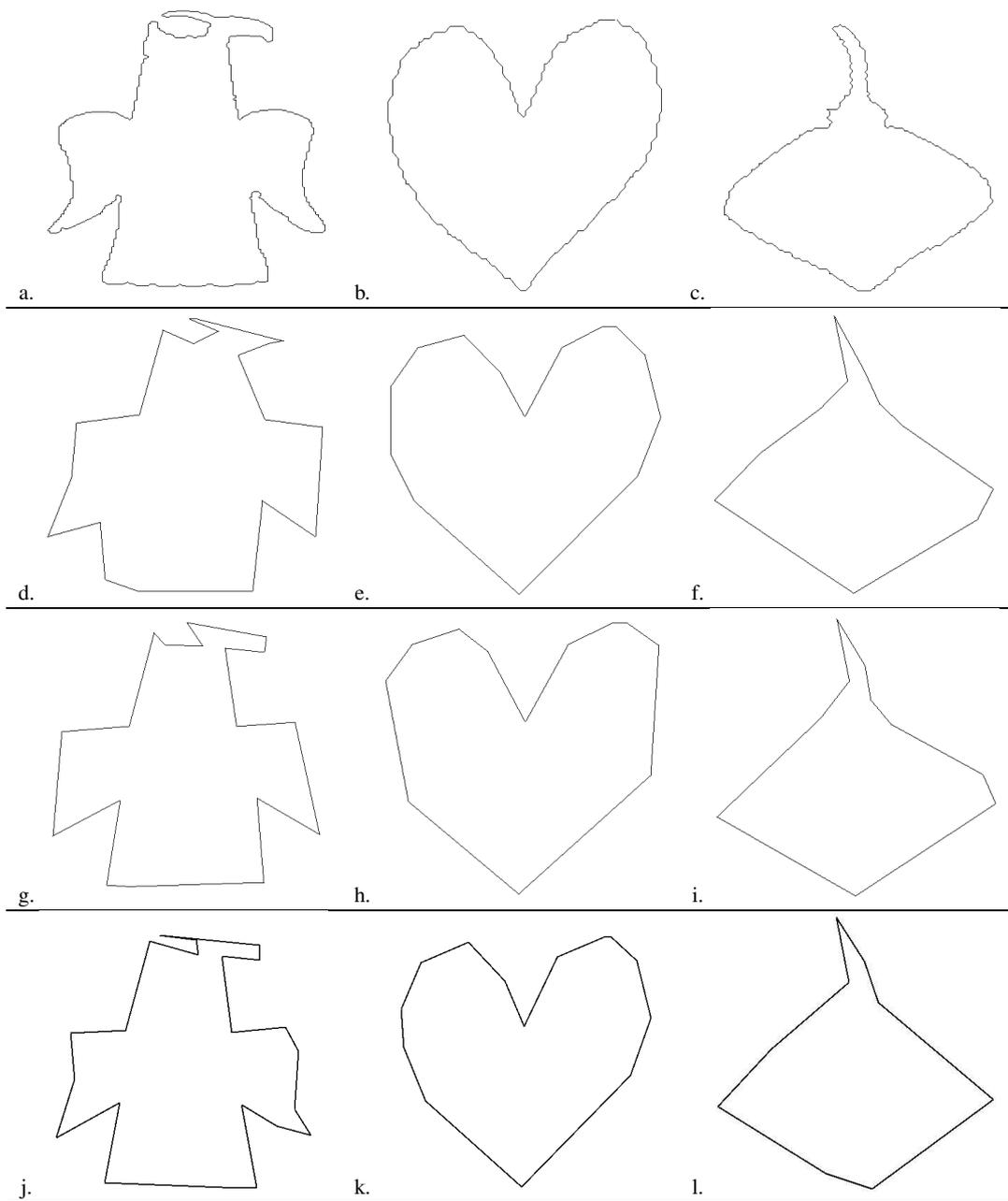


Figure 7: Other experimental results.

a)-c) Initial images. d)-f) Results of the polygonal approximation obtained at high order with algorithm 1. g)-i) Final results obtained with the multiorder analysis (algorithm 2). j)-l) Results obtained with the Wall and Danielsson's algorithm [16] from the QGAR platform (<http://www.qgar.org/>)

Moreover, an error criterion [8, 14] has been implemented to assess the gain of accuracy due to the multiorder approach. The error $e_{i,j}$ is defined from a curve in the range $[c_i; c_j]$ and its orthogonal projection onto the polar line $x \cos(\theta) + y \sin(\theta) = \rho$ as follows: Let $C = \{c_1, \dots, c_N\} = \{(x_1, y_1), \dots, (x_N, y_N)\}$ be a curve defined by N sorted points. Let $Q = \{q_1, \dots, q_{M+1}\}$ be the polygon computed from C using the multi-order algorithm. Each point of Q belongs to C with $q_1 = c_1$ and $q_{M+1} = c_N$. Let us now consider a segment (q_m, q_{m+1}) of Q and the associated series of points $\{c_i, \dots, c_j\}$ in C . The error between them is directly given by:

$$e^2(p_i, p_j) = \sum_{k=i+1}^{j-1} (\sin(\theta_{ij})y_k + \cos(\theta_{ij})x_k - \rho_{ij})^2$$

$\cos(\theta_{ij})$ and $\sin(\theta_{ij})$ are defined from the polar representation $x \cos(\theta) + y \sin(\theta) = \rho$ of the straight line (c_i, c_j) . Considering all the segments of Q , the global error E is calculated as follows:

$$E = \sum_{m=1}^M e^2(q_m, q_{m+1})$$

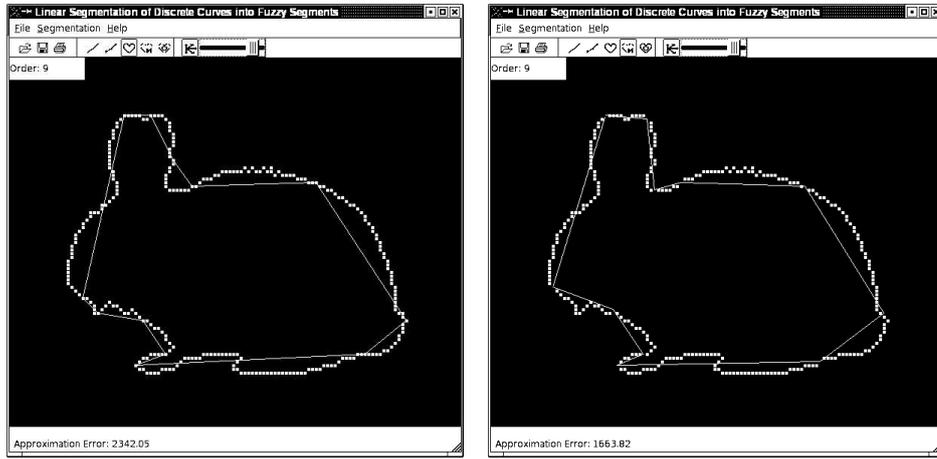


Figure 8: Errors (see at the bottom of each window) reached using blurred segmentation (algorithm 1, on the left hand side) and multiorder segmentation approaches (algorithm 2, on the right hand side).

In most cases, the error reached is lower by using the multiorder segmentation (algorithm 2) than by using the basic blurred segmentation (algorithm 1) for the same order (see figure 8). This refinement of the location of the multiorder points permits to have a better representation of the global shape of the object. Applications on several shapes are provided into the appendix section.

Figure	8	9	10	11	12
(a) Blurred segmentation	2342.05	1558.66	6696.26	1400.55	24840.6
(b) Multiorder segmentation	1663.82	795.007	5810.34	1343.04	10327.7
Ratio	0.71	0.51	0.87	0.96	0,42

Table 1: Error variations in blurred and multiorder segmentations.

Table 1 provides approximation errors reached using both methods applied on figures 8 to 12. We also give a ratio (b/a) to see the improvement due to the multiorder algorithm. A low ratio means that the multiorder segmentation improves the result obtained using the first method.

The more the shape has acute angular points the lower is the error ratio. In this case, the multiover method is more suitable. When shapes are relatively smoothed, both approaches give similar results because the problem of wrong location of extremities points by the first method is not improved by the second method as shown in figure 10 and 11.

6 Conclusion

We have presented a multiover analysis algorithm which provides a fast polygonal approximation of a digital curve from a scale space study. Unlike the other methods where an error of approximation is predefined and set through a trial and error process to achieve the better compromise between the number of segments and their location, our method is threshold-free and automatically provides a partitioning of a digital curve into its meaningful parts. We are aware that in some cases the stability criterion based on the number of segments is not optimal (local minima problem). Further work will be devoted to find better conditions to stop to analyse a digital curve at different levels of thickness. Moreover we wish to extend this approach to a rough classification of shapes. Nonetheless the approach is not invariant to rotation since the approximation method always begins with the same point. Such property should be checked before embedding the method in a pattern recognition process.

Acknowledgements

The authors thank T. Nguyen, internship student at Loria during summer 2004, he tested and implemented all algorithms in C++ with Qt library.

References

- [1] Ansari N., Delp E. J., "On Detecting Dominant Points", *Pattern Recognition*, vol. 24, 5, pp 441-451, 1991.
- [2] Davis T., "Fast Decomposition of Digital Curves into Polygons Using the Haar Transform", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, 8, pp 786-790, 1999.
- [3] P. E. Danielsson, "Euclidean distance mapping", *Computer Vision, Graphics, and Image Processing*, 14, pp. 227- 248, 1980.
- [4] Debled-Rennesson I., Rémy J.-L., Rouyer-Degli J., "Segmentation of Discrete Curves into Fuzzy Segments", *9th International Workshop on Combinatorial Image Analysis, Electronic Notes in Discrete Mathematics*, vol. 12, may 2003.
- [5] Debled-Rennesson I., Rémy J.-L., Rouyer-Degli J., "Segmentation of Discrete Curves into Fuzzy Segments, extended version", *INRIA Report RR-4989* (<http://www.inria.fr/rrrt/rr-4989.html>), november 2003.
- [6] Debled-Rennesson I., "Estimation of Tangents to a Noisy Discrete Curve", *Vision Geometry XII, Electronic Imaging, San José, Jan 2004*.
- [7] Kolesnikov A., Fränti P., "Reduced-search dynamic programming for approximation of polygonal curves", *Pattern Recognition Letters*, vol. 24, 14, pp 2243-2254, 2003.
- [8] Perez J., Vidal E., "Optimum polygonal approximation of digitized curves", *Pattern Recognition Letters*, vol. 15, pp 743-750, 1994.
- [9] Ramer U., "An Iterative Procedure for the Polygonal Approximation of Plane Curves", *Computer Graphics and Image Processing*, vol. 1, pp 244-256, 1972.

- [10] Ray B., Ray K., "A new split-and-merge technique for polygonal approximation of chain coded curves", *Pattern Recognition Letters*, vol. 16, pp 161-169, 1995.
- [11] Reveillès J.-P., "Géométrie discrète, calculs en nombre entiers et algorithmique", *Thèse d'état. Université Louis Pasteur, Strasbourg*, 1991.
- [12] Rosin P. L., West G. A., "Segmentation of Edges into Lines and Arcs", *Image and Vision Computing*, vol. 7, 2, 1989, pp 109-114, 1989.
- [13] Rosin P. L., "Techniques for Assessing Polygonal Approximations of Curves", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, 6, pp 659-666, 1997.
- [14] Salotti M., "An efficient algorithm for the optimal polygonal approximation of digitized curves", *Pattern Recognition Letters*, vol. 22, pp 215-221, 2001.
- [15] Sklansky J., Gonzalez V., "Fast Polygonal Approximation of Digitized Curves", *Pattern Recognition*, vol. 12, 1980, pp 327-331, 1980.
- [16] Wall K., Danielsson P., "A Fast Sequential Method for Polygonal Approximation of Digitized Curves", *Computer Vision, Graphics and Image Processing*, vol. 28, 1984, pp 220-227, 1984.
- [17] Witkin A., "Scale space filtering", *Proc. International Joint Conference on Artificial Intelligence*, 1983.
- [18] Wu W., Wang M., "Detecting the Dominant Points by the Curvature-Based Polygonal Approximation", *Graphical Models and Image Processing*, vol. 55, 2, pp 79-88, 1993.
- [19] Yin P., "A Tabu Search Approach to Polygonal Approximation of Digital Curves", *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 14, 2, pp 243-255, 2000.

Appendix

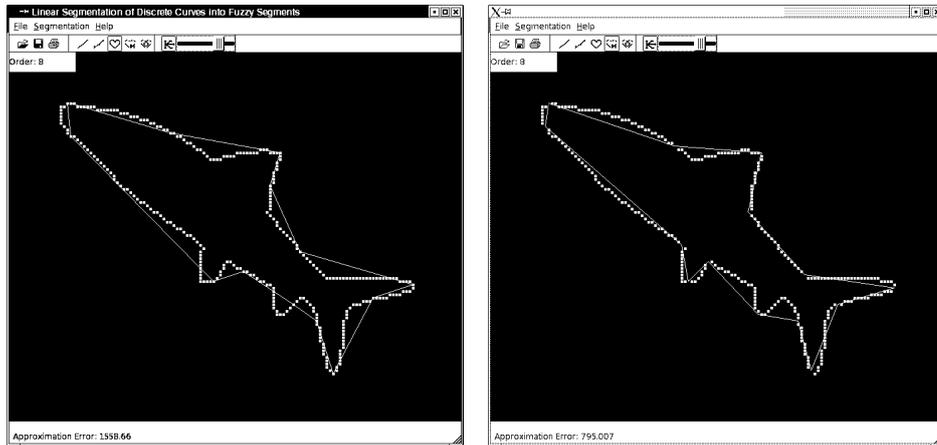


Figure 9: Errors (see at the bottom of windows) reached using blurred segmentation (algorithm 1) and multi-order segmentation approaches (algorithm 2).

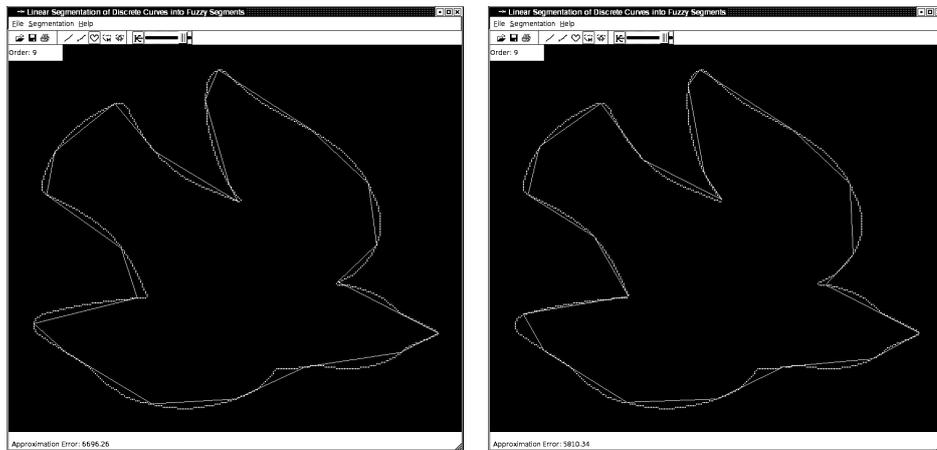


Figure 10: Errors (see at the bottom of windows) reached using blurred segmentation (algorithm 1) and multi-order segmentation approaches (algorithm 2).

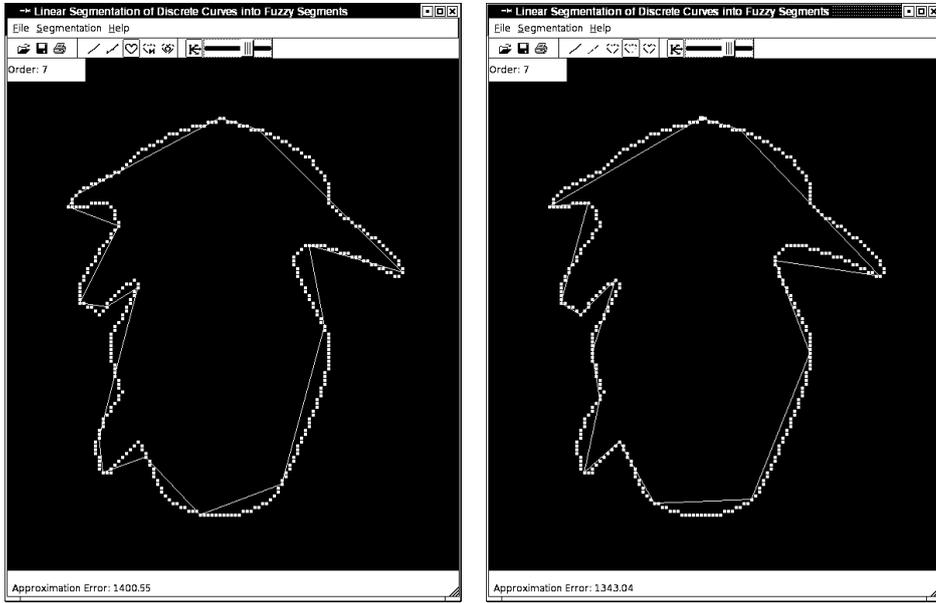


Figure 11: Errors (see at the bottom of windows) reached using blurred segmentation (algorithm 1) and multi-order segmentation approaches (algorithm 2).

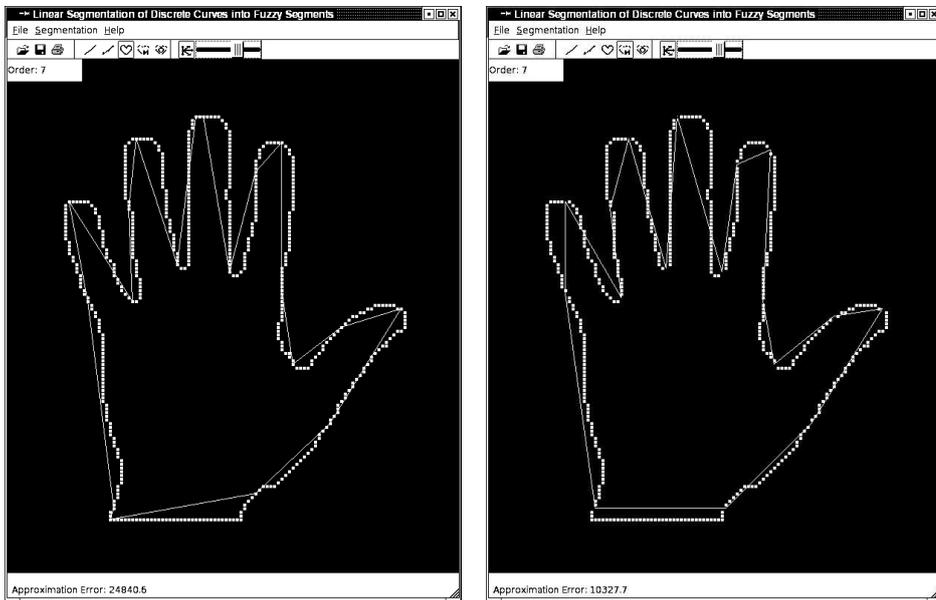


Figure 12: Errors (see at the bottom of windows) reached using blurred segmentation (algorithm 1) and multi-order segmentation approaches (algorithm 2).

CONVEX AND CONCAVE PARTS OF DIGITAL CURVES

Helene Doerksen-Reiter
Fachbereich Mathematik
Universitaet Hamburg
helene.doerksen@hamburg.de

Isabelle Debled-Rennesson
Equipe Adage
LORIA Nancy
debled@loria.fr

Abstract Decomposition of a digital curve into convex and concave parts is of relevance in several scopes of image processing.

In digital plane convexity cannot be observed locally. It becomes an interesting question, how far one can decide whether a part of a digital curve is convex or concave by a method which is “as local as possible”. In [3], it was proposed to define the meaningful parts of a digital curve as meaningful parts of the corresponding polygonal representation. This technique has an approximative character.

In our considerations, we use geometry of arithmetical discrete line segments [15, 1]. We will introduce an exact method to define convex and concave parts of a digital curve.

Keywords: digital geometry, digital convexity, discrete line, convex and concave curves, polygonal representation.

Introduction

Roots of digital geometry can be found in practical applications of digital image processing and computer graphics. A considerable part of books on digital geometry is devoted to convexity (see e.g. [17, Chapter 4.3]). It is a simple observation that convex parts of objects determine visual parts which are of importance, for example, for recognition of objects by comparing with given shapes e.g. from a database. However, the problem is that many significant parts are not convex since a visual part may have concavities. So, one is interested in the decomposition of the boundary of a digital set into convex

and concave parts. In an earlier paper [12], such a partition was performed by segmenting the boundary into digital line segments. In an other paper [3], it was proposed to define the meaningful parts of the boundary by meaningful parts of the corresponding polygonal representation. The first method is much rougher, however, both techniques have an approximative character.

Decomposition into meaningful parts allows simplification the shapes by discrete evolution [13]. The aim of this paper is to investigate the discrete evolution process and its properties specifically with respect to convex and concave parts of the boundary from a low-level point of view. The principal idea is to use the geometrical properties of arithmetical discrete lines [15, 1].

In 1987, Scherl [16] proposed a method based on sets of descriptors which are points of local support with respect to a finite number of directions. In a certain sense, Scherl's descriptors segment the boundary of a set into parts which are "suspicious candidates" for being convex or concave parts.

In the section 1 of this paper, we recall definitions about discrete curves. Then, in the section 2, we present the definition and geometrical properties of arithmetical discrete lines which are used to define the fundamental segments of 8-curves introduced in the section 3. Thanks to this notion, an adaptation of the concept of convex and concave curve is proposed in section 4. Finally, in section 5, the decomposition of curves into maximal convex or concave parts is studied and, in section 6, an application to fundamental polygonal representations of digital curves is presented.

1. Digital Curves on \mathbb{Z}^2

The *digital space* \mathbb{Z}^d is the set of all points in Euclidean space \mathbb{R}^d having integer coordinates. The digital space \mathbb{Z}^2 is also called *digital plane*. The subsets of digital plane are termed *digital sets*, often they are also called *digital objects* or *digital images*. Single elements of \mathbb{Z}^2 are termed *grid points*. A digital set consisting of all grid points which are lying on a horizontal, vertical or diagonal real line in \mathbb{R}^2 is called a *horizontal, vertical or diagonal grid line*, respectively.

The neighborhood structure is a significant concept in the study of digital objects. In our considerations, we concentrate on 8-neighborhood structure for sets and 4-neighborhood structure for their complements. Generally, the choice of two different notions, one for the object and another for its complement is related to avoiding certain paradoxes [11].

DEFINITION 1.1 *Given an 8-connected digital set $\mathcal{K} \subseteq \mathbb{Z}^2$. \mathcal{K} is called a digital 8-curve whenever each point $x \in \mathcal{K}$ has exactly two 8-neighbors in \mathcal{K} with the possible exception of at most two points, the so-called end points of the curve, having exactly one neighbor in \mathcal{K} .*

A curve without end points is termed a closed curve.

Each digital 8-curve can be ordered (or oriented) in a natural manner. Let us consider an ordered finite 8-curve $\mathcal{K} = (\kappa_1, \dots, \kappa_n)$, where κ_i is an 8-neighbor of κ_{i+1} for $i = 1, 2, \dots, n - 1$. Then the curve \mathcal{K} can be described by means of a simple compact ordered data structure containing the coordinates of κ_1 and a sequence of code numbers in $\{0, 1, \dots, 7\}$ indicating for each point of \mathcal{K} which of its neighbors will be the next point on the curve. This data structure was proposed by Freeman [5] and is known as the *chain code*.

DEFINITION 1.2 *Given an ordered digital 8-curve $\mathcal{K} = (\kappa_1, \dots, \kappa_n)$. For $k \in \{0, 1, \dots, 7\}$ the curve \mathcal{K} is called a $(k, k + 1(\text{mod } 8))$ -curve whenever the chain code representation of \mathcal{K} consists exclusively of two chain codes k and $k + 1(\text{mod } 8)$.*

In 1987, Scherl [16] introduced so-called *shape descriptors* which are boundary points of an 8-connected digital set belonging to local extrema of linear functionals corresponding to the main directions in digital plane: $0^\circ, 45^\circ, 90^\circ, 135^\circ, 180^\circ, 225^\circ, 270^\circ$ and 315° . Hence, descriptors are segments of horizontal, vertical and diagonal grid lines. Descriptors belong to locally convex or concave parts of the set and termed *T-* or *S-descriptors*, respectively. In Figure 1, dark voxels represent descriptor points of a digital image.

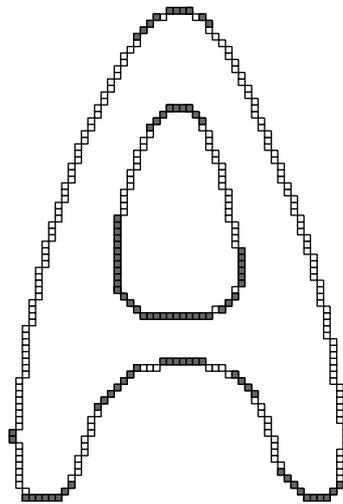


Figure 1. Scherl's descriptors of 'Letter A'.

The succession of descriptor points on the oriented boundary of a digital set is not arbitrary [3, 16]. It can be shown that the boundary of an 8-connected digital set can be decomposed into $(k, k + 1(\text{mod } 8))$ -curves such that the part

on the boundary between two such successive curves consists only of descriptor points [3].

Since each $(k, k + 1 \pmod{8})$ -curve is a translation of some $(0, 1)$ -curve in the later sections we may concentrate, without loss of generality, on $(0, 1)$ -curves.

2. Discrete Lines and Convexity

The arithmetical definition of discrete lines was introduced by J.-P. Reveillès [15].

DEFINITION 2.1 *A discrete line with a slope a/b , $b \neq 0$ and $\text{pgcd}(a, b) = 1$, lower bound μ , arithmetical thickness ω is the set of grid points which satisfies the double diophantine inequality*

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

with all integer parameters.

A (finite or infinite) subsequence of a discrete line is called a discrete line segment.

We note the preceding discrete line $\mathcal{D}(a, b, \mu, \omega)$. We are mostly interested in *naïve* lines which verify $\omega = \sup(|a|, |b|)$, we shall note them $\mathcal{D}(a, b, \mu)$. Without loss of generality, we may consider discrete lines under restrictions $a, b > 0$ and $a < b$, therefore $\omega = \max(a, b) = b$.

The real straight lines $ax - by = \mu$ and $ax - by = \mu + b - 1$ are respectively called *upper leaning line* and *lower leaning line* of $\mathcal{D}(a, b, \mu)$. The grid points which satisfy the mentioned equalities are called *upper* or *lower leaning points*. We remark that the distinction between lower and upper leaning points depends on the equation, there is here no geometrical invariancy. According to Proposition 2.2 (see below), we may note U_F (L_F) the upper (lower) leaning point of a discrete line segment whose x -coordinate is minimal. In the same way, we note U_L (L_L) the upper (lower) leaning point whose x -coordinate is maximal. In Figure 2 an example of a discrete line segment is given.

It can be shown [15] that a discrete line $\mathcal{D}(a, b, \mu)$ with slope $a/b \leq 1$ has exactly one grid point on each vertical line. If $a/b < 1$ then the intersection between $\mathcal{D}(a, b, \mu)$ and any horizontal line is composed by $[b/a]$ or $[b/a] + 1$ successive grid points, where $[]$ means the integer part.

A discrete line $\mathcal{D}(a, b, \mu)$, where $0 < a < b$, satisfies the chord property and is 8-connected [1, 15]. It follows that a discrete line segment is a digital straight line segment in sense of Huebler, Klette, Voss [7]. The finite digital curves with the chord property are discrete line segments. There are infinite digital curves which satisfy the chord property and, however, are not discrete lines [6].

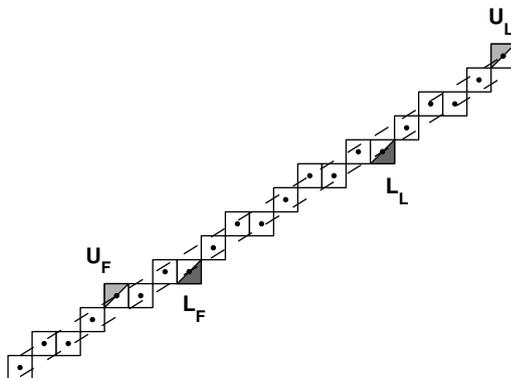


Figure 2. Discrete line $\mathcal{D}(5, 8, -4)$.

PROPOSITION 2.1 *A discrete line $\mathcal{D}(a, b, \mu)$, $0 < a < b$ is an 8-curve.*

Proof Without loss of generality, we may restrict our considerations to discrete lines $\mathcal{D}(a, b, 0)$. The discrete line $\mathcal{D}(a, b, 0) = \{(x, y) \in \mathbb{Z}^2 \mid 0 \leq ax - by < b\}$ is 8-connected. It holds $N_2(x, y) = (x, y + 1)$ and $ax - b(y + 1) = ax - by - b < b - b = 0 \notin \mathcal{D}(a, b, 0)$. Analogously, it can be easily shown that $N_3(x, y), N_6(x, y), N_7(x, y) \notin \mathcal{D}(a, b, 0)$.

Let $(x, y) \in \mathcal{D}(a, b, 0)$ with $ax - by = 0$. It holds $N_1(x, y), N_4(x, y) \notin \mathcal{D}(a, b, 0)$ and $N_0(x, y), N_5(x, y) \in \mathcal{D}(a, b, 0)$. If $ax - by = b - 1$ then $N_0(x, y), N_5(x, y) \notin \mathcal{D}(a, b, 0)$ and $N_1(x, y), N_4(x, y) \in \mathcal{D}(a, b, 0)$.

By considering different locations of $(x, y) \in \mathcal{D}(a, b, 0)$ with $0 < ax - by < b - 1$, we can prove that (x, y) has exactly two 8-neighbors. They are either $N_0(x, y), N_4(x, y)$, or $N_1(x, y), N_5(x, y)$, or $N_0(x, y), N_5(x, y)$, or $N_1(x, y), N_4(x, y)$.

We deduce that $\mathcal{D}(a, b, \mu)$ is an 8-curve. \square

The previous result turned out that the movement from left to right along a discrete line with $0 < a < b$ occurs by using two translations, either $(x, y) \mapsto (x + 1, y)$ or $(x, y) \mapsto (x + 1, y + 1)$. We have shown:

PROPOSITION 2.2 *Each discrete line $\mathcal{D}(a, b, \mu)$, $0 < a < b$ is a $(0, 1)$ -curve.*

The term of convexity is a central subject of many geometrical investigations. Particularly, in the application oriented disciplines of geometry it plays an important role. The basic constructions of digital geometry are discrete lines, discrete line segments and digitally convex sets. They belong since beginning of the research to digital geometry to the frequently examined objects.

In Euclidean geometry, a set in \mathbb{R}^d is said to be convex if whenever it contains two points, it also contains the line segment joining them. Already in the two-dimensional case, there were observed difficulties by direct transfer of this definition into digital circumstances (see e.g. [6]). In the literature, there exist different types of digital convexity. The most common of them are studied in [8–10, 14].

Let $\mathcal{K} = (\kappa_1, \dots, \kappa_n)$ be a segment of a discrete line $\mathcal{D}(a, b, \mu)$. The problem to determine the convex hull of the elements of \mathcal{K} is solved in [2]. The convex hull of \mathcal{K} is a closed polygonal curve which can be subdivided into two polygonal curves joining κ_1 and κ_n : the *lower frontier* and *upper frontier* of the convex hull. It is clear that the lower leaning points belong to the lower frontier, upper leaning points belong to the upper frontier. How to detect all points which belong to the lower and upper frontier is shown in [2, Proposition 3, p.120]. Since the curve \mathcal{K} is a segment of a discrete line the intersection of \mathcal{K} and its convex hull consists only of elements of \mathcal{K} . This fact justifies the following concept of a *digitally convex curve*:

DEFINITION 2.2 *A digital curve \mathcal{K} is said to be lower digitally convex (upper digitally convex) if there is no grid point between \mathcal{K} and the lower (upper) frontier of the convex hull of \mathcal{K} .*

Algorithm SegConv from [2] can be applied for testing convexity of digital curves. This algorithm has the linear time complexity.

3. Fundamental Segments of 8-curves

DEFINITION 3.1 *Let $\mathcal{K} = (\kappa_1, \dots, \kappa_n)$ be a $(0, 1)$ -curve. A part $(\kappa_i, \dots, \kappa_j)$ is called a fundamental segment of \mathcal{K} whenever one of the following conditions is true:*

- 1 $i = 1, j = n$ and $(\kappa_1, \dots, \kappa_n)$ is a segment of $\mathcal{D}(a, b, \mu)$. Then \mathcal{K} consists of one single fundamental segment.
- 2 $i = 1, j < n$ and $(\kappa_1, \dots, \kappa_j)$ is a segment of $\mathcal{D}(a, b, \mu)$ such that $(\kappa_1, \dots, \kappa_{j+1})$ is not a segment of any discrete line. Here, $(\kappa_1, \dots, \kappa_j)$ is the first fundamental segment of \mathcal{K} .
- 3 $i > 1, j = n$ and $(\kappa_i, \dots, \kappa_n)$ is a segment of $\mathcal{D}(a, b, \mu)$ such that $(\kappa_{i-1}, \dots, \kappa_n)$ is not a segment of any discrete line. Here, $(\kappa_i, \dots, \kappa_n)$ is the last fundamental segment of \mathcal{K} .
- 4 $i > 1, j < n$ and $(\kappa_i, \dots, \kappa_j)$ is a segment of $\mathcal{D}(a, b, \mu)$ such that $(\kappa_{i-1}, \dots, \kappa_j)$ and $(\kappa_i, \dots, \kappa_{j+1})$ are not segments of any discrete line.

The fundamental segment $(\kappa_i, \dots, \kappa_j)$ will be denoted $\mathcal{F}(a, b, \mu)$.

This definition means that the convex hull of a fundamental segment of \mathcal{K} and the left or right added point consists at least of one grid point of the complement of \mathcal{K} [2, Remark 6]. Hence, fundamental segments are maximal possible subsets of \mathcal{K} belonging to discrete lines.

By definition, fundamental segments do not depend on the orientation of \mathcal{K} . All fundamental segments can be ordered in the sense of the oriented curve, we mark these $\mathcal{F}_i(a_i, b_i, \mu_i)$, $i = 1, \dots, m$. It is clear that two successive fundamental segments have always different slopes and their common part is not empty and it is always a segment of a discrete line. In addition, more than two fundamental segments can possess common parts of \mathcal{K} .

Clearly, the decomposition of a $(0, 1)$ -curve into fundamental segments is unique. The problem to find this decomposition is equivalent to the problem to determine subsets of the curve having constant tangents. It can be computed within linear time [4].

In Figure 3, fundamental segments of a digital $(0, 1)$ -curve are indicated.

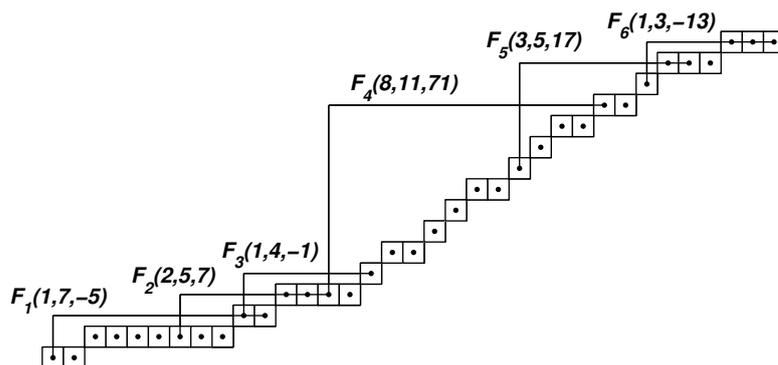


Figure 3. Fundamental segments of a digital curve. The first point of the curve is $(0, 0)$. Lower bounds μ_i , $i = 1, \dots, 6$ are computed with respect to $(0, 0)$.

4. Convex and Concave Curves

Fundamental segments allow an adaption of the concept *convex* and *concave curve* from continuous theory.

DEFINITION 4.1 Let \mathcal{K} be a $(0, 1)$ -curve and $\mathcal{F}_i(a_i, b_i, \mu_i)$, $i = 1, \dots, m$ be the successive fundamental segments of \mathcal{K} . The curve \mathcal{K} is called *convex* (*concave*) whenever the sequence of the slopes of fundamental segments is *increasing* (*decreasing*), i.e. $\frac{a_j}{b_j} < \frac{a_{j+1}}{b_{j+1}}$ ($\frac{a_j}{b_j} > \frac{a_{j+1}}{b_{j+1}}$), $1 \leq j \leq m - 1$.

If \mathcal{K} is a discrete line segment then it is convex as well as concave.

Since the sequence of fundamental segments does not depend of the orientation of \mathcal{K} , the concave curve is a convex one if the orientation of \mathcal{K} is turned back. In the following considerations, we will prove only convex case, the concave case can be formulated analogously and shown by duality.

Leaning points of fundamental segments of convex curves are not located arbitrary. Namely, they appear in the successive order on the curve. This statement is proved in the next proposition.

We mark the x - and y -coordinate of a point P as x_P and y_P , respectively.

PROPOSITION 4.1 *Let \mathcal{K} be a convex $(0,1)$ -curve and $\mathcal{F}_i(a_i, b_i, \mu_i)$, $i = 1, \dots, m$ be fundamental segments of \mathcal{K} . Then $x_{L_{L_j}} \leq x_{L_{F_{j+1}}}$ for all $1 \leq j \leq m - 1$.*

Proof Given fundamental segments $\mathcal{F}_j(a_j, b_j, \mu_j)$ and $\mathcal{F}_{j+1}(a_{j+1}, b_{j+1}, \mu_{j+1})$ of \mathcal{K} . Let Π be the polygonal set consisting of edges $e_1: a_j x - b_j y = \mu_j + b_j - 1$ and $e_2: a_{j+1} x - b_{j+1} y = \mu_{j+1} + b_{j+1} - 1$ which are lower leaning lines of $\mathcal{F}_j(a_j, b_j, \mu_j)$ and $\mathcal{F}_{j+1}(a_{j+1}, b_{j+1}, \mu_{j+1})$. One single vertex $V = (x_V, y_V)$ of Π is the intersection point of e_1 and e_2 . Obviously, \mathcal{K} is above Π . Since the sequence of the slopes is increasing, it holds $x_{L_{L_j}} \leq x_V$ and $x_V \leq x_{L_{F_{j+1}}}$. \square

In Figure 4 a convex curve with different locations of lower leaning points of fundamental segments is presented.

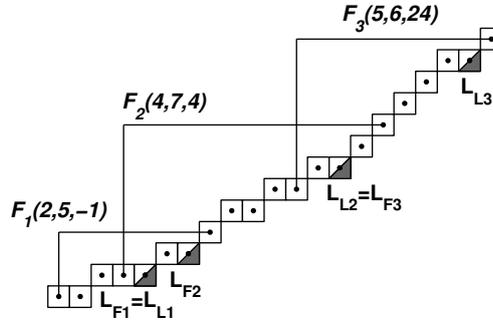


Figure 4. A convex curve and its lower leaning points of fundamental segments.

If for leaning points L_{L_j} and $L_{F_{j+1}}$ of $\mathcal{F}_j(a_j, b_j, \mu_j)$ and $\mathcal{F}_{j+1}(a_{j+1}, b_{j+1}, \mu_{j+1})$ of a convex curve \mathcal{K} holds $x_{L_{L_j}} = x_{L_{F_{j+1}}}$ then $L_{L_j} = L_{F_{j+1}}$ is a vertex of the lower frontier of the convex hull of \mathcal{K} . Before the situation $x_{L_{L_j}} < x_{L_{F_{j+1}}}$ will be examined, we introduce a concept of *supporting lines*.

A real line L is called a *lower supporting line in $P \in \mathcal{K}$* (briefly, LSL) if $P \in L$ and there exists a neighborhood $N(P)$ of P such that all elements in

$\mathcal{K} \cap N(P)$ are lying on or above L . All elements of a convex curve with a fundamental segment $\mathcal{F}(a, b, \mu)$ having leaning points L_F and L_L are always on or above the lower leaning line of $\mathcal{F}(a, b, \mu)$. Hence, $ax - by = \mu + b - 1$ is a LSL in L_F and L_L . Moreover, there exists no grid point between the segment (L_F, \dots, L_L) of \mathcal{K} and the real line segment $[L_F, L_L]$. If the whole curve \mathcal{K} is on or above a LSL then the LSL is also called a *global lower supporting line* (briefly, GLSL).

If one has an arbitrary $(0, 1)$ -curve \mathcal{K} with fundamental segments $\mathcal{F}_i(a_i, b_i, \mu_i)$, $i = 1, \dots, m$ and a GLSL in $P \in \mathcal{K}$ such that $P \in (L_{F_1}, \dots, L_{L_m})$ then it is clear that P is an element of the set $\{L_{F_1}, L_{L_1}, L_{F_2}, L_{L_2}, \dots, L_{F_m}, L_{L_m}\}$. Obviously, there can exist a GLSL in P which is before L_{F_1} or after L_{L_m} , however, this case does not play any role for our further considerations.

PROPOSITION 4.2 *Let \mathcal{K} be a convex $(0, 1)$ -curve and $\mathcal{F}_i(a_i, b_i, \mu_i)$, $i = 1, \dots, m$ be fundamental segments of \mathcal{K} . Let us assume $x_{L_{L_j}} < x_{L_{F_{j+1}}}$ for some $1 \leq j \leq m - 1$. Then there is no grid point between the polygonal set with vertices $L_{F_j}, L_{L_j}, L_{F_{j+1}}, L_{L_{j+1}}$ and fundamental segments $\mathcal{F}_j(a_j, b_j, \mu_j)$ and $\mathcal{F}_{j+1}(a_{j+1}, b_{j+1}, \mu_{j+1})$.*

Proof Sketch of the proof is given in Figure 5.

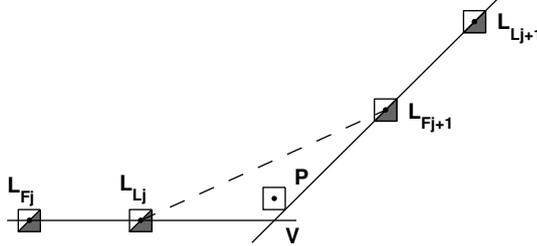


Figure 5. Sketch for Proposition 4.2.

The statement must be shown only for the part $(L_{L_j}, \dots, L_{F_{j+1}})$. Obviously, there is no grid point between $(L_{L_j}, \dots, L_{F_{j+1}})$ and the polygonal set with edges $a_j x - b_j y = \mu_j + b_j - 1$ and $a_{j+1} x - b_{j+1} y = \mu_{j+1} + b_{j+1} - 1$ and a vertex V .

Let us assume $P \in \mathcal{K}$ is the single point inside of the triangle with vertices $L_{L_j}, L_{F_{j+1}}, V$. It follows that there exists a GLSL in P . Hence, $P \in \{L_{F_1}, L_{L_1}, \dots, L_{F_m}, L_{L_m}\}$ and $P \neq L_{F_j}, L_{L_j}, L_{F_{j+1}}, L_{L_{j+1}}$. It leads to a contradiction that $\mathcal{F}_j(a_j, b_j, \mu_j)$ and $\mathcal{F}_{j+1}(a_{j+1}, b_{j+1}, \mu_{j+1})$ are successive.

Analogously, the case, where more than one points are inside of the triangle, leads to this contradiction. \square

Now we are able to show the equivalence between convex and lower digitally convex curves.

THEOREM 4.1 *Let $\mathcal{K} = (\kappa_1, \dots, \kappa_n)$ be a $(0, 1)$ -curve. The curve \mathcal{K} is convex if and only if \mathcal{K} is lower digitally convex.*

Proof Let $\mathcal{F}_i(a_i, b_i, \mu_i)$, $i = 1, \dots, m$ be fundamental segments of \mathcal{K} .

1. Let us assume \mathcal{K} is convex. We consider a polygonal curve Π consisting of vertices of the lower frontier of the convex hull of $\mathcal{F}_1(a_1, b_1, \mu_1)$ before L_{F_1} , intersection points of lower leaning lines of $\mathcal{F}_j(a_j, b_j, \mu_j)$ and $\mathcal{F}_{j+1}(a_{j+1}, b_{j+1}, \mu_{j+1})$, $j = 1, \dots, m - 1$ and vertices of the lower frontier of the convex hull of $\mathcal{F}_m(a_m, b_m, \mu_m)$ after L_{L_m} . Since \mathcal{K} is convex Π possesses increasing slopes and there is no grid point between Π and curve \mathcal{K} . By Proposition 4.1 for two successive fundamental segments $\mathcal{F}_j(a_j, b_j, \mu_j)$ and $\mathcal{F}_{j+1}(a_{j+1}, b_{j+1}, \mu_{j+1})$ holds $x_{L_{L_j}} \leq x_{L_{F_{j+1}}}$. If $x_{L_{L_j}} = x_{L_{F_{j+1}}}$ then $L_{L_j} = L_{F_{j+1}}$ is a vertex of Π . In the case $x_{L_{L_j}} < x_{L_{F_{j+1}}}$ for the slope s of the real line segment $[L_{L_j}, L_{F_{j+1}}]$ holds $\frac{a_j}{b_j} < s < \frac{a_{j+1}}{b_{j+1}}$. According to Proposition 4.2, there is no grid point between $(L_{L_j}, \dots, L_{F_{j+1}})$ and $[L_{L_j}, L_{F_{j+1}}]$. In this situation we modify the polygonal curve Π by replacing the vertex which is intersection point of lower leaning lines of $\mathcal{F}_j(a_j, b_j, \mu_j)$ and $\mathcal{F}_{j+1}(a_{j+1}, b_{j+1}, \mu_{j+1})$ by vertices L_{L_j} and $L_{F_{j+1}}$. Hence, modified Π has all successive vertices of the lower frontier of the convex hull of \mathcal{K} and there is no grid point between \mathcal{K} and the frontier.

2. We assume that there is no grid point between \mathcal{K} and the lower frontier of the convex hull of \mathcal{K} . In the case $m = 1$, the statement is, obviously, true. If $m > 1$ then the curve \mathcal{K} possesses at least two fundamental segments. It is clear that the points L_{F_1} and L_{L_m} are always vertices of the lower frontier.

Let us assume $m = 2$ and the slopes of $\mathcal{F}_1(a_1, b_1, \mu_1)$ and $\mathcal{F}_2(a_2, b_2, \mu_2)$ are decreasing. Then L_{F_1} and L_{L_2} are vertices of the lower frontier and there exists no another vertex between them. Since the slopes are decreasing, there must be at least one grid point between $(L_{F_1}, \dots, L_{L_2})$ and $[L_{F_1}, L_{L_2}]$ that leads to a contradiction.

For $m > 2$ the similar arguments lead to a contradiction when we assume that the slopes of $\mathcal{F}_j(a_j, b_j, \mu_j)$ and $\mathcal{F}_{j+1}(a_{j+1}, b_{j+1}, \mu_{j+1})$, $1 \leq j \leq m - 1$ are decreasing. \square

5. Decomposition of Curves into Meaningful Parts

Let us consider a digital $(0, 1)$ -curve \mathcal{K} . By means of fundamental segments of \mathcal{K} and their slopes, we are able to define convex and concave parts of \mathcal{K} which are maximal.

DEFINITION 5.1 *Let \mathcal{K} be a finite $(0, 1)$ -curve and $\mathcal{F}_i(a_i, b_i, \mu_i)$, $i = 1, \dots, m$ be fundamental segments of \mathcal{K} . A part consisting of successive fundamental*

segments $\mathcal{F}_u(a_u, b_u, \mu_u), \dots, \mathcal{F}_v(a_v, b_v, \mu_v)$, $1 \leq u \leq v \leq m$ is called a maximal convex part of \mathcal{K} whenever one of the following conditions is true:

- 1 $u = 1, v = m$ and $\frac{a_j}{b_j} < \frac{a_{j+1}}{b_{j+1}}, 1 \leq j \leq m - 1$.
- 2 $u \neq 1, v \neq m, \frac{a_{u-1}}{b_{u-1}} > \frac{a_u}{b_u}, \frac{a_v}{b_v} > \frac{a_{v+1}}{b_{v+1}}$ and $\frac{a_j}{b_j} < \frac{a_{j+1}}{b_{j+1}}$ for all $u \leq j \leq v - 1$.
- 3 $u = 1, v \neq m, \frac{a_v}{b_v} > \frac{a_{v+1}}{b_{v+1}}$ and $\frac{a_j}{b_j} < \frac{a_{j+1}}{b_{j+1}}$ for all $1 \leq j \leq v - 1$.
- 4 $u \neq 1, v = m, \frac{a_{u-1}}{b_{u-1}} > \frac{a_u}{b_u}$ and $\frac{a_j}{b_j} < \frac{a_{j+1}}{b_{j+1}}$ for all $u \leq j \leq m - 1$.

A maximal concave part of \mathcal{K} is defined in the same manner by replacing the signs ' $<$ ' and ' $>$ ' in the above definition.

It is clear that a convex (concave) curve consists exactly of one single maximal convex (concave) part. The parts of a curve \mathcal{K} overlap each other. If \mathcal{K} is neither convex nor concave then each of its maximal convex and concave part has at least two fundamental segments. The common component of two successive meaningful parts consists of exactly one fundamental segment.

Let us concentrate on Figure 6, where the curve from Figure 3 is represented again. The slopes of 6 successive fundamental segments are $\frac{a_i}{b_i} = 0.1429, 0.4, 0.25, 0.7273, 0.6, 0.3333$. We deduce that the curve possesses four maximal parts: convex consisting of $\mathcal{F}_1(a_1, b_1, \mu_1)$ and $\mathcal{F}_2(a_2, b_2, \mu_2)$; concave with $\mathcal{F}_2(a_2, b_2, \mu_2)$ and $\mathcal{F}_3(a_3, b_3, \mu_3)$; convex with $\mathcal{F}_3(a_3, b_3, \mu_3)$ and $\mathcal{F}_4(a_4, b_4, \mu_4)$; concave consisting of $\mathcal{F}_4(a_4, b_4, \mu_4)$, $\mathcal{F}_5(a_5, b_5, \mu_5)$ and $\mathcal{F}_6(a_6, b_6, \mu_6)$. The point P belonging to each maximal part is indicated.

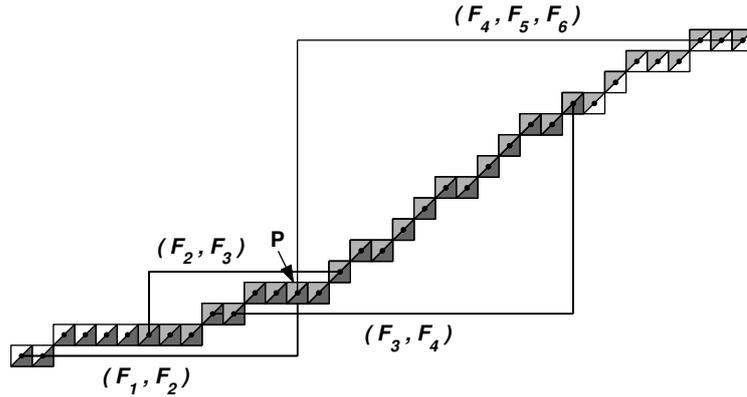


Figure 6. Maximal convex and concave parts of the curve from Figure 3.

We are interested in partition an arbitrary curve \mathcal{K} into maximal possible meaningful parts. First, we are able to decompose \mathcal{K} into $(0, 1)$ -curves using Scherl's descriptors (see Section 1). Next, the pre-decomposition into convex and concave parts can be defined in the following manner: the segment of \mathcal{K} between descriptor points is convex or concave whenever it is convex or concave part of the corresponding $(0, 1)$ -curve. Obviously, the pre-decomposition possesses no maximal possible parts. Finally, using the fact that T -descriptors belong to a convex part and S -descriptors to a concave part we can determine the maximal possible meaningful parts of the curve.

6. Fundamental Polygonal Representations of Digital Curves

In this section, we shortly discuss an important application of decomposition curves into fundamental segments. Let \mathcal{K} be a finite $(0, 1)$ -curve and $\mathcal{F}(a, b, \mu)$ a fundamental segment of \mathcal{K} . The (whole) segment $\mathcal{F}(a, b, \mu)$ is located above the lower leaning line $ax - by = \mu + b - 1$ and under the upper leaning line $ax - by = \mu$. Moreover, there is no grid point between the segment and the leaning lines. We consider two successive fundamental segments $\mathcal{F}_j(a_j, b_j, \mu_j)$ and $\mathcal{F}_{j+1}(a_{j+1}, b_{j+1}, \mu_{j+1})$. The common part of the segments is not empty and located above both lower leaning lines $a_jx - b_jy = \mu_j + b_j - 1$ and $a_{j+1}x - b_{j+1}y = \mu_{j+1} + b_{j+1} - 1$, also under both upper leaning lines $a_jx - b_jy = \mu_j$ and $a_{j+1}x - b_{j+1}y = \mu_{j+1}$. Hence, the segments $\mathcal{F}_j(a_j, b_j, \mu_j)$ and $\mathcal{F}_{j+1}(a_{j+1}, b_{j+1}, \mu_{j+1})$ are above (under) the polygonal curve Π with edges given by the mentioned real lines and the vertex given by their intersection point, respectively. There exists no grid point between the polygonal curves and the fundamental segments.

These considerations allow to introduce a concept of *fundamental polygonal representations*:

DEFINITION 6.1 *Let \mathcal{K} be a finite $(0, 1)$ -curve and $\mathcal{F}_i(a_i, b_i, \mu_i)$, $i = 1, \dots, m$ fundamental segments of \mathcal{K} . A polygonal curve Π with edges given by lower (upper) leaning lines of $\mathcal{F}_i(a_i, b_i, \mu_i)$ and vertices given by their intersection points in successive order is called lower (upper) fundamental polygonal representation of \mathcal{K} .*

Figure 7 shows the upper and lower fundamental polygonal representations of the curve from Figure 3.

We collect some simple properties of fundamental polygonal representations:

- 1 There is no grid point between fundamental polygonal representations and the digital curve.

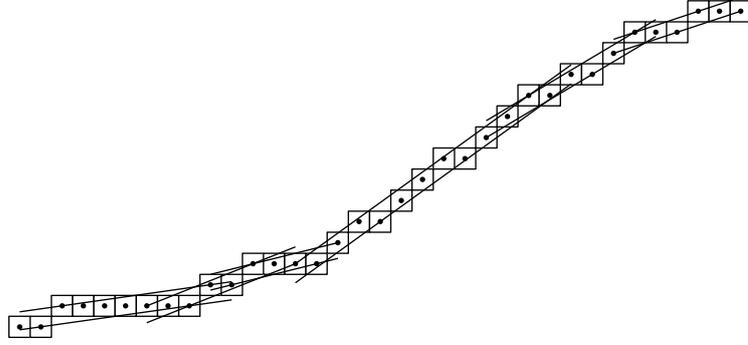


Figure 7. Fundamental polygonal representations of the curve from Figure 3.

- 2 Vertices of fundamental polygonal representations are, generally, not grid points.
- 3 The lower (upper) fundamental polygonal representation of a convex (concave) digital curve possesses only convex (concave) vertices.
- 4 The fundamental polygonal representations have the same convexity properties as the digital curve.
- 5 The representations are translations of each other and have the same Euclidean lengths.
- 6 If for lower (upper) leaning points of fundamental segments $\mathcal{F}_i(a_i, b_i, \mu_i)$, $i = 1, \dots, m$ of a convex (concave) curve holds

$$L_{L_j} = L_{F_{j+1}} \quad (U_{L_j} = U_{F_{j+1}}) \quad 1 \leq j \leq m - 1$$

then vertices of the lower (upper) fundamental polygonal representation between L_{F_1} and L_{L_m} (U_{F_1} and U_{F_m}) are vertices of the lower (upper) frontier of the convex hull of \mathcal{K} .

7. Conclusions

Discrete lines, discrete line segments and digitally convex sets are basic constructs of digital geometry. Using Scherl's descriptors each digital set can be decomposed into convex and concave parts by the method proposed here which is exact. This technique is related to the characterization of discrete lines by Debled-Rennesson and Reveillès [1].

An important application of this decomposition is the possibility to find a polygonal representation with the same convexity properties. The polygonal

representation of a set can be used as a basis for further simplification of the representing polygonal set by discrete evolution [13]. However, in spite of the precision, the disadvantage of the presented decomposition is the fact that the corresponding polygonal representation can possess vertices whose coordinates are not integers.

As alternative polygonal representations would be representations having “only few” uncorresponding convex or concave parts, but whose vertices are elements of \mathbb{Z}^2 . Moreover, such representations can be computed within linear time. Our group is preparing a further paper about linear algorithm for polygonal representations of digital sets.

References

- [1] Debled-Rennesson, I., Reveillès, J.-P.: A linear algorithm for segmentation of digital curves. *Int. J. Pattern Recognition, Artif. Intell.* 9, 635-662, 1995.
- [2] Debled-Rennesson, I., Rémy, J.-L., Rouyer-Degli, J.: Detection of discrete convexity of polyominoes. *Discrete Applied Mathematics* 125, 115-133, 2003.
- [3] Eckhardt, U. and Reiter, H.: Polygonal Representations of Digital Sets. *Algorithmica* 38, vol. 1, 2004.
- [4] Feschet, F. and Tougne, L.: Optimal time computation of the tangent of a discrete curve: application to the curvature. *DGCI'99, LNCS 1568*, pp. 31-40, 1999.
- [5] Freeman, H.: On the encoding of arbitrary geometry configurations. *IRE Trans. EC-10*, 260-268, 1961.
- [6] Huebler, Albrecht: Diskrete Geometrie fuer die Digitale Bildverarbeitung. Dissertation B, Friedrich-Schiller-Universitaet Jena, 1989.
- [7] Huebler, A., Klette, R., Voss, K.: Determination of the Convex Hull of a Finite Set of Planar Points Within Linear Time. *Elektronische Informationsverarbeitung und Kybernetik EIK* 17, 121-139, 1981.
- [8] Kim, C.E.: Digital convexity, straightness and convex polygons. *IEEE Trans. Pattern Anal. Mach. Intell. PAMI-4*, 618-626, 1982.
- [9] Kim C.E., Rosenfeld, A.: On the convexity of digital regions. *Pattern Recognition* 5, 1010-1015, 1980.
- [10] Kim C.E., Rosenfeld, A.: Digital straight lines and convexity of digital regions. *IEEE Trans. Pattern Anal. Mach. Intell. PAMI-4*, 149-153, 1982.
- [11] T. Y. Kong and A. Rosenfeld: Digital Topology: Introduction and Survey. *Computer Vision, Graphics, and Image Processing*, 48:357-393, 1989.
- [12] Latecki, L. and Lakaemper, R.: Convexity rule for shape decomposition based on discrete contour evolution. *Computer Vision and Image Understanding*, 73(3), 441-454, 1999.
- [13] Latecki, L. and Lakaemper, R.: Application of planar shape comparison to object retrieval in image databases. *Pattern Recognition*, 35:15-29, 2002.
- [14] Minsky, M., Papert, S.: *Perceptrons. An Introduction to Computational Geometry*. The MIT Press, Cambridge London, 1965.
- [15] Reveillès, J.-P.: *Géométrie discrète, calcul en nombres entiers et algorithmique*. Thèse d'État, Strasbourg, 1991.

- [16] Scherl, Wolfgang: Bildanalyse allgemeiner Dokumente. (Informatik-Fachberichte, Band 131), Springer-Verlag, Berlin, Heidelberg, New York, London, Paris, Tokyo, 1987.
- [17] Voss, K.: Discrete Images, Objects and Functions in \mathbb{Z}^n . Algorithm and Combinatorics, vol. 11, Springer-Verlag, Berlin, Heidelberg, New-York, 1993.

Detection of the Discrete Convexity of Polyominoes^{*}

Isabelle Debled-Rennesson^a, Jean-Luc Rémy^b and
Jocelyne Rouyer-Degli^c

^a*Institut Universitaire de Formation des Maîtres de Lorraine*

^b*Centre National de la Recherche Scientifique*

^c*Université Henri Pointcaré, Nancy I*

LORIA

(Laboratoire LOrrain de Recherche en Informatique et ses Applications)

Campus Scientifique, B.P. 239, F54506 Vandœuvre-lès-Nancy

{ debled,remy,rouyer}@loria.fr

Abstract

The convexity of a discrete region is a property used in numerous domains of computational imagery. We study its detection in the particular case of polyominoes. We present a first method, directly relying on its definition. A second method, which is based on techniques for segmentation of curves in discrete lines, leads to a very simple algorithm whose correctness is proven. Correlatively, we obtain a characterisation of lower and upper convex hulls of a discrete line segment. Finally, we evoke some applications of these results to the problem of discrete tomography.

Key words: Discrete Convexity, Segmentation, Discrete line, Polyominoes, Discrete tomography.

1 Introduction

Discrete convexity intervenes in numerous domains with regard to geometry and particularly to image processing [4]. It is an important property of plane figures which permits, for instance, methods for geometrical shapes regularisation. The notion of discrete convexity is strongly linked with the paradigm

^{*} Expanded version of a talk presented at the Discrete Geometry for Computer Imagery (Uppsala, December 2000).

of discrete lines. This is underlined in this article where a simple and efficient algorithm for detection of polyomino convexity is presented. Polyominoes are objects in which any couple of cells may be linked through a path containing only horizontal and vertical moves (4-connectivity). After having checked the hv-convexity (cells of each column and each row are consecutive) of a polyomino P , convexity is checked on the curve points of the border characterising it.

The proposed method uses a variant of the linear algorithm for segmentation of curves in straight lines given in [7,6]. The eventual non-convexity of P is detected by the algorithm, during its scanning of curves of the border of P ; if the whole border is scanned, P is convex. The proof of this algorithm uses a result on convex hull of a discrete segment presented in this article.

This technique and methods directly coming from definitions of discrete convexity have been used for the study of the reconstruction of 2-dimensional discrete sets from their horizontal and vertical projections. In particular, we studied the reconstruction of convex polyominoes. The approaches we use are presented at the end of this article. This work enters in the more general framework of discrete tomography whose applications are numerous, particularly in data compression, image processing, or still in medical imagery for medical diagnosis help in radiography.

In the second section, we introduce miscellaneous definitions of discrete convexity. Then a first simple method, deduced from a definition, is proposed. In the following section, fundamental elements of discrete geometry are given so that we obtain a very efficient algorithm for convexity detection on hv-convex polyominoes. Then, we propose a use of these methods in discrete tomography. At last, a conclusion and further research prospects are given.

2 Discrete Convexity

Convexity is well defined in the continuous case but in the discrete one, several definitions exist. The studied discrete figures are finite 8-connected subsets of discrete points in the plane and are named *discrete regions*.

In 1970, Slansky [18] defines a discrete region as convex if and only if there is a convex (Euclidean) region whose image (after digitizing) is this discrete region. This definition depends on the digitizing process used.

On the other hand, Minsky and Papert [14] gave the following definition of the convexity of a discrete region R : R is convex if and only if there is no triplet of colinear discrete points (c_1, c_2, c_3) such as c_1 and c_3 belong to R and

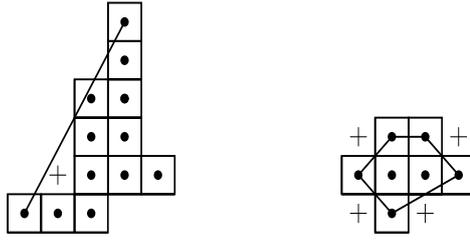


Fig. 1. On the left, a non-convex discrete region. On the right, a convex discrete region

c_2 belongs to the complementary of R .

Then, Kim and Rosenfeld [12,13,11] have given several equivalent definitions. They have shown that a discrete region R is convex

- on the one hand, if and only if its convex (Euclidean) hull does not contain any discrete point of the complementary of R .
- on the other hand, if and only if it fulfils the *area property* i.e. if and only if, for all points p_1 and p_2 of R , $P(R,p_1,p_2)$ does not possess any point of the complementary of R , where $P(R,a,b)$ represents the polygon whose edges are made by the segment ab and the edges of R (see Fig. 1).

These last two definitions shall be used in the study of the convexity of polyominoes. We refer to Kim and Rosenfeld [12,13,11] for a comparison of their definitions with Minsky and Papert's one.

A hv-convex polyomino is a polyomino whose cells of each column are consecutive (v-convexity) as well as those of each row (h-convexity).

Some immediate properties may be deduced from this definition:

- a hv-convex polyomino is a discrete region "without hole",
- a convex polyomino is hv-convex.

The convexity study of a polyomino so starts by checking its hv-convexity, this is done through a simple scanning of each row [4]. In the following, we study the convexity of hv-convex polyominoes. It is clear that this study may therefore be reduced to the convexity study of the limit or border of the hv-convex polyominoes (see Sect. 3 and 4).

3 Direct Use of the Area Property to Detect the Convexity of a hv-convex Polyomino

Let T be a hv-convex polyomino, anchored at (k, l) , i.e., containing in its first column a cell at row k and in the last column a cell at row l . We call left limit of T , and we note L , the set of cells with minimal column index of each line.

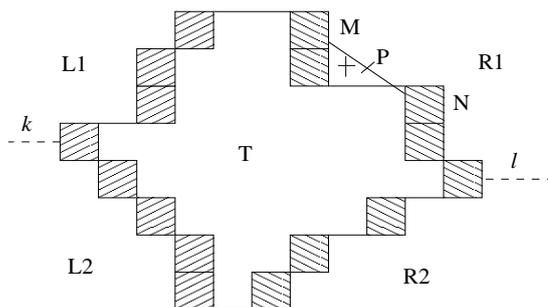


Fig. 2. Limits of the polyomino T

We define in the same way the right limit R of T. We distinguish in L the higher part L1, included between the first line and the left anchorage k , and the lower part L2, included between the line k and the last line. In the same way, we distinguish in R the parts R1 and R2 limited by the right anchorage l (see Fig. 2).

To check that T is convex, we only have to apply the area property on L1, L2, R1, and R2. So T is convex if and only if, for every couple (M, N) of points of L1, L2, R1, or R2, and every y included between the row indexes of M and N, there is no discrete point whose row index y is located between the segment MN (inclusive) and the polyomino T (exclusive). Let (x_1, y_1) be the coordinates of M, (x_2, y_2) the coordinates of N. The coordinates of the point P of the segment MN with row index y , $y_1 < y < y_2$, are (x, y) , with $x = x_1 + \frac{(y-y_1)(x_2-x_1)}{y_2-y_1}$. In order that T be convex, it is necessary and sufficient that:

- i. if M and N belong to L1 or to L2, the discrete point with coordinates $(\lceil x \rceil, y)$, located to the right of the point P, belongs to T,
- ii. if M and N belong to R1 or to R2, the discrete point with coordinates $(\lfloor x \rfloor, y)$, located to the left of the point P, belongs to T.

If the hv-convex polyomino T is represented by a 0-1 matrix whose dimension is $m * n$, the computation of its limits is in $O(m + n)$, m and n being the numbers of rows and columns of T, respectively, and the tests on couples of points are in $O(\min(m^3, n^3))$. Indeed, when $m > n$, it is permitted to exchange the roles of columns and rows.

4 Use of Discrete Lines to Detect the Convexity

Let us consider a hv-convex polyomino included in a minimal rectangle Rec whose size is $m * n$. Let $([A, A'], [B, B'], [C, C'], [D, D'])$ be the intersection between the border of the polyomino and Rec (see Fig. 3). By considering

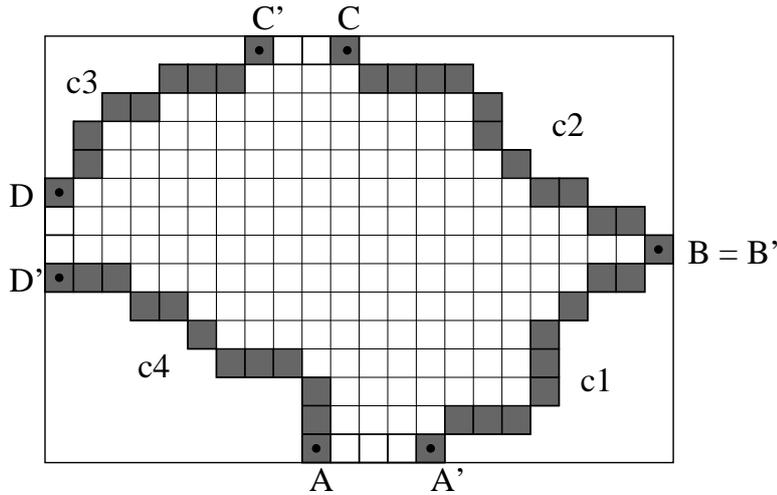


Fig. 3. A hv-convex polyomino, gray pixels represent the curves c_1 , c_2 , c_3 , and c_4

the pixels of the border, the points A , A' , B , B' , C , C' , D and D' delimit 8-connected curves of pixels which characterise the hv-convex polyomino. These 4 curves c_1 , c_2 , c_3 , and c_4 are made of the points of the polyomino border being respectively located between A' and B for c_1 , B' and C for c_2 , C' and D for c_3 and between D' and A for c_4 , respectively.

As it has been indicated in section 2, to check the convexity of a hv-convex polyomino, we only have to consider the border; curves c_1 , c_2 , c_3 , and c_4 must therefore be studied. The principle consists in segmenting curves c_1 , c_2 , c_3 , and c_4 , the eventual non-convexity of a polyomino shall be detected very simply through this operation.

In the following paragraph, the notion of discrete line [16,15,7] is recalled as well as some properties of these object types with, particularly, a new result on the construction of the convex hull of a discrete line segment. Moreover, an algorithm of discrete line segment recognition, with its use in the detection of convex polyominoes and its adaptation to the problem are presented.

4.1 Discrete Lines

The arithmetic definition of a discrete line was introduced by J.P. Reveillès [16,15,7]:

Definition 1 A discrete line with a slope $\frac{a}{b}$ with $b \neq 0$ and $\text{pgcd}(a,b)=1$, with lower bound μ , arithmetical thickness ω , is the set of points (x, y) of \mathbb{Z}^2 which

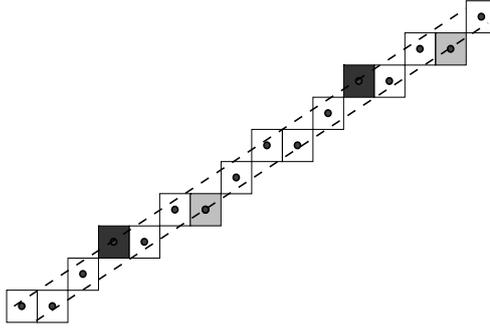


Fig. 4. A segment of the naïve line $\mathcal{D}(5, 8, -1)$ for $x \in [0, 15]$, the upper leaning points are in dark grey and the lower leaning points are in light grey, leaning lines are dotted lines

satisfies the double diophantian inequation

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

with all integer parameters.

We note the preceding discrete line $\mathcal{D}(a, b, \mu, \omega)$. We are mostly interested in naïve lines which verify $\omega = \sup(|a|, |b|)$ (see Fig. 4), we shall note them $\mathcal{D}(a, b, \mu)$. To simplify the writing, we shall suppose in the following that the slope coefficients verify $0 \leq a$ and $0 \leq b$, therefore $\omega = \max(a, b)$. Real straight lines $ax - by = \mu$ et $ax - by = \mu + \omega - 1$ are named the *leaning lines* of the discrete naïve line $\mathcal{D}(a, b, \mu)$. An integer point of these lines is named a *leaning point*.

The leaning line located above (resp. under) \mathcal{D} in the first quadrant ($0 \leq a$ and $0 \leq b$) respects the following equation $ax - by = \mu$ (resp. $ax - by = \mu + \omega - 1$), it is named *upper leaning line* (resp. *lower leaning line*) of \mathcal{D} . All points of \mathcal{D} are located between these two lines at a distance strictly lower than 1. It is clear that every segment of a discrete line is a convex region.

A naïve line \mathcal{D} may be seen as a set of the integer points taken on the union of real lines $ax - by = c$ with $c = \mu, \mu + 1, \dots, \mu + \omega - 1$. We name α -*levelled dotted line* of the line \mathcal{D} , the set of integer points located on the real line $ax - by = \alpha$.

Definition 2 $r(M)$ is named the remainder at point $M(x_M, y_M)$ with respect to \mathcal{D} and is defined by:

$$\mathbf{r}(\mathbf{M}) = \mathbf{a}\mathbf{x}_M - \mathbf{b}\mathbf{y}_M.$$

Proposition 3 Let M_0M_1 be a segment of the line $\mathcal{D}(a, b, \mu)$, the lower convex hull of the M_0M_1 points is the polygonal curve going through the following points:

- The first and the last lower leaning point of the segment, named L_F et L_L .

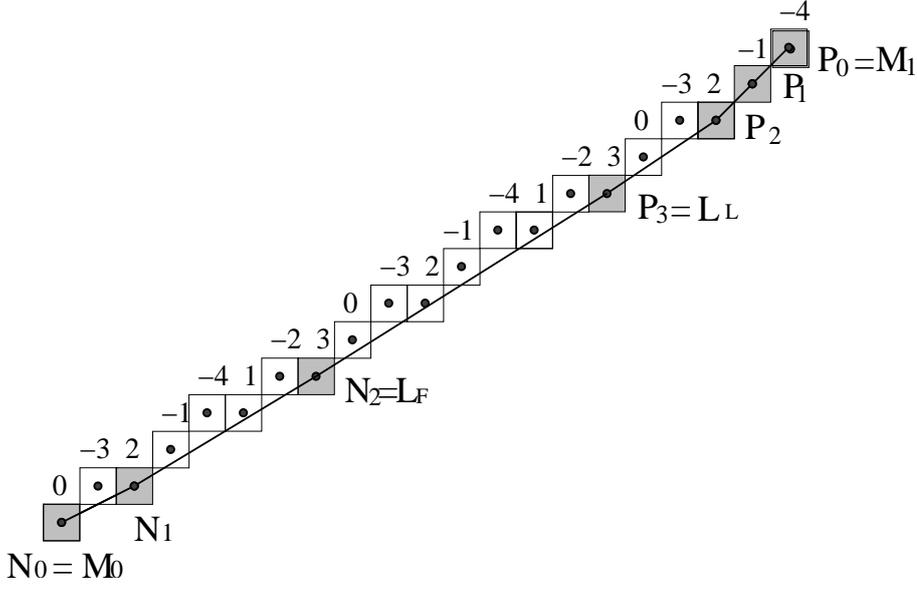


Fig. 5. Segment of the naïve line $\mathcal{D}(5, 8, -4)$, the lower convex hull of the segment is drawn, the value of the remainder is indicated for each point

- Between M_0 and L_F : the (N_i) sequence with $x_{M_0} \leq x_{N_i} \leq x_{L_F}$, $N_0 = M_0$ such that $x_{N_i} < x_{N_{i+1}}$ and $r(N_i) < r(N_{i+1}) \leq \mu + \omega - 1$
- Between M_1 and L_L : the (P_i) sequence with $x_{M_1} \geq x_{P_i} \geq x_{L_L}$, $P_0 = M_1$ such that $x_{P_i} > x_{P_{i+1}}$ and $r(P_i) < r(P_{i+1}) \leq \mu + \omega - 1$.

(see Fig. 5 for an example).

Remark 4

1. Colinear points of the N_i and P_i sequences may simply be omitted in the characterisation of the lower convex hull.
2. To construct the upper convex hull of a discrete line segment, the N_i and P_i sequences are determined in the same way from the upper leaning points.

Proof All points of \mathcal{D} are located above the lower leaning line, the segment points located between L_F and L_L are therefore above the segment $L_F L_L$ and the points L_F and L_L belongs to the lower convex hull of the segment.

Let us consider now the part of the segment between M_0 and L_F . Between these two points, we find at most $\omega - 2$ points, with all different remainders. Let N_i be a point of the sequence different of L_F , and N_{i+1} the first point encountered since N_i such that N_{i+1} belongs to the interval $[r(N_i), \mu + \omega - 1]$, the segment points between N_i and N_{i+1} are located above the dotted line going through N_i and therefore above the segment $N_i N_{i+1}$.

Let N_i, N_{i+1}, N_{i+2} , be three consecutive points of the sequence. Let us prove

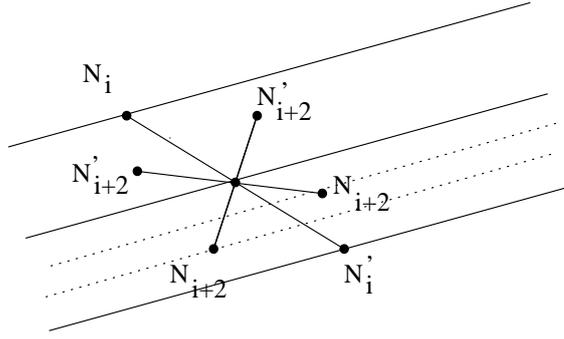


Fig. 6. Figure illustrating proof of Proposition 3 (Case 1)

that N_{i+2} is located above or on the straight line $N_i N_{i+1}$, thus we shall have demonstrated the convexity of the sequence of points N_i . Two cases must be studied:

Case 1. If $r(N_{i+1}) - r(N_i) \leq r(L_F) - r(N_{i+1})$ and $x_{L_F} \geq 2x_{N_{i+1}} - x_{N_i}$ (see Fig. 6), the point N_i' , symmetrical point of N_i with respect to N_{i+1} , is a point on the segment located between N_{i+1} and L_F . Let us suppose that N_{i+2} is a point located between the dotted lines of level $r(N_{i+1})$ and $r(N_i')$ such that $x_{N_{i+1}} < x_{N_{i+2}} < x_{N_i'}$. The symmetrical point N_{i+2}' of N_{i+2} with respect to N_{i+1} is a point of the segment whose remainder is included between $r(N_i)$ and $r(N_{i+1})$ and whose x-coordinate is included between those of N_i and N_{i+1} , which is contradictory with the hypotheses on N_i and N_{i+1} . Necessarily $N_{i+2} = N_i'$. Case $r(N_{i+1}) - r(N_i) \leq r(L_F) - r(N_{i+1})$ and $x_{L_F} < 2x_{N_{i+1}} - x_{N_i}$ may not occur (property related to Klein Theorem and discrete lines [6]).

Case 2. If $r(N_{i+1}) - r(N_i) > r(L_F) - r(N_{i+1})$, N_{i+2} is, by construction, a point located between the dotted lines of level $r(N_{i+1})$ and $r(L_F)$ such that $x_{N_{i+1}} < x_{N_{i+2}} \leq x_{L_F}$. Let us suppose that N_{i+2} is located under the $N_i N_{i+1}$ line, by using the symmetrical point of N_{i+2} with respect to N_{i+1} we also get a contradiction. A similar reasoning may be applied to the sequence (P_i) between M_1 and L_L . \square

4.2 Discrete Line Segment Recognition

Let us consider $\Sigma(M_0, M_1)$ a segment of \mathcal{D} , discrete naïve line with characteristics a, b, μ , $0 \leq a$ and $0 \leq b$, M_0 and M_1 are respectively the first and the last point of the segment. Let us suppose that the point $M(x_M, y_M)$ (with $M = M_1 + (1, 0)$, $M = M_1 + (1, 1)$ or $M = M_1 + (0, 1)$) is added to Σ . Is $\Sigma' = \Sigma \cup \{M\}$ a straight line segment and, if it is the case, what are its characteristics a', b', μ' ?

This problem is solved in [7,6] and relies on some particular leaning points of a discrete line. They are the leaning points located at the extremities of the segment currently being recognised. We note U_F (resp. L_F) the upper (resp. lower) leaning point whose x-coordinate is minimal. In the same way, we note U_L (resp. L_L) the upper (resp. lower) leaning point whose x-coordinate is maximal.

Theorem 5 ([7,6]) *Let us consider $r(M)$, the remainder at point $M(x_M, y_M)$ with respect to \mathcal{D} ($r(M) = ax_M - by_M$).*

- (i) *If $\mu \leq r(M) < \mu + \omega$, then $M \in \mathcal{D}(a, b, \mu)$, $\Sigma \cup \{M\}$ is a segment of the straight line \mathcal{D} .*
- (ii) *If $r(M) = \mu - 1$, then $\Sigma \cup \{M\}$ is a segment of the straight line whose slope is given by the vector $U_F M$.*
- (iii) *If $r(M) = \mu + \omega$, then $\Sigma \cup \{M\}$ is a segment of the straight line whose slope is given by the vector $L_F M$.*
- (iv) *If $r(M) < \mu - 1$ or $r(M) > \mu + \omega$, then $\Sigma \cup \{M\}$ is not a segment of a discrete line.*

Remark 6 *In the case (iv) M is called strongly exterior to \mathcal{D} and :*

- *If $r(M) < \mu - 1$ (M is above Σ), there is an integer point between $U_F M$ and Σ .*

- *If $r(M) > \mu + \omega$ (M is under Σ), there is an integer point between $L_F M$ and Σ .*

The existence of these points is demonstrated at page 644 of [7].

Algorithm 1

```

remainder =  $ax_M - by_M$  ;
if  $\mu \leq$  remainder  $< \mu + \omega$ 
then
  if remainder =  $\mu$  then  $U_L = M$  endif
  if remainder =  $\mu + \omega - 1$  then  $L_L = M$  endif
else if
  remainder =  $\mu - 1$ 
then
   $L_F = L_L$  ;
   $U_L = M$  ;
   $a = |y_M - y_{U_F}|$  ;
   $b = |x_M - x_{U_F}|$  ;
   $\mu = ax_M - by_M$  ;
else if
  remainder =  $\mu + \omega$ 
then
   $U_F = U_L$  ;

```

```

     $L_L = M ;$ 
     $a = |y_M - y_{L_F}| ;$ 
     $b = |x_M - x_{L_F}| ;$ 
     $\mu = ax_M - by_M - \sup(|a|, |b|) + 1 ;$ 
  else
    the new point may not be added to the segment
  endif
endif
endif
endif

```

AddPoint *adding a point M to the front extremity of a segment of a naïve line $\mathcal{D}(a, b, \mu)$ in the first quadrant.*

Theorem 5 allows us to obtain the incremental algorithm *AddPoint* of discrete line segments recognition by scanning a sequence of 8-connected pixels named *discrete path*. A linear segmentation algorithm for 8-connected curves [7,6] is immediately deduced from this result by considering the longest segments, last point of a segment being the first one of the next segment. To detect the convexity of a polyomino, we use a variant of this algorithm which is presented in the following section.

4.3 Use of a Segmentation Algorithm for the Detection of Convex Polyominoes

Detecting the convexity of a hv-convex polyomino P (see Fig. 3) consists in studying its convexity at the neighbourhood of each section of its borders c1, c2, c3, and c4. In this whole paragraph, convexity is described for the discrete curve c1, detection for the other curves is deduced by symmetry.

A curve is said *lower convex* if there is no discrete point between itself and its lower convex hull.

We must therefore prove that c1 is lower convex.

The algorithm for discrete line segments recognition is used on c1, it scans points of c1 one after another and stops when the added point may not belong to a discrete line segment containing all points already scanned plus the one added. The recognition shall not continue for a new segment at this added point but at the last lower leaning point of the segment which has just been recognised. When such a reject forces a segment change, the non convexity may be detected. The rejected point is located strongly above or strongly under the segment. Thanks to Remark 6, there is, in the second case, an integer point located between the discrete segment already scanned and the real segment from M to the first lower leaning point. This segment to which M

is added is therefore not lower convex and so neither is $c1$. In this case, the algorithm stops on a non-convexity statement of the polyomino.

Let us consider the procedure $\text{Recognizesegment}(M_0)$ whose input is a point M_0 of the curve $c1$ and which then adds the next points of $c1$ from M_0 as long as the scanned set of points is a discrete line segment. This procedure outputs M , the last point tested, which is either the rejected point, or the last point of $c1$, the characteristics (a, b, μ) of the scanned segment with a and b positive, and the last lower leaning point L_L as well as a boolean variable end , equal to false while $c1$ has still not been completely scanned.

Definition 7 *Let S be a discrete line segment obtained by the procedure Recognizesegment , we call reduced segment of S and we note S' , the segment containing the points of S located before the last lower leaning point existing in S , inclusive of this point.*

Algorithm 2

```

Input :  $c1$ , 8-connected sequence of points of the first quadrant
 $\text{convex} = \text{true}$ 
 $\text{end} = \text{false}$ 
 $\text{firstpoint} = \text{first point of } c1$ 
while  $\text{convex}$  and not  $\text{end}$  do
   $\text{Recognizesegment}(\text{firstpoint}) \rightarrow (M, a, b, \mu, L_L, \text{end})$ 
  if  $ax_M - by_M > \mu + \max(a, b)$ 
    then
       $\text{convex} = \text{false}$ 
    else
       $\text{firstpoint} = L_L$ 
    endif
  endwhile

```

SegConv testing convexity of the part of a polyomino in the first quadrant.

Theorem 8 *A curve $c1$ of the first quadrant is lower convex if and only if the SegConv algorithm completely scans $c1$ ($\text{convex}=\text{true}$ at the end).*

Proof Let us consider a point M which may not be added to the current segment with characteristics a, b, μ such that $ax_M - by_M > \mu + \max(a, b)$; then, according to Remark 6, there is an integer point between the current discrete segment and the straight line segment $L_L M$; therefore, according to the area property, $c1$ is not convex. In other words, when the algorithm stops because the variable convex has been set to false, $c1$ is not convex.

Let us suppose that $c1$ is completely scanned by the algorithm SegConv , then,

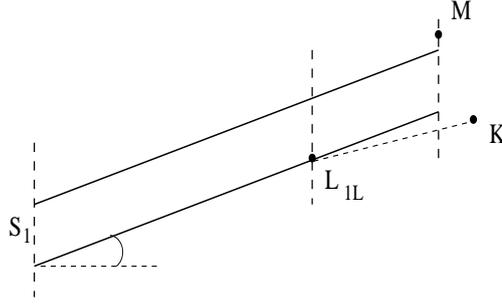


Fig. 7. Figure illustrating proof of Theorem 8

all changes of straight line segments have been done on points M which verify $ax_M - by_M < \mu - 1$. Each point is therefore located above the segment previously scanned.

Let $S1$ and $S2$ be two segments successively recognised by SegConv during the scanning of $c1$. $S1$ characteristics are a_1, b_1, μ_1 and its slope is $\alpha_1 (= \frac{a_1}{b_1})$. Let us consider L_{1L} the last lower leaning point of $S1$ and M the rejected point which does not belong to $S1$.

Let $S'1$ be the reduced segment of $S1$. Let us prove that the last edge of the convex hull of $S'1$ has a slope which is lower than the one of the first edge of the convex hull of $S2$.

As $S'1$ ends at the lower leaning point L_{1L} then, according to Proposition 3, if at least two lower leaning points are present in the segment, the slope of the last edge of the convex hull of $S'1$ is α_1 otherwise, the slope is lower than α_1 . Let $L_{1L}K$ be the first edge of the convex hull of $S2$, whose slope is β , we must consider two cases:

- If $K \in S1$, necessarily K is located above the lower leaning line of $S1$ therefore $\beta > \alpha_1$.
- If $K \notin S1$, let us suppose that $\beta \leq \alpha_1$, i.e., that K is under, or on, the lower leaning line of $S1$ (see Fig. 9). The points L_{1L} , M , and K belong to the segment $S2$. However that is impossible because the distance between M and the leaning line of $S1$ is greater than 1, and so, *a fortiori*, the one between M and $L_{1L}K$. Therefore these three points may not belong to the same discrete line segment and $\beta > \alpha_1$.

This property implies that the lower convex hull of the union of $S'1$ and $S'2$ is the union of the lower convex hulls of these both segments. As a generalisation, the lower convex hull of $c1$ is the union of convex hulls of the $S'i$. However each discrete segment is, by definition, lower convex. Therefore $c1$ is also lower convex, according to the definition of convexity by Kim and Rosenfeld (see Sect. 2) based on the convex hull of the set. \square

Complexity of the construction of curves $c_i, i = 1..4$, is in $O(m + n)$, m and

n being the numbers of rows and columns of the polyomino. Let us say a few words about the complexity of the SegConv algorithm. We first notice that the algorithm actually terminates. Indeed, at each step, the variable `firstpoint` is incremented by at least one element (in a non-void segment, L_L is always different of the first element). On the other hand, the segmentation algorithm described in [7,6] is linear in the number of points of the curve to scan. However, the variant we propose obliges to re-start every time at the last lower leaning point encountered, that leads to some overlaps. This algorithm is actually linear despite its overlaps. The proof of its linear complexity will be published in a next article.

5 Applications to Discrete Tomography

The objectives of discrete tomography is the reconstruction of a finite set of points from a number of projections. Several authors have been interested in this subject by using miscellaneous approaches to obtain exact, approached, or random solutions [9].

The particular case of polyominoes reconstruction from their horizontal and vertical projections is a NP-complete problem [19]. Moreover, if we suppose h-convexity (or v-convexity), the problems remains NP-complete [2]. On the other hand, if we consider hv-convex polyominoes, we get today several polynomial algorithms to decide if there is or not a polyomino corresponding to given projections [2,3,5]. The best existing algorithm is the one of Chrobak and Dürr [5]. This algorithm determines, for all couples of rows k and l , if there is a hv-convex polyomino anchored in k and l , with given horizontal and vertical projections. It is done by translating the existence of hv-convex polyominoes anchored in k and l as a 2SAT problem (conjunction of disjunctive clauses with at most two boolean variables), whose size is in $O(mn)$. A 2SAT problem may be solved linearly in its size [1] and the number of possible choices for k and l is $\min(m^2, n^2)$, as roles of rows and columns may be swapped. Therefore, it gives a complexity in $O(mn \min(m^2, n^2))$ for the reconstruction of a hv-convex polyomino with given vertical and horizontal projections. In fact, we may limit choices of k and l , and then obtain an even more efficient implementation [5].

If we add to the algorithm of Chrobak and Dürr a convexity test, by using any method presented above, the complexity of the algorithm remains in $O(mn \min(m^2, n^2))$. On the other hand, if we want to decide if there is a convex polyomino with given projections by first seeking the hv-convex polyominoes having these projections, we shall have at worst an exponential complexity, because we can have an exponential number of hv-convex solutions [8].

We also have started to seek for the convex polyominoes with given orthogonal

projections by using the paradigm of constraint programming. Let us give a short description of it:

- (1) A constraint program is the input
 - of *a priori domains* in which variables of the problem are authorised to take their values,
 - of *constraints* to which these variables are submitted,
 - and of a *strategy* of assigning these variables.
- (2) The running of a constraint program consists in the repetition until a solution emerges, or as long as possibilities exist, of the following iteration:
 - select, as a function of the prescribed strategy, a variable to assign, and an assignment value, among those in the domain of the variable which satisfy the constraints,
 - update all constraints related to the variable which has just been assigned.

This iteration is associated, in case of a dead end (none of the variables remaining for assignment may be assigned), with a backtracking process, back to the last choice point.

This approach had already been used for the detection of the existence of hv-convex polyominoes with given orthogonal projections [20]. We have [17] modified the program written at this occasion, by adequately enriching the set of constraints.

This new program has successfully run on a first sample of data. Moreover, we have observed the following: if we just take care of how to express the constraints, the adopted process allows us to take them into account globally, rather than successively. So, we avoid to apply first constraints related to the hv-convexity, and then the ones related to the convexity itself. This global treatment allows us to obtain convex polyominoes more quickly, when they exist, without having to enumerate previously the hv-convex polyominoes, which may be more numerous. And in the same way, it is possible to decide more rapidly that none of them exists, when it is the case.

These first results are encouraging and push us to deepen this approach.

6 Conclusion

Our work has permitted a step forward in two directions.

On the one hand, we obtain a characterisation of lower and upper convex hulls of a discrete line segment based on the arithmetic interpretation of these objects.

On the other hand, we obtain a linear algorithm to test the convexity of a polyomino, which is very simple to express. We prove the correctness of this algorithm. After this paper was prepared, the authors' attention was called by a reviewer to Hübler, Klette and Voß' paper [10]. We will compare their approach and ours in our oral presentation.

Moreover, we have explored an alternative approach of the convexity test, in particular by using the methods of constraint programming. We have also applied the convexity test to some problems of discrete tomography. In both cases, the first results obtained urge us to continue on this way.

Acknowledgements

We thank the members of the PolKA research group of LORIA for their review and constructive remarks.

References

- [1] B. ASPVALL, M. F. PLASS, and R. E. TARJAN. A linear-time algorithm for testing the truth of certain quantified boolean formulas. *Information Processing Letters*, volume 8, number 3, pages 121–123, 1979.
- [2] E. BARCUCCI, A. DEL LONGO, M. NIVAT, and R. PINZANI. Reconstructing convex polyominoes from horizontal and vertical projections. *Theoretical Computer Science*, volume 155, number 2, pages 321–347, 1996.
- [3] E. BARCUCCI, A. DEL LONGO, M. NIVAT, and R. PINZANI. Medians of polyominoes: A property for the reconstruction. *International Journal of Imaging Systems and Technology*, volume 8, pages 69–77, 1998.
- [4] J.-M. CHASSERY, A. MONTANVERT. Géométrie discrète en analyse d'images. *Traité des nouvelles technologies, série Images*, Hermes, 1991.
- [5] M. CHROBAK and C. DÜRR. Reconstructing hv-convex polyominoes from orthogonal projections. *Information Processing Letters*, volume 69, pages 283–289, 1999.
- [6] I. DEBLED-RENNESON. Etude et reconnaissance des droites et plans discrets. Thèse. Université Louis Pasteur, Strasbourg, 1995.
- [7] I. DEBLED-RENNESON and J.-P. REVEILLÈS. A linear algorithm for segmentation of digital curves. In *International Journal of Pattern Recognition and Artificial Intelligence*, volume 9, pages 635–662, Décembre 1995.

- [8] A. DEL LUNGO and M. NIVAT and R. PINZANI. The number of convex polyominoes reconstructible from their orthogonal projections. *Discrete Mathematics*, volume 157, pages 65–78, 1996.
- [9] G. T. HERMAN and A. KUBA. *Discrete Tomography*. Birkhauser, 1999.
- [10] A. HÜBLER, R. KLETTE and K. VOß. Determination of the Convex Hull of a Finite Set of Planar Points Within Linear Time. *Elektronische Informationsverarbeitung und Kybernetik*, pages 121–139, 1981.
- [11] C .E. KIM. Digital convexity, straightness and convex polygons. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume PAMI-4, pages 618–626, 1982.
- [12] C. E. KIM and A. ROSENFELD. On the convexity of digital regions. In *Pattern Recognition*, volume 5, pages 1010–1015, 1980.
- [13] C. E. KIM and A. ROSENFELD. Digital straight lines and convexity of digital regions. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume PAMI-4, pages 149–153, 1982.
- [14] M. MINSKY and S. PAPERT. Perceptrons. In *M.I.T. Press*, 1969.
- [15] J.-P. REVEILLÈS. Géométrie discrète, calculs en nombre entiers et algorithmique. Thèse d'état. Université Louis Pasteur, Strasbourg, 1991.
- [16] J.-P. REVEILLÈS. Structure des droites discrètes. In *Journées mathématique et informatique*, Marseille-Luminy, Octobre 1989.
- [17] I. SIVIGNON. Reconstruction de polyominos convexes par programmation par contraintes. Rapport du stage de Magistère d'Informatique 1ère année, ENS Lyon, effectué au LORIA, 1999.
- [18] J. SLANSKY. Recognition of convex blobs. In *Pattern Recognition*, volume 2, pages 3–10, 1970.
- [19] G. J. WOEGINGER. The reconstruction of polyominoes from their orthogonal projections. Technical report, Technische Universität Graz, 1996.
- [20] T. ZAJAC. Reconstructing Convex Polyominoes using Constraint Programming. Master Thesis, Wroclaw University of Technology, Pologne.