



Modélisation, Analyse, Représentation des Images Numériques Approche combinatoire de l'imagerie

Christophe Fiorio

► To cite this version:

Christophe Fiorio. Modélisation, Analyse, Représentation des Images Numériques Approche combinatoire de l'imagerie. Traitement des images [eess.IV]. Université Montpellier 2, 2008. tel-01168561

HAL Id: tel-01168561

<https://hal-lirmm.ccsd.cnrs.fr/tel-01168561>

Submitted on 26 Jun 2015

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

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

Numéro d'identification :

ACADÉMIE DE MONTPELLIER

U N I V E R S I T É M O N T P E L L I E R I I

— SCIENCES ET TECHNIQUE DU LANGUEDOC —

Habilitation à Diriger les Recherches

présentée à l'Université Montpellier II

SPÉCIALITÉ	: Informatique
<i>Formation Doctorale</i>	: Informatique
<i>École Doctorale</i>	: Sciences pour l'Ingénieur

Modélisation, Analyse, Représentation des Images
Numériques

Approche combinatoire de l'imagerie

par

Christophe Fiorio

Soutenue le 26 novembre 2008 devant le Jury composé de :

M. Achille BRAQUELAIRE, Professeur, Univ. Bordeaux 1, LaBRI,	Rapporteur
M. Rémy MALGOUYRES, Professeur, Univ. d'Auvergne, LAIC,	Rapporteur
Mme Annick MONTANVERT, Professeur, Univ. Pierre Mendès France, Grenoble, Gipsa-Lab,	Rapporteur
M. Jean-Pierre GUÉDON, Professeur, Univ. Nantes, IRRCyN,	Examineur
M. Srečko BRLEK, Professeur, Univ. Quebec à Montréal, LACIM,	Examineur
M. Jean-Claude BAJARD, Professeur, Univ. Montpellier II, LIRMM,	Examineur

Sommaire

1	Introduction	1
1.1	Domaines de recherche	2
1.2	Mes contributions	4
1.2.1	Modélisation	4
1.2.2	Analyse	5
1.2.3	Représentation des images numériques	6
1.2.4	Mise en œuvre des modèles	8
2	Modèle formel des images : topologie numérique	9
2.1	Introduction	10
2.2	Espaces topologiques	11
2.3	Topologie numérique : pixels et connexité	13
2.4	Topologie interpixels et complexes cellulaires	21
2.4.1	Introduction	21
2.4.2	Connected Ordered Topological Space	22
2.4.3	Complexe cellulaire abstrait : approche “Kovalevsky”	27
2.4.4	Complexe polyédral : star-topologie et fonction d’illumination	30
2.4.5	Complexe convexe (ou polyédral)	32
2.4.6	Star-topologie	33
2.4.7	Fonction d’illumination	40
2.5	Synthèse et perspectives	44
3	Analyse combinatoire des images	47
3.1	Approche algorithmique de la segmentation	48
3.1.1	Union-Find	50
3.1.2	Critères d’homogénéité pour l’Union-Find	56
3.2	Représentation à base topologique des images de régions	64
3.2.1	Introduction aux cartes combinatoires	65
3.2.2	TGF et Carte des bords	67

3.2.3	Quelques perspectives de l'utilisation des cartes en imagerie	74
3.3	Conclusion	74
4	Modélisation géométrique discrète	77
4.1	Introduction	78
4.2	Plans discrets : une structure fondamentalement en 2 dimensions	81
4.3	Objets non linéaires : objets à épaisseur variable	84
4.4	Conclusion et perspectives	90
5	Conclusion et perspectives	91
	Bibliographie	97
A	Two linear time Union-Find stratégies for image processing [84]	127
B	Segmentation en régions avec critère statistique basée sur les inégalités de concentration [89, 90]	145
C	Topological model for two-dimensional image representation : definition and optimal extraction algorithm [55]	161
D	On some applications of generalized functionality for arithmetic discrete planes [19]	207

Chapitre 1

Introduction

1.1 Domaines de recherche

Trouver un titre qui synthétise mon domaine d'étude n'a pas été facile. En fait, le sous-titre « Approche combinatoire de l'imagerie » est à la fois le point commun de mes travaux, son fil conducteur, et surtout son inspiration première. L'idée de départ, initiée par Michel Habib qui m'a proposé ce sujet de thèse, était d'essayer d'améliorer les processus de segmentation utilisant le graphe d'adjacence des régions en s'appuyant sur les résultats de l'algorithmique de graphe. Ce point de départ s'est transformé, au fur et à mesure de l'avancée de mes travaux, en une approche globale de l'imagerie par des méthodes combinatoires et discrètes. Je pourrais donc reprendre à mon compte la citation suivante de Jean Françon, prononcée en 1996, en l'élargissant, au delà de la géométrie, à la vision par ordinateur :

« Why not work with discrete geometrical objects and discrete operations only? That is, to develop a purely discrete modelling methodology? ... This work is nowadays, not done; but I consider it can be done. ... I consider as a fundamental trend of discrete (or digital) geometry to produce the concept of a full discrete computer imagery. » [94]

Si aujourd'hui, le domaine de la géométrie discrète s'est bien implanté, et si on trouve de plus en plus de résultats algorithmiques et combinatoires en analyse d'images, à l'époque du début de ma thèse c'était assez rare, les méthodes issues du traitement du signal prédominaient largement.

Pourtant l'information qui est réellement disponible sur la rétine des yeux n'est qu'une collection de points, de même qu'une image est un ensemble de pixels. Que ce soit au niveau de l'œil ou de l'image, on ne dispose à la base que d'une information ponctuelle qui donne une indication quant à la quantité de lumière et la couleur qui ont été projetées à cet endroit. Considérer l'image comme un espace discret est donc finalement assez naturel. J'ai donc abordé l'analyse d'image comme un problème de partitionnement de l'ensemble des points de l'image, chaque partie, appelée région, devant correspondre à des zones significatives de l'image. Ce processus de segmentation, une des étapes initiales de tout système de vision, est basé sur une analogie avec le système visuel humain. En effet, selon la théorie, la plus communément admise et citée, expliquant ce phénomène, la psychologie Gestalt [135, 52, 191, 204], le processus visuel humain commence par construire des percepts en effectuant des *regroupements perceptifs* et une unification des informations présentes dans la scène. La reconnaissance en elle-même n'est que la réponse finale, résultant d'un processus d'interprétation nécessitant des connaissances permettant de lever toute ambiguïté. Clairement, nous nous situons avant cette étape de reconnaissance et notre problématique est donc bien le regroupement perceptuel des pixels.

Très vite, alors que je travaillais avec Philippe Charnier sur un outil de segmentation utilisant le graphe d'adjacence des régions [2], j'ai été convaincu que l'un des problèmes était le manque d'un modèle formel sur lequel s'appuyer permettant de spécifier correctement ce qu'on cherchait à obtenir. D'ailleurs, Radu Horaud et Olivier Monga [120, 121] avaient déjà indiqué qu'il était nécessaire d'établir une théorie de la vision par ordinateur. En fait, l'élaboration d'une théorie scientifique demande trois étapes :

1. énoncer la théorie, spécifier, et élaborer les concepts de base qui doivent exprimer le cadre formel à la base de la théorie ;
2. exprimer ces concepts sous forme mathématique ;
3. réaliser un ensemble expérimental qui permette de vérifier la théorie.

C'est cette démarche qui a guidé tous mes travaux : élaborer une théorie, définir un cadre formel, proposer un modèle mathématique, définir des algorithmes s'appuyant sur ce modèle, et si possible les mettre en œuvre dans le cadre d'une application. Partant souvent d'une problématique proche des applications, j'ai donc souvent fait des détours par la théorie. Ceci explique pourquoi j'ai étudié la topologie des espaces discrets ou la géométrie discrète alors que l'objectif initial était, et reste, l'analyse d'images. On retrouve dans ces allers retours entre la théorie et la pratique, le titre principal de ce document « Modélisation, Analyse, Représentation des Images Numériques » :

Modélisation des espaces images par une topologie adaptée aux espaces discrets (section 2.4.6), définition d'une approche interpixels (section 2.4), modélisation statistique des images (section 3.1.2) ;

Analyse des images grâce à des algorithmes combinatoires basés sur l'approche interpixels (section 3.1.1) et sur notre modèle statistique (section 3.1.2) ;

Représentation des images segmentées à l'aide d'un modèle combinatoire à base topologique (section 3.2.2), étude de la géométrie (discrète) afin de représenter les contours des régions et volumes segmentés (chapitre 4).

Ces trois thèmes, finalement imposés par la démarche que j'ai adoptée, forment les chapitres 2, 3, 4 de ce document. On s'apercevra, à leur lecture, qu'ils ne correspondent pas à une chronologie linéaire, et que j'ai navigué d'un thème à l'autre au gré des besoins et des collaborations. En fait, ce que je cherche à faire, c'est proposer des outils permettant une analyse complète de l'image, prête à être utilisée dans un processus de reconnaissance intégrant la connaissance liée à l'application. Et pour cela, je pense que :

Afin de permettre une analyse complète de l'image, il est indispensable de s'appuyer sur un modèle topologique et statistique des images pour aborder de manière cohérente et globale le traitement des images numériques, c'est-à-dire la segmentation, la modélisation des frontières et le calcul de la géométrie des objets identifiés.

La section 1.2 suivante peut s'apparenter à un guide de lecture de ce document, j'y présente brièvement mes contributions à ces trois thèmes de recherche.

1.2 Mes contributions

Les chapitres qui suivent présentent plus en détail ces résultats et, surtout, me permettent de donner mon point de vue sur le sujet. Cette section se contente juste d'indiquer, pour chaque thème évoqué ci-dessus, l'état de mes contributions et constitue en quelque sorte un guide de lecture de ce document.

1.2.1 Modélisation

Topologie des espaces images

Un des premiers thèmes que j'ai développé est la topologie des images numériques. Comme indiqué à la section 1.1, ce thème m'est apparu évident après un premier résultat d'analyse d'images [2] obtenu en collaborant avec Phillipe Charnier alors également en thèse. Nous manquions d'informations dans le graphe d'adjacence des régions pour traiter correctement tous les cas. À la recherche d'une structure permettant de mieux représenter les images segmentées (voir la section 1.2.3), j'ai pris conscience de l'importance de s'appuyer sur un espace topologique bien défini si l'on voulait pouvoir disposer d'une représentation adaptée. C'est ainsi que j'ai commencé à étudier la topologie des espaces discrets (voir le chapitre 2).

J'ai rapidement acquis la conviction que l'approche interpixels [81] est la plus générale, la plus simple, et la plus cohérente pour définir des espaces topologiques numériques adaptés aux images et aux traitements. Même si les éléments interpixels de l'image n'ont pas de représentation physique directe et ne sont que l'abstraction des liaisons topologiques entre les points de l'image, ils permettent justement de se détacher de la représentation informatique et assurent une cohérence aux espaces discrets que sont les images numériques. D'autre part, il semble évident que la recherche d'une topologie pour les espaces images se fait dans l'idée de se rapprocher de la topologie des espaces euclidiens. Un moyen simple semble donc de considérer notre espace image comme un espace quotient de l'espace euclidien. Je pense donc que l'approche quotient associée à celle d'interpixels et de complexe cellulaire (voir définition 2.15) permet de définir correctement une topologie adaptée aux espaces des images numériques, tout en permettant de modéliser les différents problèmes liés à l'analyse d'images. C'est ainsi que nous avons proposé la *topologie-étoile* [1] (voir la section 2.4.6). De nombreux travaux ont eu lieu sur le sujet de la topologie des images. Il est intéressant de noter que les propositions sont finalement assez proches les unes des autres, comme je l'explique à la section 2.5, et

une proposition de synthèse, s'appuyant sur la topologie-étoile, et intégrant les résultats des autres propositions, est donc à envisager.

Polyomino et tomographie discrète

Une autre façon de modéliser une image est de s'intéresser à l'objet représenté en considérant l'image comme binaire. Dans ce cas, l'objet peut être vu comme un polyomino. Cette modélisation, si elle n'est pas adaptée à l'analyse d'images en niveaux de gris ou couleur, est très utile pour coder un objet discret. Elle est aussi un bon modèle pour la tomographie discrète. C'est suite au séminaire de Maurice Nivat, auquel j'assistais, que je me suis intéressé à la question de calculer le plus efficacement possible le déplacement relatif d'un polyomino sur un autre afin d'obtenir une superposition maximale au sens du nombre de points communs. Cette question posée par M. Nivat n'avait pas alors de solution efficace. Nous avons proposé avec Gilles d'Andréa un algorithme optimal dans le cas des polyominos HV-convexes [59]. Nous avons même utilisé cet algorithme dans le cas d'une application afin de vérifier la qualité d'impression d'étiquettes sur des produits alimentaires [58]. Enfin, j'ai été invité à un workshop de tomographie discrète afin de parler du problème général de la superposition maximale de polyomino ou de polyocubes [83].

Ces travaux n'ont pas donné d'autres suites, car si ce sujet était passionnant, il s'éloignait trop de ma préoccupation d'analyse d'images. C'est pourquoi je n'ai pas développé ces résultats dans ce document. Le lecteur intéressé devra se reporter aux deux articles écrits pour l'occasion : [58, 59].

1.2.2 Analyse

Comme évoqué à la section 1.2.1 précédente, mon premier résultat en analyse d'images vient d'une collaboration avec Philippe Charnier [2]. Si j'ai activement participé à ce résultat, l'idée initiale était due à P. Charnier car partie intégrante de sa thèse [44].

Mes premières contributions personnelles correspondent aux algorithmes *Scanline* et *Mergesquare* [84], fruits de la collaboration avec Jens Gustedt, alors en post-doc au LIRMM. Le fait de voir l'image comme un espace discret m'a tout de suite amené à voir la segmentation comme un problème de partition d'ensembles, et c'est donc tout naturellement que je me suis intéressé à l'Union-Find. Avec Jens Gustedt, nous avons étudié cette structure avec un double objectif :

1. adapter ce problème en prenant en compte la contrainte de la connexité ;
2. étudier la possibilité de proposer une solution linéaire à l'Union-Find dans le cadre particulier des images.

Ce double but a été atteint et est formalisé par la proposition de deux algorithmes correspondant à deux stratégies différentes [84]. Nous avons continué l'étude théorique

en améliorant les performances pratiques par une optimisation de la gestion de la mémoire [85]. Cette optimisation de l'espace mémoire occupé nous a permis, par la suite, de proposer une extension à la segmentation d'images 3D de nos algorithmes [86]. Ces travaux sont présentés à la section 3.1.1.

Comme le lecteur le notera sans doute, nos principaux résultats dans ce domaine n'ont pas été publiés dans des revues ou conférences spécialisées en analyse d'images. En effet, si nos algorithmes sont très efficaces d'un point de vue temps de calcul, nos résultats pratiques en segmentation n'étaient pas meilleurs que les algorithmes existants ce qui nuisait à leur acceptation dans la communauté de l'analyse d'images. Nous étions donc confrontés directement au problème du passage de la théorie à la pratique. En fait, nos algorithmes ont besoin d'un critère de décision, qu'on appelle prédicat d'homogénéité, qui fixe les conditions de regroupement des pixels en régions. Jusqu'alors, Jens Gustedt et moi utilisions un critère simpliste, basé sur la différence des moyennes de niveaux de gris des régions considérées. Si ce critère ne donne pas de mauvais résultats, il ne permet pas non plus d'atteindre la qualité de segmentation obtenues avec des critères bien plus sophistiqués. Mais ces derniers étaient très coûteux en temps de calcul et la performance des algorithmes les utilisant n'étaient pas au rendez-vous. Il était cependant évident que si je voulais faire connaître nos résultats dans la communauté de l'analyse d'images, il me faudrait utiliser un prédicat d'homogénéité plus performant.

Avec Richard Nock, nous avons donc commencé à étudier un critère statistique plus pertinent, mais pouvant être utilisé dans nos algorithmes sans en dégrader les performances calculatoires. Fidèle à ma démarche, nous avons d'abord proposé un modèle statistique de génération des images, sur lequel nous avons bâti un critère d'homogénéité [88, 89, 90]. Ces travaux ont permis de faire connaître les algorithmes de segmentation basés sur l'Union-Find [84] et ont inspiré de nombreuses suites. J'en donne quelques unes à la section 3.1.2. Récemment j'ai entrepris avec André Mas d'améliorer ce critère en essayant de palier son principal défaut : la sursegmentation. Je présente en partie ces travaux à la section 3.1.2, le détail pourra être trouvé dans une publication encore en cours de rédaction.

1.2.3 Représentation des images numériques

Graphe topologique et cartes combinatoires

Comme précisé à la section 1.1 ci dessus, mon but est de proposer des outils permettant une analyse complète de l'image avant la phase de reconnaissance. Cette phase de reconnaissance, qui introduit de la connaissance, nécessite une structure de représentation évoluée, si possible de type graphe. C'est pourquoi je me suis intéressé à ce problème de représentation des images segmentées. En outre, après avoir travaillé sur le

graphe d'adjacence des régions [2], la structure habituellement utilisée, j'étais convaincu qu'il fallait chercher quelque chose qui permette de bien rendre compte de la topologie de l'image. Cette partie de mes travaux est présentée à la section 3.2.2. J'ai d'abord proposé une structure de multi-graphe planaire inspirée des cartes combinatoires, le *graphe topologique des frontières* [82, 3]. Cette proposition m'a permis d'établir une collaboration étroite avec Yves Bertrand, alors Chargé de Recherche au CNRS à Strasbourg, aujourd'hui Professeur des Universités à Poitiers, qui utilisait les cartes comme outils de modélisation géométrique des images. Nous sommes rapidement parvenus à la conclusion que, si mon algorithme d'extraction du graphe topologique [82] était performant et original, la structure de graphe topologique des frontières ne permettait pas une extension facile en dimension 3. C'est pourquoi, tout en gardant l'idée des précodes pour l'algorithme d'extraction, nous avons proposé une extension en dimension n , basée directement sur le modèle des cartes combinatoires [24]. Cette collaboration s'est ensuite concrétisée formellement par le co-encadrement en thèse de Guillaume Damiand, ce qui nous a permis d'affiner notre proposition pour le cas 2D et 3D [21, 22, 55]. Ces travaux ont donné de nombreuses suites, notamment par G. Damiand qui a poursuivi dans cette voie. On trouvera tous les détails et les citations aux sections 3.2.2, 3.2.2 et 3.2.3 consacrées aux modèles de représentation basés sur les cartes combinatoires.

Géométrie discrète

Différents thèmes évoqués ci-dessus, la topologie des espaces numériques, les polyominos et les modélisations à base topologique, sont des thèmes présents dans la communauté de géométrie discrète. J'ai donc participé, depuis le début, aux travaux de cette communauté. Dans la perspective de représenter au mieux les images, et surtout les contours des régions segmentées, je me suis donc très tôt intéressé à la problématique de la modélisation des objets discrets. Mais ce n'est qu'assez récemment que j'ai mené des recherches dans ce domaine. Ainsi, avec Damien Jamet, Valérie Berthé et Fabrice Philippe, nous avons étudié la structure des plans discrets, qu'ils soient rationnels ou non, fins ou épais. Nous avons ainsi pu prouver que les plans discrets possèdent la propriété de *fonctionnalité généralisée* [18, 19] et avaient donc bien une structure intrinsèquement 2D, quel que soit leur épaisseur. Je présente ce résultat à la section 4.2.

Ensuite, avec Jean-Luc Toutant, que j'ai encadré en thèse, nous nous sommes intéressés aux objets discrets non linéaires, et en particulier aux cercles et courbes [87, 91, 92]. L'originalité de ces travaux est liée à la notion d'épaisseur. En effet, ma conviction était que l'épaisseur constante proposée par J.-P. Réveilles pour les droites et plans discrets [195] n'était pas adaptée aux objets non linéaires. Nous avons donc mis en évidence qu'il est important de séparer l'expression analytique de la courbe, de celle de l'épaisseur, qui est en fait l'expression de la discrétisation que l'on choisit. Le fait de séparer l'expres-

sion analytique de la courbe de celle de l'épaisseur est important car cela nous permet de lier l'épaisseur, non seulement à la forme de la courbe, mais aussi à la topologie de l'espace considéré. Ces résultats sont présentés à la section 4.3. Le lecteur intéressé par encore plus de détails pourra, en plus des publications [87, 91, 92], se référer à la thèse de Jean-Luc Toutant [223] qui propose, en outre, la définition de normes de séparabilité permettant ainsi, en définissant l'épaisseur à partir de ces normes, de garantir que nos objets discrets sont séparants, ce qui est une propriété topologique importante.

1.2.4 Mise en œuvre des modèles

Plusieurs de mes contributions, citées ci-dessus, ont débouché sur des collaborations qui ont donné lieu à d'autres contributions dans des domaines d'application différents. Je présente ici ces résultats.

En 2003, j'ai établi une collaboration avec William Puech du LIRMM. L'idée était d'utiliser les résultats de la géométrie discrète pour essayer de proposer des algorithmes d'insertion de données cachées plus robustes aux transformations des images. Malheureusement, nous n'avons pas encore pu proposer quelque chose de concluant liant géométrie discrète et insertion de données cachées. Néanmoins, le point de vue discret, que nous avons apporté avec Jean-Luc Toutant, a permis de proposer des améliorations sur l'invisibilité des données insérées en adaptant la quantification aux données [196, 225, 224, 222].

Lors d'un séminaire de Andreas Dietz, présentant la problématique de la modélisation des molécules chimiques en informatique, je lui ai proposé d'essayer d'utiliser les cartes combinatoires pour cela. En effet, leur problème majeur était de coder l'arrangement spatial relatif des atomes, soit en quelque sorte la topologie des atomes. Cette collaboration a été fructueuse et a donné lieu à deux publications [69, 70].

Bien sûr, une application évidente de mes résultats se trouve dans l'analyse d'images. C'est ainsi que j'ai établi depuis plusieurs années une collaboration avec Gilles Rabatel du CEMAGREF sur la problématique de la segmentation d'images naturelles de feuilles. Dans un premier temps, nous avons, avec Benoît de Mezzo que nous co-encadrions en thèse, utilisé le graphe topologique des frontières afin de proposer un algorithme de reconnaissance de feuilles dans une scène végétale complexe [64]. Puis nous nous sommes intéressés à la segmentation d'images par une collaboration régions-contours. Là, c'est la structure Union-Find qui nous a permis d'arriver à un résultat concluant [108]. Dernièrement, avec Nathalie Gorretta que nous co-encadrons également, nous avons envisagé de proposer des outils de segmentation utilisant à la fois les informations spatiales et hyperspectrales. Des premiers résultats ont été publiés dernièrement [110, 109].

Chapitre 2

Modèle formel des images : topologie numérique

Préambule

La définition d'un modèle formel des images numériques est un problème qui s'est posé à moi au cours de ma thèse. Le thème de la topologie des images numériques y tient donc une part importante. C'est resté, depuis, l'une de mes préoccupations, même si je n'ai pas publié dernièrement dans ce domaine. En revanche, mes travaux concernant la représentation des images ou la segmentation des images tiennent toujours compte de cette problématique. C'est devenu en quelque sorte un fil rouge de mes travaux de recherche. Les résultats obtenus au cours de ma thèse et par la suite ([1, 81, 82]) l'ont été avec l'aide de Jean-Pierre Aubert qui m'a initié à la topologie. Bien sûr, Ehoud Ahronovitz qui m'encadrait alors a aussi été l'un des acteurs de ces résultats. En revanche les réflexions présentées ici sont personnelles et représentent donc ma vision de ce domaine à l'heure actuelle. Et si ce thème est présenté ici c'est parce que, d'après moi, il est indispensable de s'appuyer sur un modèle topologique cohérent des images si l'on veut espérer aborder de manière cohérente et globale le traitement des images numériques, c'est-à-dire la segmentation, la modélisation des frontières, le calcul de la géométrie des objets identifiés, afin de permettre une analyse complète de l'image. J'en ai également acquis la conviction que l'approche interpixels reste la plus générale, la plus simple, et la plus cohérente, pour définir des espaces topologiques numériques adaptés aux images et aux traitements associés. Tous mes autres travaux s'appuient d'ailleurs sur cette modélisation interpixels. Mais, même si cette thématique m'a accompagné tout au long de ses années, je n'ai paradoxalement que peu publié dans le domaine, occupé à publier dans mes autres thématiques faisant l'objet de collaborations. C'est pourquoi ce chapitre est le plus développé de ce document.

2.1 Introduction

Les premiers papiers de géométrie discrète pour les images numériques sont apparus très tôt (voir par exemple [95, 78]) et concernaient principalement le codage des lignes et des courbes. Très vite s'est alors posé le problème de la connexité des objets dans les images [197] et donc tout naturellement la question de la topologie des espaces discrets, ou plus précisément des espaces discrétisés [181, 180], c'est-à-dire des images informatiques.

En fait, les concepts, classiques en topologie, d'intérieur, de fermeture, de bord d'un objet, d'adjacence, de connexité, de courbe, de surface, de groupe fondamental, d'homotopie jouent un rôle essentiel dans les applications des domaines de l'imagerie tels que le traitement d'images, l'infographie, l'analyse d'images, la vision, etc... Le besoin de formaliser ces notions et de les adapter aux espaces particuliers que sont les images numériques s'est donc fait rapidement sentir et a été noté par différents au-

teurs [199, 130, 146, 149, 81, 94, 133, 134].

La topologie des images numériques, que j'appellerai *topologie numérique* en référence au terme anglais *digital topology* qui permet de bien la différencier de la topologie discrète qui a un sens mathématique particulier, cherche donc à définir de manière cohérente les notions d'adjacence, de connexité, point simple, d'équivalence topologique d'objets et donc les notions d'ouvert, de fermé, de groupe fondamental.

T.Y. Kong et A. Rosenfeld [143] définissaient ainsi la topologie numérique :

Digital topology is the study of the topological properties of image arrays. Its results provide a sound mathematical basis for image processing operations such as image thinning, border following, countour filling and object counting.

Dans ce chapitre j'exposerai mon point de vue concernant la définition des espaces topologiques propres aux images numériques. Pour cela, à la section 2.2, je commencerai par rappeler quelques définitions classiques en mathématique concernant les espaces topologiques. J'exposerai ensuite, à la section 2.3, l'approche originelle et classique de la topologie numérique basée sur l'adjacence entre pixels. J'expliquerai pourquoi j'ai estimé que cette approche n'était pas adaptée aux applications liées à l'analyse d'images et en particulier la segmentation en régions notamment, car ces approches se restreignent au cas binaire, adapté à l'analyse d'un objet, mais pas celle de régions, et surtout par l'incohérence des notions de frontières ou d'objets. La section 2.4 sera donc dédiée à la présentation des approches alternatives proposées par E. Khalimsky, R. Kopperman, P.R. Meyer (section 2.4.2), V.A. Kovalevsky (section 2.4.3) et enfin, à la section 2.4.6, la *star-topologie* que j'ai proposée dans ma thèse [1, 81]. Je concluerai, section 2.5, par l'exposé de mon approche de la topologie numérique et de l'axe de recherche que je privilégie aujourd'hui.

2.2 Espaces topologiques

Cette section présente les définitions de base concernant les espaces topologiques. Celles-ci sont générales et ne sont pas spécifiques aux espaces discrets des images numériques.

Définition 2.1 (Espace topologique). Un espace topologique $(E, \mathcal{O})$ où $\mathcal{O}$ est une collection de sous-ensembles de E appelés *ouverts* de E tel que :

- (A1) l'ensemble vide $\emptyset$ et E sont ouverts ;
- (A2) toute union d'ouverts est un ouvert ;
- (A3) toute intersection de deux ouverts est un ouvert.

À ces trois axiomes de base, se rajoute souvent un quatrième axiome dit axiome de

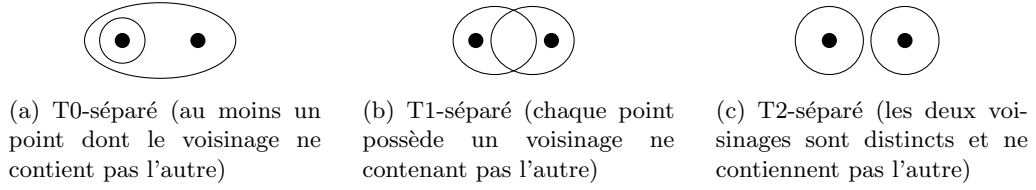


FIG. 2.1 – Axiomes de séparation.

axiomes de séparation. Soit $(E, \mathcal{O})$, un espace topologique, les trois axiomes de séparation les plus courants sont (voir la figure 2.1) :

T0 ou de Kolmogorov : pour tout couple de points distincts, il existe un *voisinage* (voir la définition 2.3 ci-après) de l'un qui ne contient pas l'autre point. L'espace $E = \{\mathbf{e}, \mathbf{f}\}$ dont les ouverts sont $\emptyset, \{\mathbf{e}\}, E$ est T0 mais pas T1.

T1 ou accessible ou de Fréchet : pour tout couple de points distincts, chaque point a un voisinage ne contenant pas l'autre point ;

T2 ou de Hausdorff ou séparé : pour tout couple de points distincts, chaque point admet un voisinage disjoint de l'autre, c'est-à-dire que la paire de voisinage a une intersection vide ; en corollaire, tout point $\mathbf{e} \in E$ est l'unique intersection de ses voisinages fermés.

Définition 2.2 (Fermé). Les *fermés* d'un espace topologique sont les complémentaires des ouverts, en conséquence de quoi $\emptyset$ et E sont des fermés de $(E, \mathcal{O})$.

À partir de ces deux notions de base sont définies plusieurs notions qui nous intéressent plus particulièrement dans le cadre des images numériques. Ainsi, par exemple, le *voisinage* (cf. la définition 2.3) permettra de définir la relation d'adjacence (définition 2.8) et par extension celle de connexité (définition 2.12), les notions d'*intérieur* (définition 2.4) et surtout de *frontière* (définition 2.6) seront utiles dans le cadre de la définition des régions ou des bords d'objet.

Définition 2.3 (Voisinage). Dans un espace topologique $(E, \mathcal{O})$, le *voisinage* $\mathcal{V}(e)$ d'un élément $\mathbf{e} \in E$ est un sous-ensemble contenant un ouvert contenant e . On notera en particulier que :

1. $E \in \mathcal{V}(e)$ (donc $\mathcal{V}(e) \neq \emptyset$) ;
2. $A, B \in \mathcal{V}(e) \Rightarrow A \cup B \in \mathcal{V}(e)$;
3. $A \in \mathcal{V}(e)$ et $A \subset B \subset E \Rightarrow B \in \mathcal{V}(e)$;
4. $\emptyset \notin \mathcal{V}(e)$.

On note $\mathcal{V}^*(e)$ le voisinage d'un élément privé de lui-même, cad $\mathcal{V}^*(e) = \mathcal{V}(e) \setminus \{\mathbf{e}\}$.

Définition 2.4 (Intérieur). Soit $(E, \mathcal{O})$ un espace topologique et $e \in X \subset E$, e est un *point intérieur* à X s'il existe $\mathcal{V}(e) \subset X$. L'*intérieur* de X est l'ensemble de tous les points intérieurs à X . Il est noté $\overset{\circ}{X}$.

On définit aussi l'*intérieur* de X comme l'union de tous les ouverts contenus dans X ou comme le plus grand ouvert contenu dans X .

Par conséquent, un ensemble $X \subset E$ est ouvert si et seulement si $X = \overset{\circ}{X}$ et $X \subset Y$ si et seulement si $X \subset \overset{\circ}{Y}$.

Définition 2.5 (Adhérence, Fermeture). Dans un espace topologique $(E, \mathcal{O})$, l'*adhérence* d'un sous-ensemble $X \subset E$ est le plus petit ensemble fermé de E qui contienne X .

L'adhérence de X est aussi appelée *fermeture* de X et se note $\overline{X}$.

On dit d'un point $\mathbf{x}$ de E qu'il est *adhérent* à X lorsque tout voisinage $\mathcal{V}(\mathbf{x}) \cap X \neq \emptyset$. L'*adhérence* de X est égale à l'ensemble des points qui lui sont adhérents. Intuitivement, l'adhérence d'une partie $X \subset E$ contient tous les points de l'espace qui sont dans X ou qui sont au « bord » de X .

Définition 2.6 (Frontière). Soit X un sous-ensemble d'un espace topologique $(E, \mathcal{O})$. On appelle *frontière* de X , et on note ∂X , l'ensemble de tous les *points frontières* de X , où un point $\mathbf{x} \in E$ est un *point frontière* de X si tout voisinage de $\mathbf{x}$ contient au moins un point dans X et un point en dehors de X .

On peut également définir ∂X comme l'adhérence sans l'intérieur ($\partial X = \overline{X} \setminus \overset{\circ}{X}$) ou comme l'intersection de l'adhérence et de l'adhérence du complémentaire ($\partial X = \overline{X} \cap \overline{X^c}$ où $X^c = X \setminus E$).

Les propriétés suivantes sont intéressantes dans le cadre des espaces images car elles nous permettront de définir précisément ce qu'est une région :

- $\partial X = \partial X^c$;
- la frontière ∂X d'un ensemble $X \subset E$ est un fermé;
- X est fermé si et seulement si $\partial X \subset X$;
- X est ouvert si et seulement si $\partial X \cap X = \emptyset$;
- l'adhérence d'un ensemble est l'union de cet ensemble et de sa frontière $\overline{X} = X \cup \partial X$.

Ces définitions ayant été rappelées, nous pouvons désormais aborder la topologie des images binaires.

2.3 Topologie numérique : pixels et connexité

Le terme de *Digital topology* apparaît pour la première fois dans l'article [199] de A. Rosenfeld. Sous ce terme, A. Rosenfeld regroupe les différentes propriétés topologiques

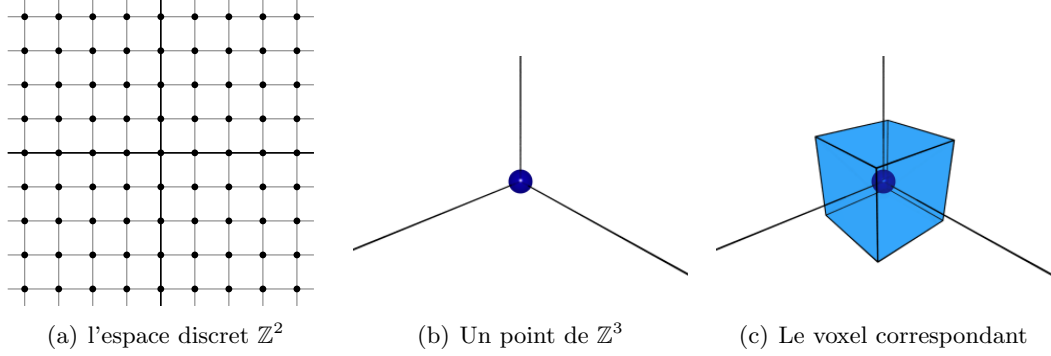


FIG. 2.2 – Pixels et voxels.

des espaces des images numériques. En fait, la première propriété étudiée fut la connexité des objets dans les images, ceci à des fins de suivi de contours pour la reconnaissance de caractères. Ainsi A. Rosenfeld et J.L. Pfaltz [202] en 1966, R.O. Duda et P.E. Harr et J.H. Munson [74] en 1967, et A. Rosenfeld en 1970 [197] ont fixé les bases de la connexité dans les images et de ce que l'on pourrait appeler l'“*école Rosenfeld*” de la topologie numérique (voir aussi [198] sur l'arbre des composantes connexes, [45, 157] pour un théorème concernant la 8-connexité et la topologie sur $\mathbb{Z}^2$ et enfin [143] pour un survol sur la topologie numérique des images binaires).

Cette approche consiste à considérer des images binaires et à étudier les propriétés topologiques des objets de ces images (c'est-à-dire les points noirs) à partir des pixels de l'image. Une image est donc considérée ici comme une matrice de points à coordonnées entières pouvant prendre deux valeurs, 0 pour un point blanc, 1 pour un point noir. Ces notions et définitions étant devenues des “standards”, il me semble important de les rappeler ici. Néanmoins, afin d'unifier les notations prévues à l'origine pour le cas $2d$ seulement, puis étendues au cas $3d$, et qui n'étaient donc pas adaptées au cas général de la dimension n , je les présente ici sous une forme plus moderne en rappelant à chaque fois l'appellation “historique”.

On considère ici une image comme un *espace discret*, c'est-à-dire une partie de $\mathbb{Z}^n$ où n est un entier naturel strictement supérieur à 1. On note $\mathbf{p}$ un point de $\mathbb{Z}^n$ et $(\mathbf{p}_1, \dots, \mathbf{p}_n)$ ses coordonnées. Nous nous intéressons dans cette section en particulier aux cas $n = 2$ et $n = 3$. Les points de $\mathbb{Z}^2$ sont appelés des *pixels* et les points de $\mathbb{Z}^3$ des *voxels*. Il est d'usage de représenter les pixels par des carrés d'arête 1 centrés en les points de $\mathbb{Z}^2$ (voir Figure 2.2.a) et les voxels par des cubes d'arête 1 centrés en les points de $\mathbb{Z}^3$ (voir Figure 2.2.a et 2.2.b). On note $\|\cdot\|_1$ et $\|\cdot\|_\infty$, les distances usuelles utilisées pour les espaces discrets des images (voir Définition 2.7).

Définition 2.7 (Distances). Soient deux points $\mathbf{p}$ et $\mathbf{q}$ de $\mathbb{Z}^n$ (c'est-à-dire à coordonnées

entières), les *distances* $\|\cdot\|_1$ et $\|\cdot\|_\infty$ sont définies comme suit :

$$\begin{aligned}\|\mathbf{p} - \mathbf{q}\|_1 &= \sum_{i=1}^n |\mathbf{p}_i - \mathbf{q}_i| \\ \|\mathbf{p} - \mathbf{q}\|_\infty &= \max(|\mathbf{p}_1 - \mathbf{q}_1|, \dots, |\mathbf{p}_n - \mathbf{q}_n|), 1 \leq i \leq n.\end{aligned}$$

La distance $\|\cdot\|_1$ est appelée *grid distance* [200] ou encore *distance de Manhattan* : elle correspond aux déplacements verticaux et horizontaux sur la grille. La distance $\|\cdot\|_\infty$ est appelée *lattice-point distance* [200], *distance de l'échiquier* ou encore *diamond distance* : elle correspond aux déplacements horizontaux, verticaux et diagonaux.

On définit alors le voisinage d'un point (pixel dans le cas $2d$, voxel dans le cas $3d$) dans une image, comme l'ensemble des points à distance 1. On remarquera donc ici une approche classique en topologie qui consiste à définir le voisinage à partir d'une métrique (bien que pour le cas discret, la notion de boule n'est pas explicitement définie) :

Définition 2.8 (κ -voisins, κ -voisinage). Soient $\mathbf{p} \in \mathbb{Z}^n$, $\mathbf{q} \in \mathbb{Z}^n$ et $\kappa \in \{0, \dots, n-1\}$. Les points $\mathbf{p}$ et $\mathbf{q}$ sont dits κ -voisins si :

$$\|\mathbf{p} - \mathbf{q}\|_\infty \leq 1 \text{ et } \|\mathbf{p} - \mathbf{q}\|_1 \leq n - \kappa.$$

- (i) L'ensemble des points de $\mathbb{Z}^n$, κ -voisins de $\mathbf{p}$ est appelé le κ -voisinage de $\mathbf{p}$ et est noté $\mathcal{V}_{n,\kappa}(\mathbf{p})$.
- (ii) On note $\mathcal{V}_{n,\kappa}^*(\mathbf{p})$ le voisinage de $\mathbf{p}$ privé de $\{\mathbf{p}\}$: $\mathcal{V}_{n,\kappa}^*(\mathbf{p}) = \mathcal{V}_{n,\kappa}(\mathbf{p}) \setminus \{\mathbf{p}\}$.

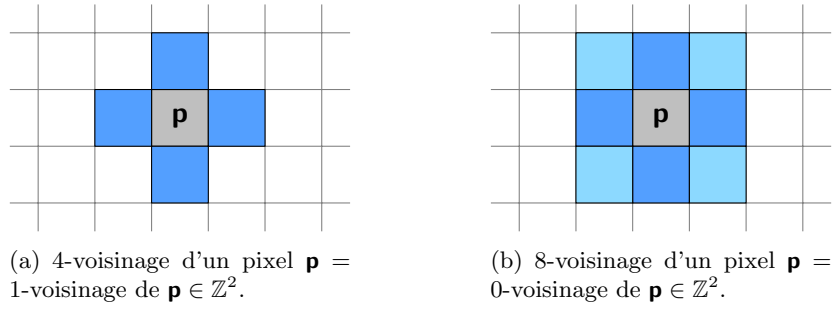
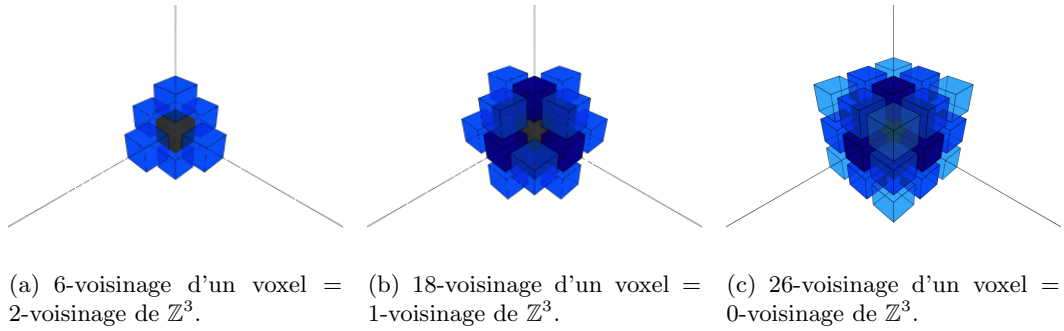
Si un pixel q appartient au κ -voisinage de p , on dit que p et q sont κ -adjacents ou tout simplement, lorsqu'aucune confusion n'est possible, *adjacents*.

Remarquons que pour $\kappa \in \{1, \dots, n-1\}$ et pour tout $\mathbf{p} \in \mathbb{Z}^n$, $\mathcal{V}_{n,\kappa}(\mathbf{p}) \subsetneq \mathcal{V}_{n,\kappa-1}(\mathbf{p})$. Autrement dit, si deux points $\mathbf{p}$ et $\mathbf{q}$ sont κ -voisins, alors ils sont $\kappa-1$ -voisins. Le lemme 2.1 permet de calculer le nombre de voisins d'un point dans une image en dimension n :

Lemme 2.1. *Le nombre $\#\mathcal{V}_{n,\kappa}^*(\mathbf{p})$ de voisins de $\mathbf{p}$ est :*

$$\forall n \geq 2, \forall \mathbf{p} \in \mathbb{Z}^n, \forall \kappa \in \{0, \dots, n-1\}, \#\mathcal{V}_{n,\kappa}^*(\mathbf{p}) = \sum_{i=1}^{n-\kappa} 2^i \binom{n}{i}.$$

En particulier, $\#\mathcal{V}_{2,0}(\mathbf{p}) = 8$, $\#\mathcal{V}_{2,1}(\mathbf{p}) = 4$, $\#\mathcal{V}_{3,0}(\mathbf{p}) = 26$, $\#\mathcal{V}_{3,1}(\mathbf{p}) = 18$ et $\#\mathcal{V}_{3,2}(\mathbf{p}) = 6$. On retrouve bien les définitions classiques et originelles des connexités en $2d$ et $3d$:

FIG. 2.3 – Les différents κ -voisinages d'un pixel dans $\mathbb{Z}^2$.FIG. 2.4 – Les différents κ -voisinages d'un voxel dans $\mathbb{Z}^3$.

Définition 2.9 (Voisinage $2d$). Dans une image $2d$, c'est-à-dire une partie de $\mathbb{Z}^2$, il existe deux voisinages :

- (i) le 1-voisinage, appelé le 4-voisinage : $\mathcal{V}_4(\mathbf{p}) = \mathcal{V}_{2,1}(\mathbf{p})$ (voir Figure 2.3(a)) ;
- (ii) le 0-voisinage, appelé le 8-voisinage : $\mathcal{V}_8(\mathbf{p}) = \mathcal{V}_{2,0}(\mathbf{p})$ (voir Figure 2.3(b)).

Définition 2.10 (Voisinage $3d$). Dans une image $3d$, c'est-à-dire une partie de $\mathbb{Z}^3$, il existe trois voisinages :

- (i) le 2-voisinage, appelé le 6-voisinage : $\mathcal{V}_6(\mathbf{p}) = \mathcal{V}_{3,2}(\mathbf{p})$ (voir Figure 2.4(a)) ;
- (ii) le 1-voisinage, appelé le 18-voisinage : $\mathcal{V}_{18}(\mathbf{p}) = \mathcal{V}_{3,1}(\mathbf{p})$ (voir Figure 2.4(b)) ;
- (iii) le 0-voisinage, appelé le 26-voisinage : $\mathcal{V}_{26}(\mathbf{p}) = \mathcal{V}_{3,0}(\mathbf{p})$ (voir Figure 2.4(c)).

La notion d'adjacence étant fixée, la notion de chemin peut alors être définie. Un chemin jouera en discret pour la connexité, le rôle des arcs des espaces de Hausdorff. Soit $\kappa \in \{0, \dots, n-1\}$. Les notions de κ -chemin et de κ -connexité d'une partie de $\mathbb{Z}^n$ se définissent alors comme suit :

Définition 2.11 (κ -chemin). Soit $\kappa \in \{0, \dots, n-1\}$, soit P une partie de $\mathbb{Z}^n$, soient $\mathbf{p}$ et $\mathbf{q}$ deux points de P . On appelle κ -chemin de $\mathbf{p}$ à $\mathbf{q}$ dans P , une suite finie ($\mathbf{p} =$

$\mathbf{p}_0, \dots, \mathbf{p}_m = \mathbf{q}$) de points de P tels que pour tout $i \in \{0, \dots, m-1\}$, $\mathbf{p}_i$ et $\mathbf{p}_{i+1}$ sont κ -voisins.

Le chemin étant pour le discret l’analogue de l’arc pour le continu, la connexité d’un ensemble se définit alors de manière classique comme suit :

Définition 2.12 (κ -connexité). Soit $\kappa \in \{0, \dots, n-1\}$. Une partie P de $\mathbb{Z}^n$ est dite κ -connexe si pour tous voxels $\mathbf{p}$ et $\mathbf{q}$ dans P , il existe un κ -chemin de $\mathbf{p}$ à $\mathbf{q}$ dans P .

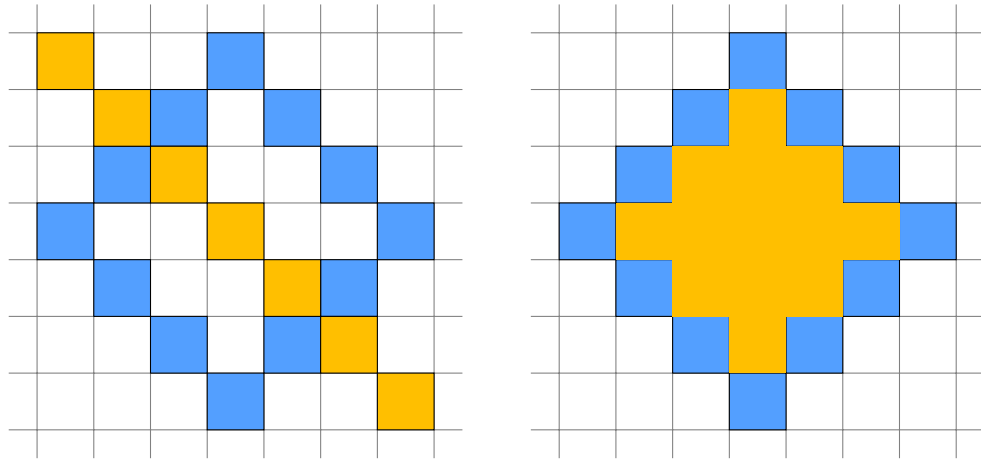
Nous dirons que l’ensemble P est *strictement κ -connexe* s’il est κ -connexe et si pour tout $\mathbf{p} \in P$, l’ensemble $P \setminus \{\mathbf{p}\}$ n’est pas κ -connexe.

Cette notion de connexité qui est une relation d’équivalence permet tout naturellement de définir celle de *composante connexe* :

Définition 2.13 (κ -composante connexe). Les classes d’équivalence d’une image (resp. d’un sous-ensemble P de points de l’image), définies par la relation de κ -connexité, sont appelées les *composantes connexes* de l’image (resp. de P). Si l’image (resp. P) ne possède qu’une seule composante connexe, l’image (resp. P) est dit *connexe*.

Définir ainsi les différentes notions de topologie à partir des éléments d’une image, c’est-à-dire ses points, paraît naturelle. De plus d’un point de vue pratique, les points de l’image étant les éléments représentés en machine, il paraît difficile de faire autrement. C’est sans doute pourquoi elle a été la première étudiée, et surtout pourquoi elle est encore couramment utilisée dans les applications. Néanmoins elle pose de nombreux problèmes et difficultés théoriques. Dès 1966, A. Rosenfeld et J.L. Pfaltz (voir [202]) signalèrent que certaines configurations de pixels amenaient à un paradoxe de connexité, que ce soit avec la 0-connexité ou la 1-connexité, paradoxe qu’on a appelé le *paradoxe de Jordan* (voir Figure 2.5) du nom du Théorème de Jordan qui stipule qu’une courbe fermée simple sépare le plan en deux composantes connexes distinctes, l’intérieur et l’extérieur. A. Rosenfeld [197] démontre un autre paradoxe lié aux caractéristiques d’Euler et, reprenant la suggestion de [74], propose d’utiliser en même temps les deux connexités, l’une pour le “fond” (les pixels blancs), l’autre pour l’“objet” (les pixels noirs). Il démontre ainsi qu’un 4-chemin fermé simple sépare l’espace en deux composantes non vides (l’intérieur et l’extérieur) et que tout 8-chemin reliant un point de l’extérieur à l’intérieur croise le 4-chemin. Ce faisant, il prouve ainsi une sorte d’analogue discret du Théorème de Jordan.

Mais le paradoxe de connexité n’est pas le seul problème que pose la topologie numérique basée uniquement sur les points de l’espace. Notamment dès que l’on passe à la dimension 3 ou que l’on veut généraliser à la dimension n , il devient très difficile de trouver une définition simple pour les objets discrets de base (surface, plan, courbes, ...) si l’on veut garantir certaines propriétés topologiques. Pour s’en convaincre il suffit de



(a) “Paradoxe de Jordan” pour la 8-connexité : le chemin orange (couleur claire) relie l’intérieur et l’extérieur sans croiser la courbe fermée simple bleue (couleur foncée).

(b) “Inverse du paradoxe de Jordan” : la composante orange (couleur claire) – l’intérieur – est séparée de la composante blanche – l’extérieur – par la courbe bleue (couleur foncée) qui n’est pourtant pas fermée.

FIG. 2.5 – Paradoxe de connexité en 2 dimension.

se référer à la littérature abondante sur le sujet (voir par exemple rien que récemment [235, 241, 47, 93, 212, 37, 91, 137, 32, 19]). En particulier, si l’on s’intéresse par exemple aux surfaces, R. Malgouyres [167] a montré qu’il n’y avait pas de caractérisation locale d’objets de $\mathbb{Z}^3$ séparant l’espace en deux composantes 6-connexes. Donc, définir une surface dans $\mathbb{Z}^3$ n’est pas aussi simple qu’il y paraît, à cause de la confusion entre un volume, un simple ensemble de voxels connexes, et une surface également ensemble de voxels connexes ! On est alors obligé de caractériser combinatoirement ces ensembles et plusieurs définitions existent, chacune privilégiant certaines propriétés topologiques de ces surfaces (voir par exemple [177], [20] ou récemment [34]).

Ces nombreux travaux et surtout ces nombreuses définitions pour le même objet discret démontrent la difficulté de définir simplement des objets discrets avec de bonnes propriétés topologiques dès lors que l’on veut tout définir à partir des points. En effet ces critères topologiques sont imprécis ou mal définis si on considère juste l’image comme un sous-espace de $\mathbb{Z}^n$ muni d’une simple relation d’adjacence basée sur les coordonnées des points. De plus, même en ne se basant que sur les points de l’image, plusieurs topologies sont possibles, 2 sur $\mathbb{Z}^2$, 5 sur $\mathbb{Z}^3$ [75] et 16 sur $\mathbb{Z}^4$ [139], ce nombre restant fini sur $\mathbb{Z}^n$ [139]. Les choses se compliquent encore lorsqu’on veut généraliser à la dimension n : choisir une bonne paire d’adjacence afin d’obtenir les propriétés topologiques souhaitées n’est alors pas chose aisée, d’autant que plusieurs sont admissibles (par exemple voir [34] pour une étude des bonnes paires d’adjacence en dimension n).

Un autre concept intéressant, car ouvrant des perspectives en terme de reconnaissance d'objets ou de squelettisation, est la possibilité de définir un *groupe fondamental* pour les images. T.Y. Kong, A.W. Roscoe et A. Rosenfeld [138, 142] ont montré qu'il était possible de définir un groupe fondamental numérique (*digital fundamental group*), analogue discret du groupe fondamental pour les topologies classiques (nous appelons topologies classiques les topologies de Hausdorff), mais sous certaines conditions. Plus précisément, les conditions imposées sur l'espace image et les adjacences entre points de l'image forment ce qu'ils nomment un *Espace Image Numérique Fortement Normal* – *Strongly Normal Digital Picture Space* – (voir les définitions 3.6.1 et 4.2.1 de [142] reprises ci-dessous). Il faut noter que l'on se trouve dans le cadre strict d'images binaires, ce qui n'est pas adapté aux images multi-niveaux (niveaux de gris, étiquetées, couleurs, ...) qui représentent actuellement l'immense majorité des applications. On remarquera surtout que leur espace impose des restrictions sur les adjacences de points noirs par rapport aux points blancs. Notamment, dans une configuration de 4 points sommets d'un quadrilatère, comme présentée Figure 2.6, si les deux points noirs en diagonale sont adjacents, alors les deux points blancs sur l'autre diagonale ne peuvent pas l'être.

Définition 2.14 (Digital Picture Space (DPS) [142]). Un Espace Image Numérique – *Digital Picture Space* – (V, β, ω) est un triplet où V est l'ensemble des points de la grille en $2d$ ou $3d$ et où β et ω sont un ensemble de segments de droite reliant les paires de points de V . β et ω définissent donc les deux relations d'adjacence de l'espace image.

Définition 2.1 (Définition 3.6.1 de [142]). Un $\text{DPS}(V, \beta, \omega)$ est dit régulier – *regular* – si il satisfait les deux conditions suivantes :

- (i) aucune β -adjacence ou ω -adjacence ne passe par un point de V autre que ses extrémités ;
- (ii) aucune β -adjacence ne rencontre une ω -adjacence avec laquelle elle ne partage pas une extrémité.

Définition 2.2 (Définition 4.2.1 de [142]). $\mathcal{S} = (V, \beta, \omega)$ est DPS fortement normal – *strongly normal* – si il est régulier et si il satisfait aussi les conditions suivantes de restriction de connexité :

- (1) $V = \mathbb{Z}^2$ (cas $2d$) ou $V = \mathbb{Z}^3$ (cas $3d$) ;
- (2) dans le cas $2d$ chaque 4-adjacence et, dans le cas $3d$ chaque 6-adjacence, est à la fois une β -adjacence et une ω -adjacence ;
- (3) toutes les β -adjacences et ω -adjacences sont des 8-adjacences dans le cas $2d$ et des 26-adjacences dans le cas $3d$;
- (4) dans tout carré unité, soit les deux diagonales sont des β -adjacences, ou soit les deux diagonales sont des ω -adjacences, ou alors l'une des deux diagonales est à la fois une β -adjacence et une ω -adjacence ;

- (5) chaque image $\mathcal{P}$ dans $\mathcal{S}$ a la propriété que chaque fois qu’une composante noire de $\mathcal{P}$ est β -adjacente ou bien ω -adjacente à une composante blanche de $\mathcal{P}$, la composante noire est 4-adjacente dans le cas $2d$ et 6-adjacente dans le cas $3d$ à une composante blanche.

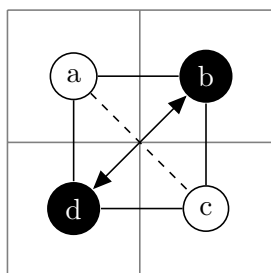


FIG. 2.6 – DPS régulier

Cette restriction de connexité (voir Définition 3.6.1 de [142] et Figure 2.6) est à comparer avec l’approche interpixels [1, 81] que nous avons développée (cf section 2.4) où nous disons qu’il faut fixer l’appartenance du pointel (sommet central entre les pixels) à l’une ou l’autre des composantes. Cela nous conforte donc dans l’idée que l’approche interpixels reste la plus générale, la plus simple, et la plus cohérente, pour définir des espaces topologiques numériques adaptés aux images et aux traitements associés.

En 2004, S. Fourey, T.Y. Kong et G.T. Herman [93], afin de généraliser l’approche par paire d’adjacence et d’éviter de devoir prouver chaque propriété topologique pour chaque paire d’adjacence possible, proposent d’aborder la topologie numérique par une définition générale à base d’axiomes de “*well-behaved digital spaces*”. En particulier ils introduisent une notion de structure discrète axiomatique de surface, un type général de structure discrète, défini à partir d’axiomes, qui modélise les sous-ensembles du plan Euclidien et d’autres surfaces. Cette structure n’est en fait qu’un complexe $2d$ très similaire aux complexes cellulaires. Ce complexe contient en particulier un ensemble $\pi \subseteq V^2$, V étant l’ensemble des points de l’image, de *proto-arêtes*. Ces proto-arêtes sont en quelque sorte des éléments interpixels, à rapprocher d’une arête fermée entre deux pixels ou voxels par exemple, munis de quelques conditions supplémentaires notamment liées à leur appartenance à des courbes fermées simples du complexe. Ces proto-arêtes servent à fixer la notion de connexité afin d’éviter les paradoxes. Bien que basé sur l’utilisation des paires d’adjacence, on retrouve donc ici le besoin (la nécessité ?) de s’intéresser à d’autres éléments de l’espace que les seuls points de l’image afin de garantir de bonnes propriétés topologiques. Là encore s’impose l’idée de se baser sur des espaces plus appropriés à la définition d’une bonne topologie numérique que le seul espace des points de l’image.

D’ailleurs, dans la foulée de ses travaux sur les relations d’adjacence sur $\mathbb{Z}^n$ (voir [139] déjà cité ci-dessus), T.Y. Kong en 2003 [140] affirme que des espaces topologiques qui permettraient que les concepts standards des topologies classiques (cad de Hausdorff) puissent être directement et utilement appliqués à une bonne numérisation doivent être des espaces de Khalimsky [130, 132], approche équivalente à celle de Kovalevsky [146] (voir la section 2.4 pour plus de détails) et que j’ai adoptée dès 1995 [1, 81]. Cette approche avait été signalée par T.Y. Kong et A. Rosenfeld dans leur survol de 1989 sur

la topologie numérique [143] comme une alternative intéressante à la topologie basée sur les pixels, notamment parce cette topologie est quotient naturel de la topologie usuelle sur les espaces euclidiens.

Néanmoins, parce que naturelle et plus étudiée, la topologie basée sur la notion de pixel ou voxel continue à être utilisée et étudiée, même si la tendance est à la baisse. Ainsi des auteurs comme L. Latecki essaient d'étendre les résultats de la topologie numérique, basée sur les pixels, à la problématique des images multi-niveaux (image de régions étiquetées, ou images en niveaux de gris) [159, 158]. Il définit pour cela des *images bien composées* :

Définition 2.3 (Définition 3.2 de [158]). *A multicolor digital picture is well-composed if for every color c , every 8-component of color c is also a 4-component.*

et il montre que :

Théorème 2.1 (Théorème 3.1 de [158]). *A multicolor digital picture is well-composed iff every component of every color is locally 4-connected.*

ce qui revient à fixer des conditions locales sur l'adjacence entre pixels de l'image, conditions parfaitement exprimées dans nos travaux [1, 81] par l'appartenance ou non du sommet, commun aux 4 pixels voisins, à une seule composante.

Les travaux de L. Latecki ont été repris par Y. Wang et P. Bhattacharya [232] dans le cadre d'images de régions et niveaux de gris. La définition des images bien composées est étendue afin de l'adapter à la problématique de la segmentation en régions et en particulier aux critères d'homogénéité qu'ils avaient proposés dans [230, 231]. On notera dans ces papiers, du point de vue de la topologie des images, l'évolution majeure consistant à ne plus définir les bords de régions comme des ensembles de pixels : les notions de bord et frontière reliés à la notion topologique euclidienne apparaissent clairement, mais sans faire référence explicitement à la notion d'interpixels.

Ceci vient renforcer l'intuition que j'avais eue en 1995, selon laquelle l'“*approche KKKM*” (de V.A. Kovalevsky [146], E. Khalimsky [130], R. Kopperman et P.R. Meyer [132]) offre plus de perspectives. Je vais maintenant la présenter plus en détail, ainsi que les solutions que nous avons proposées.

2.4 Topologie interpixels et complexes cellulaires

2.4.1 Introduction

Les concepts présentés ici reposent sur l'idée que considérer l'image comme un ensemble homogène de points ne permet de définir simplement une topologie pour laquelle

définir ou montrer des analogues discrets, aux concepts et propriétés classiques des topologies usuelles de l'espace euclidien, découle naturellement de cette définition. De plus, la topologie numérique, basée sur la connexité des pixels, impose bien souvent d'examiner les configurations locales de points. Or la combinatoire de ces configurations augmente rapidement dès que l'on monte en dimension pour les images, rendant de fait très difficile l'étude des objets discrets en dimension 3 ou supérieure.

Deux approches générales ont été étudiées :

- celle de E. Khalimsky [130], souvent appelée l'approche « E. Khalimsky, R. Kopperman, P.R. Meyer » (appelée approche *KKM* en référence à [132]), définit l'espace discret comme un produit de *Connected Ordered Topological Space*, un espace discret $1d$ basé sur une relation d'ordre total ;
- et celle basée sur la notion de complexe cellulaire (voir définition 2.15 proposée par V.A. Kovalevsky¹ [146] qui utilise la notion d'interpixels, éléments abstraits représentant les arêtes, les sommets et d'une manière générale les faces des éléments de l'image.

Bien que différentes, ces notions sont en fait très proches et il a été démontré par [118] que les espaces topologiques ordonnés peuvent être complétés par une fonction dimension qui en fait un complexe cellulaire. En fait le concept important commun à ces deux approches est que l'espace discret des images n'est plus considéré comme un espace homogène avec un seul type d'élément.

2.4.2 Connected Ordered Topological Space

L'article [11] de L.A. Ankeney et G.X. Ritter est, à ma connaissance, le premier papier abordant la topologie des images différemment de l'approche Rosenfeld basée uniquement sur les pixels. Leur idée était de s'appuyer sur des propriétés topologiques afin de paralléliser des opérations de filtrage ou de compression. Ils définissent le voisinage de chaque cellule (pixel dans le cas des images) en fonction de la parité de leurs coordonnées. Ils s'appuient sur le concept d'automate cellulaire de J. von Neumann [183] et de E.F. Moore [176] associant chaque pixel de l'image à une cellule de l'automate. Un traitement sur une image est alors défini comme un changement de l'état d'un pixel, ce changement tenant compte de l'état de ses pixels voisins, comme pour les automates cellulaires. Ils s'appuient alors sur des exemples d'espaces topologiques utilisés pour les automates cellulaires, définissant cette notion de voisinage et pouvant s'appliquer aux images. En particulier ils s'intéressent à la topologie de von Neumann :

Définition 2.4 (Définition topologie de von Neumann [11]). Soit $L = \{-1, 0, 1\}$ et $p = (p_1, \dots, p_k) \in \mathbb{Z}^k$. Le voisinage $N[c(p)]$ — ou simplement $N(p)$ — de $c(p) \in C^k$ où

¹G.T. Herman et D. Webster [119] avait déjà émis cette idée mais sans la développer comme V.A. Kovalevsky

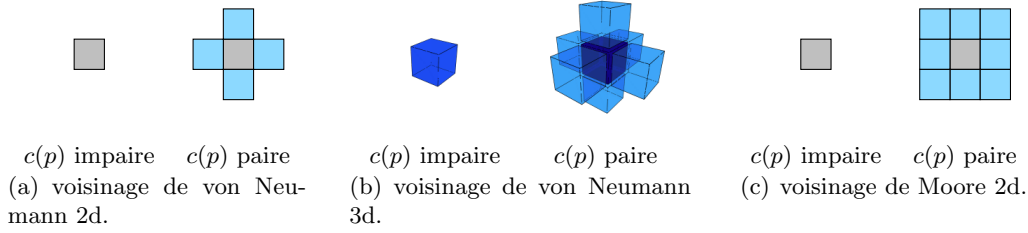


FIG. 2.7 – Voisinage de von Neumann et Moore

C^k est l'ensemble des cellules de dimension k et où $c(p) = \{q | q = (q_1, q_2, \dots, q_k), q_i \in \mathbb{R}, |p_i - q_i| \leq \frac{1}{2}, \text{ et } 1 \leq i \leq k\}$ est une cellule de dimension k pour $p \in \mathbb{Z}^k$, est défini par :

$$N(p) = \begin{cases} \{c(p)\} & \text{if } \sum_{i=1}^k p_i \text{ est impaire} \\ \{c(p_1, \dots, p_{i-1}, p_{i+n}, p_{i+1}, \dots, p_k), 1 \leq i \leq k, n \in L\} & \end{cases}$$

Un point dans l'espace de Von Neumann est appelé *pair* si la somme de ses coordonnées est paire, sinon il est appelé *impaire*.

et à la topologie de Moore :

Définition 2.5 (Définition topologie de Moore[11]). Soit $L = \{-1, 0, 1\}$ et $p = (p_1, \dots, p_k) \in \mathbb{Z}^k$. Le voisinage $N[c(p)]$ — ou simplement $N(p)$ — de $c(p) \in C^k$ est défini par :

$$N(p) = \begin{cases} \{c(p_1 + l_1, \dots, p_k + l_k), l_i \in L\} & \text{si } p_i \text{ est pair } \forall i = 1, \dots, k \\ \{c(p)\} & \text{si } p_i \text{ est impair pour au moins un } i. \end{cases}$$

Ainsi L.A. Ankeney et G.X Ritter introduisent-ils dans ce papier ([11]) deux idées : celle de deux types de points et de voisinages en fonction de la parité des coordonnées du point ; celle de considérer les points de l'image comme une cellule, partie de l'espace euclidien. Ces deux idées sont finalement proches de celles que l'on retrouve dans l'approche de E. Khalimsky et V.A Kovalevsky bien que théoriquement différentes : les voisinages de E. Khalimsky sont obtenus à partir d'une relation d'ordre et sont différents de ceux de Von Neumann ou Moore (voir Figures 2.7 et 2.9(b)), quant aux cellules utilisées par V.A. Kovalevsky, elles sont définies abstraitement et comme éléments d'un complexe cellulaire (voir la définition 2.15 à la section 2.4.3).

E. Khalimsky en 1987 [130] a proposé une topologie pour les espaces finis dénombrables basée sur la notion d'espace topologique connexe ordonnée [132, 131], appelé COTS (voir Définition ci-dessous).

Définition 2.6 (Définition 2.1 de [132, 131]). Un Espace Topologique Ordonné Connexe – *Connected Ordered Topological Space* – (COTS) est un espace topologique connexe X

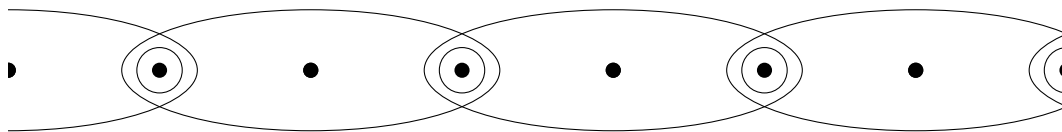


FIG. 2.8 – Exemple d'une portion de COTS montrant les voisinages minimaux de chaque point

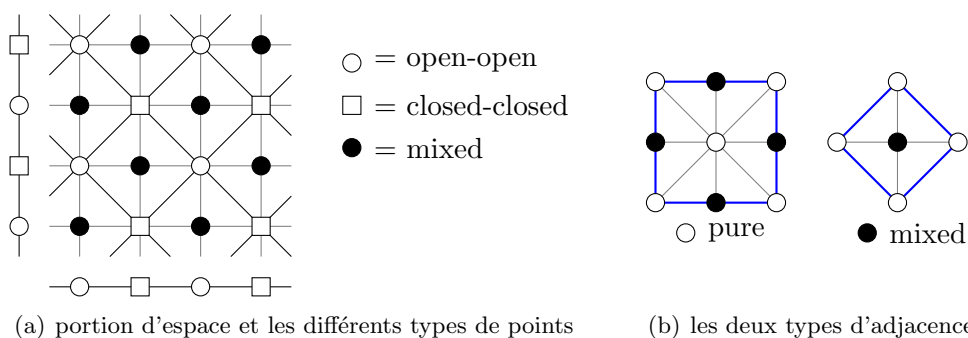


FIG. 2.9 – Topologie de l'espace "KKM" $2d$: les lignes bleues (épaisses) indiquent l'adjacence à l'intérieur du voisinage, le point central n'appartenant pas lui-même au voisinage

muni de la propriété suivante :

Si Y est un sous-ensemble de trois points, il existe un $y \in Y$ tel que Y rencontre deux composantes connexes de $X \setminus \{y\}$, i.e., quels que soient trois points, l'un des trois « sépare » les deux autres (cf la figure 2.8 pour un exemple).

Cette topologie a été présentée comme bien adaptée aux traitements informatiques de part sa nature discrète. Dans [132, 131], E. Khalimsky, R. Kopperman et P. R. Meyer développent cette idée et l'appliquent aux images numériques. Ils proposent de voir l'image comme un espace discret, produit de deux COTS. On obtient alors trois types de points (voir Figure 2.9(a)) et des voisinages selon les coordonnées du point de l'image, et donc de la position du point sur la grille (voir Figure 2.9(b)). Cette topologie présente l'originalité de ne pas être "homogène" au sens où l'espace n'est plus considéré comme un ensemble de points qui ont les mêmes propriétés. Un point de l'espace de E. Khalimsky en dimension 2 peut être ouvert, fermé ou mixte, c'est-à-dire à la fois ouvert et fermé. De plus les voisinages de ces points ne sont pas les mêmes selon qu'ils sont purs, c'est-à-dire ouvert-ouvert ou fermé-fermé, ou mixtes, c'est-à-dire ouvert-fermé ou fermé-ouvert (cf Figure 2.9(b)). Mais le principal intérêt réside dans les propriétés topologiques de cet espace. Ainsi, pour la première fois, le lien direct est fait avec des définitions classiques en topologie. En effet, non seulement il est montré dans [132] que l'espace de E. Khalimsky en dimension 2 est un espace T_0 -séparé, mais les notions d'arcs, de courbes, de courbes

de Jordan sont également définies et un théorème de Jordan est prouvé.

Un des principaux reproches fait aux approches dites “non homogènes”, et souvent un frein à leur utilisation en pratique, est que ces espaces ne correspondent pas à ce qu’affiche un écran d’ordinateur. Nous discuterons de ce point de manière plus approfondie à la section 2.5, mais il est intéressant de noter que E. Khalimsky, R. Kopperman et P. R. Meyer ont apporté un élément de réponse dans [131] dès 1990. En effet, ils s’intéressent dans cette étude aux notions d’ensembles de points connexes, de bords, de frontières et par extension à la notion de scène et d’écran. La notion de *scène*, en particulier, est intéressante (voir Définition 1 de [131]) car, bien que définie dans un souci de préciser les notions de bords et de frontières, elle correspond finalement à ce qu’en pratique on utilisera en segmentation d’image. En effet pour ce type d’application, une image est toujours vue comme un ensemble de régions partitionnant l’image en composantes connexes.

Définition 2.7 (Définition 1 de [131]). Une *scene* Π dans un plan discret – digital plane² – est une partition de ce plan, dont tous les éléments sont connexes.

La difficulté posée par leur définition très générale est justement que l’espace de E. Khalimsky est hétérogène et qu’une partition quelconque n’aura que peu de sens en pratique car ce qui nous intéresse dans ce genre d’application, c’est la partition des ensembles de pixels. Cette difficulté est levée grâce aux définitions d’*ensembles réguliers*, de *scènes robustes*, d’*écrans* et d’*affichage* :

Définition 2.8 (Définition 11 de [131] (*Regular set, Robust scene, Cartoon*)). Un ensemble A est dit *régulier* – *regular* – si $\overline{A} = \overline{\overset{\circ}{A}}$ et $\overset{\circ}{A} = \overset{\circ}{\overline{A}}$. Une *scène robuste* – *robust scene* – est une scène Π , dont chaque élément a un intérieur connexe et est contenu dans la fermeture de son intérieur. Une *scène robuste* – *robust scene* – est donc une collection d’ensembles réguliers – *regular set* –. Un *dessin* – *cartoon* – est une union finie de séparateurs fermés.

Définition 2.9 (Définition 17 de [131] (*Screen, display*)). Un *écran pur* – *pure screen* – est un sous-espace $U = \{(x, y) : x, y \text{ à la fois fermé ou ouvert}\} \subseteq X \times Y$ (chacun muni de la topologie du sous-espace). Un *écran ouvert* – *open screen* – est un sous-espace $P = \{(x, y) : x, y \text{ ouvert}\} \subseteq X \times Y$. Étant donnée une scène robuste Π , son *dessin associé* – *associated cartoon* – est $C_\Pi = X \times Y \setminus \bigcup \{\overset{\circ}{A} : A \in \Pi\}$ et son *affichage* – *display* – est $D_\Pi = \{A \cup P : A \in \Pi\}$.

Ces définitions sont intéressantes car elles montrent qu’une approche théorique de la topologie des images numériques, ainsi qu’un espace non homogène, ne sont pas un frein

²Un *digital plane* pour [131] est un espace de E. Khalimsky en dimension 2, c’est-à-dire le produit de deux COTS.

aux applications. Bien au contraire ces notions permettent de définir formellement les objets que l'on manipulera en pratique tout en offrant des garanties de robustesse quant aux résultats. La Proposition 18 de [131] vient corroborer ce fait :

Proposition 2.1 (Proposition 18 de [131]). *Robust scenes have the same associated cartoons iff they have the same displays.*

Le *cartoon* représentant en quelque sorte le dessin des frontières, cette proposition nous garantit qu'on peut retrouver les régions d'une image à partir de leurs frontières si la partition en régions est une scène robuste. Le théorème 19 de [131] nous donne même un moyen d'obtenir facilement une telle partition :

Théorème 2.2 (Théorème 19 de [131]). *Each partition of P into 4-connected sets is the display of a robust scene.*

Malheureusement, ce théorème nous conseille d'utiliser la 4-connexité pour nos applications de segmentation, alors que ce type d'espace permet beaucoup plus de finesse dans la définition des régions comme le montre V.A. Kovalevsky dans son approche (voir section 2.4). Bien évidemment, cette approche ne se limite pas au cas $2d$, et on pourra se reporter à [141] pour un survol de cette approche et l'extension des résultats précédents à la dimension 3.

Le fait de voir tous les éléments de l'espace comme des points, seulement différenciés par leur type (ouvert, fermé ou mixte), pose donc certaines difficultés en pratique, ou amène à des représentations qui ne correspondent pas à l'intuition. Ainsi M. Coupric, G. Bertrand et Y. Kenmochi [50] définissent courbes et surfaces comme suite d'éléments de l'espace "KKM" à partir d'une notion d'ordre topologique. Les objets obtenus, courbes et surfaces, sont de plus équivalents à la supercouverture (modèle de discrétisation proposé par E. Andrès, voir par exemple [31, 35]) dans l'espace de Khalimsky de leur homologue euclidien. En revanche, si l'on fait l'analogie entre l'espace de E. Khalimsky et un complexe euclidien, une courbe peut contenir des surfels et une surface des voxels (voir par exemple Figure 9 de [50]). On a donc des objets euclidiens de dimension 1 et 2 qui contiennent des cellules de dimension 2 et 3, respectivement. Si dans l'espace de E. Khalimsky cela n'apparaît pas, puisqu'il n'y a pas de notion de dimension des éléments de l'espace, leur représentation dans un espace discret usuel (une grille) fait apparaître ces incongruités. Finalement, soit on traite tous ces points de manière équivalente et on se retrouve avec des objets difficiles à traiter en pratique, soit on fait attention à leur type, comme pour la notion de *pure screen*, et on se retrouve avec des restrictions pas toujours faciles à appréhender intuitivement. C'est pour cela que j'ai préféré l'approche interpixels proposée par V.A. Kovalevsky dans [146] avec l'ajout d'une notion de dimension des éléments de l'espace.

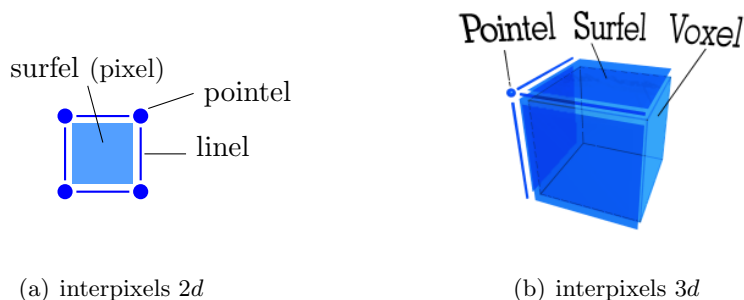


FIG. 2.10 – Éléments interpixels en dimension 2 et 3 pour un complexe cubique

M.B. Smith [214] propose une approche différente, basée sur la notion d'espace clos, c'est-à-dire d'espaces topologiques définis à base d'une relation de fermeture. D'après M.B. Smith, cette approche est plus générale. Ainsi par exemple, les relations entre la théorie des graphes et la topologie développée dans [214], notamment par la définition de *graphe topologique*, permettent de fixer les liens entre le monde continu et le monde discret en voyant les espaces "continus" comme limites d'espaces discrets (*digital spaces* en anglais pour être exact). Néanmoins, afin d'avoir une structure satisfaisante et d'aboutir à la notion d'espace quasi-discret, il munit ses espaces clos d'une relation de spécialisation, un préordre. On aboutit donc à un espace T0-séparé et à une structure topologique équivalente à celle des COTS (cf section 2.4.2), celle des complexes cellulaires (cf Section 2.4.3) ou celle de la *topologie-étoile* (cf Section 2.4.6). Cette approche a donc été de fait peu reprise. À ma connaissance, seul l'auteur lui-même, dans [215] (voir la discussion Section 2.5), et A. Galton [99], qui se base sur la définition d'espaces clos pour définir des espaces topologiques quasi-discrets lui permettant de modéliser le temps et le mouvement, ont repris ces travaux.

2.4.3 Complexe cellulaire abstrait : approche "Kovalevsky"

V.A. Kovalevsky [146] propose quant à lui de s'appuyer sur le concept de *complexe cellulaire abstrait* (voir Définition 2.15), introduisant de fait les éléments interpixels (voir Figure 2.10 pour un exemple en dimension 3) dans les images, pour définir une topologie des images numériques.

Définition 2.15 ([146] complexe cellulaire abstrait). Un complexe cellulaire abstrait $C = (E, B, \dim)$ (ou plus simplement complexe abstrait) est un ensemble E d'éléments abstraits équipé d'une relation binaire antisymétrique, anti-réflexive et transitive $B \subset E \times E$ (relation d'ordre), appelée relation *bord*, ainsi que d'une fonction dimension $\dim : E \rightarrow \mathbb{N}$, telle que $\forall (e, e') \in B, \dim(e) < \dim(e')$.

Définir ainsi son image comme un complexe abstrait permet à V.A. Kovalevsky de rester très général et de s'affranchir de la "topologie euclidienne" comme il l'affirme dans [144]. Néanmoins, les exemples qu'il donne dans [146] dans le cadre de l'application à l'analyse d'images sont naturellement des complexes cubiques avec les éléments inter-pixels de la Figure 2.10. Cette approche interpixels permet d'éviter tout naturellement tout paradoxe de connexité. En effet, la relation *bord* permet de définir simplement la notion de chemin (suite de cellules telles que quelles que soient deux cellules consécutives de la suite, l'une borde l'autre) et donc celle de connexité. La connexité se faisant donc par les cellules, deux courbes ne peuvent se croiser que si elles ont en commun une cellule et sont dans la même composante connexe. Impossible donc de reproduire le paradoxe de connexité. Ainsi dans le cas de la Figure 2.5(a) de la page 18, le sommet des pixels où se produit un croisement devra être soit bleu (couleur foncée) et donc déconnectera la courbe orange (chemin clair), soit orange et donc déconnectera la courbe bleue. De même dans le cas de la Figure 2.5(b) de la page 18, si la courbe bleue est connexe, l'intérieur ne peut être connexe à l'extérieur, les sommets des pixels de la courbe étant considérés comme bleus. Dans le cadre des complexes cellulaires, il est donc facile de démontrer un théorème de Jordan comme le fait V.A. Kovalevsky dans [146]. De la même manière, la notion topologique de frontière devient cohérente et il existe bien une seule frontière commune à deux sous-ensembles incidents :

Définition 2.10 (Définition 7 de [146] (*Boundary*)). Le *bord* – *boundary* – d'un sous-ensemble $S \subset C$ relativement à C est le sous-ensemble $Fr(S, C)$ de tous les éléments e de C tel que tout voisinage ouvert de e contienne à la fois des éléments de S et de son complémentaire $C \setminus S$.

Les complexes cellulaires étant équivalents à des espaces T0-séparés de Alexandroff, il existe un plus petit voisinage de e dans C , noté $St(e, C)$ et appelé *étoile ouverte* de e , et on remplacera avantageusement dans la définition précédente, "any open neighborhood" par "an open star $St(e, C)$ ".

Un autre résultat important de [146] est qu'il montre que les topologies basées sur les complexes cellulaires sont les seules valables pour les ensembles discrets finis (cf théorème ci-dessous) :

Théorème 2.3 (Théorème de V.A. Kovalevsky [146]). *Tout espace topologique fini muni de la propriété de séparation est isomorphe à un complexe cellulaire abstrait au sens de la définition 2.15.*

Ceci n'est pas en contradiction avec l'affirmation de Khalimsky [130] pour qui la notion d'espace topologique connexe ordonné est la seule cohérente pour les ensembles discrets finis. En effet, non seulement les espaces $2d$ définis par E. Khalimsky [130] et V.A. Kovalevsky [146] sont homéomorphes (cf ci-dessous, Theorem 3 de [134]), mais de

plus, comme l'a montré G. T. Herman [118], un complexe cellulaire peut être vu comme un espace topologique connexe ordonné muni d'une fonction dimension.

Théorème 2.4 (Théorème 3 de [134]). *Les espaces de Khalimsky et Kovalevsky sont homéomorphes.*

Néanmoins, G.T. Herman tempère l'affirmation de V.A. Kovalevsky en montrant que la notion de dimension rajoutée à l'ensemble ordonné, pour prouver l'équivalence entre un complexe cellulaire et un espace topologique T0-séparé, ne correspond pas à la dimension naturelle des cellules du complexe. Ainsi les pointels, lignels et surfels (pixels) auraient alors respectivement comme dimension 0, 6 et 8^3 ! Donc si le [théorème de V.A. Kovalevsky](#) est bien vérifié, il n'y a pas stricte équivalence avec les complexes cellulaires que l'on choisirait naturellement. La notion de dimension naturelle attachée aux cellules d'un complexe cellulaire est donc plus contraignante que celle qu'on pourrait attacher à un espace topologique T0-séparé. Donc, contrairement à ce qu'affirme V.A. Kovalevsky, G.T. Herman pense que la question, de savoir si « un complexe cellulaire avec une fonction dimension affectant aux cellules leur “dimension géométrique” est un cadre formel suffisamment large pour couvrir tous les problèmes potentiels de l'analyse d'image » reste donc ouverte.

Si l'approche pixel, puis l'approche *KKM* ont été privilégiées, on note un regain d'intérêt pour l'approche cellulaire notamment, car dans le cas de la dimension 3, la modélisation des images en est simplifiée puisqu'elle peut se déduire naturellement du cadre formel des complexes cellulaires. Ainsi si l'on examine les travaux de P.K. Saha, D. Dutta [205], et de P.K. Saha, A. Rosenfeld [206, 207], ils s'intéressent aux propriétés topologiques d'espaces $3d$ représentés par des tétraèdres, des polyèdres et pour finir des voxels (au sens général « élément volumique de l'image $3d$ » ou tout simplement cellule de l'image) car ces objets géométriques sont classiquement utilisés pour modéliser les images $3d$. Afin de pouvoir étudier certaines propriétés topologiques d'ensembles de telles cellules, et notamment celles liées à la connexité telles que les notions de tunnels, cavités ou de points simples, ils introduisent la propriété de *forte normalité* pour de tels ensembles :

Définition 2.16 (*Normalité et forte normalité ([207])*). Soit $\mathcal{P}$ un ensemble de tuiles (sous-ensembles fermés, bornés et convexes de $\mathbb{R}^3$) ; $\mathcal{P}$ sera dit *normal* (ou *localement fini*) si, $\forall P \in \mathcal{P}$, le nombre de tuiles qui intersectent P est fini.

$\mathcal{P}$ sera dit *fortement normal* (*SN*) si $\forall P \in \mathcal{P}$, $\forall P_1, \dots, P_{n,n \geq 1} \in \mathcal{P}$ qui intersectent P , et si $I = P_1 \cap \dots \cap P_n \neq \emptyset$, alors $I \cap P \neq \emptyset$.

³La fonction dimension rajoutée est : $dim(e) = \max_{e' \in E} |St(e')| - |St(e)|$, donc pour un pointel $dim(e) = 0 = 8 - 8$, pour un lignel $dim(e) = 6 = 8 - 2$, et pour un surfel $dim(e) = 8 = 8 - 0$.

Cette propriété permet de garantir certaines propriétés topologiques sur les voisinages. Ainsi, par exemple, il est montré que :

Théorème 2.5 (Théorème 1 de [207]). *Si $\mathcal{P}$ est fortement normal – strongly normal –, alors pour tout $\mathcal{P}' \subseteq \mathcal{P}$, le voisinage $N_{\mathcal{P}'}$ de tout $p \in \mathcal{P}'$ est simplement connexe.*

Leur motivation à l'origine de ces travaux était de pouvoir répondre aux questions suivantes :

- Est-ce que la suppression d'une tuile préserve la topologie de la représentation dans le voisinage de la tuile supprimée ?
- Si la suppression change la topologie, est-il possible de mesurer ces changements ?

Il est intéressant de noter que si l'on adjoint à leur propriété de forte normalité la propriété naturelle pour des grilles (et qui était indiquée à l'origine de ces travaux dans [206]) qui est que l'intersection de deux tuiles soit une face, une arête ou un sommet commun aux deux tuiles, on retrouve alors la structure d'un complexe polyédral. Une bonne représentation à base topologique telles que celles basées sur les cartes combinatoires (voir section 3.2 ou par exemple [210]) peut alors permettre de répondre à ces questions. Cela ne vient que renforcer l'idée selon laquelle, un complexe cellulaire associé à une représentation à base topologique est un bon modèle pour les images numériques.

Néanmoins, ce type d'approche purement cellulaire pose le problème pratique du rapport avec les pixels, ou voxels, lorsque l'on veut faire du traitement d'images. C'est la raison pour laquelle E. Khalinsky, R. Kopperman et P.R. Meyer, dans [131], ont défini la notion de *pure screen* et *open screen*. C'est cette même motivation, celle de considérer une image comme un espace hétérogène mais en définissant les principales propriétés (connexité, bord, ...) à partir des points représentés à l'écran qui nous a amenés à proposer la *star-topologie* ([1, 81]). Dans la section suivante nous développons l'approche par complexe polyédral et notamment celle de la *star-topologie*.

2.4.4 Complexe polyédral : star-topologie et fonction d'illumination

Alors que plusieurs auteurs ([157, 159, 158, 230, 232]) continuaient de développer l'approche pixel de la topologie numérique (voir section 2.3), j'ai entrepris dans ma thèse de définir une topologie numérique, basée sur la notion d'interpixels et de complexe cellulaire, adaptée aux images niveaux de gris ou couleurs, et à la notion de région telle qu'elle est utilisée en analyse d'images et en particulier en segmentation d'images. Cette dernière préoccupation était déjà présente dans les travaux de V.A Kovalevsky [146] et apparaît aussi dans les travaux de Y. Wang et P. Bhattacharya [232]. Ces derniers étendent la notion d'*image bien composée* proposée par L. Latecki (voir Définition 1 présentée Section 2.3) à celle de *région bien composée* pour définir des régions "topologiquement correctes" avec une frontière extérieure assimilable à une courbe de Jordan.

Bien que considérant la frontière comme formée des éléments de bord des pixels de la région, et donc commune à deux régions adjacentes, ils n'utilisent pas explicitement la notion d'interpixels et surtout ne s'en servent pas pour définir la connexité d'une région. En effet, ils imposent à leurs régions d'être *bien composées* dans le sens suivant :

Définition 2.11 (Définition 3.1 de [232]). Dans une grille quelconque (régulière ou non), un ensemble de pixels S , d'une image en niveaux de gris, est bien composé si pour tout point du bord $\mathbf{v}$, $S \cap \mathcal{V}(\mathbf{v}) \neq \emptyset$ implique que $S \cap \mathcal{V}(\mathbf{v})$ est connexe par arête au point $\mathbf{v}$.

Si leur définition n'est pas aussi stricte que celle de la 4-connexité car s'appliquant à des grilles quelconques, de fait, dans le cadre d'une grille régulière, cela revient à obliger les régions à être 4-connexes. En effet, si deux pixels du bord d'une région sont strictement 8-adjacents (voir Figure 2.3(b) à la page 16)) le voisinage de la région autour de ce point ne peut être connecté par arête. S'ils avaient considéré explicitement l'interpixels et imposé l'appartenance d'un tel sommet à une et une seule région, alors une région pourrait avoir des points de 8-adjacence stricte tout en interdisant à une autre région d'être strictement 8-connexe en ce même point. Néanmoins la définition des points de bords d'une région comme les points de $\mathbb{R}^2$ bords des pixels, les références à l'espace euclidien sous-jacent et à la topologie classique, rejoignent l'idée, développée dans ma thèse, de s'appuyer sur la notion d'espace quotient. Cette dernière solution permet d'être plus précise dans la formalisation des liens entre l'espace discret et la topologie classique euclidienne.

En effet, contrairement à V.A. Kovalevsky dans [144], nous pensons qu'il est préférable d'aborder les complexes cellulaires comme le quotient d'un espace euclidien dans le cadre des images. V.A. Kovalevsky affirme dans [144] que l'intérêt de considérer des complexes cellulaires abstraits est de pouvoir avoir une théorie indépendante de la topologie euclidienne. Je pense au contraire que dans la mesure où l'on s'intéresse aux images, pas simplement en tant que structure, mais en tant que données d'une application, que ce soit de l'analyse, de la reconnaissance, ou même de la modélisation, on cherche toujours à représenter (modéliser) le monde réel, et il est donc préférable de se placer dans un cadre théorique nous permettant de nous rapprocher des espaces euclidiens. Sinon, pourquoi chercher à définir pour l'espace des images numériques les mêmes propriétés que les objets euclidiens ? Cet avis est d'ailleurs partagé par d'autres auteurs comme [11, 141] ou [13, 14, 72]. Ces derniers se basent également sur les notions de complexe polyédral et/ou d'espace quotient pour modéliser l'espace image.

Je présenterai donc dans un premier temps notre proposition, la star-topologie [1, 82, 81], puis celle de [13, 14, 72] qui ont également proposé de se baser sur un espace quotient et un complexe polyédral mais en s'appuyant sur une architecture multi-niveaux pouvant embrasser différents niveaux de représentations : euclidien, logique, conceptuel et simplicial.

2.4.5 Complexe convexe (ou polyédral)

Nous proposons donc de suivre la voie ouverte par V.A. Kovalevsky [146], mais en faisant le choix de complexes cellulaires convexes⁴ (appelés généralement *complexes polyédraux* en géométrie discrète). Nous avons été les premiers à faire ce choix des complexes convexes pour la représentation des images. La justification de ce choix est donnée par le fait que le bord d'une cellule est constitué par un ensemble de faces de cette cellule, ce qui est bien le cas pour les structures classiques de la géométrie discrète ou des pavages issus de la numérisation.

Notre démarche peut être appréhendée à partir de deux approches complémentaires :

- la donnée d'une relation d'ordre caractérisant une topologie sur un ensemble fini ;
- le passage au quotient de la topologie classique.

Concernant la première approche, on rappelle qu'on ne peut trouver de topologie séparée au sens de Hausdorff sur un ensemble fini (c'est-à-dire où 2 points distincts admettent des voisinages disjoints), mais simplement T0-séparée ce qui signifie : étant donné deux points distincts, il existe un voisinage de l'un qui ne contient pas l'autre. Le théorème suivant est celui permettant de faire le lien entre les espaces T0-séparés et une relation d'ordre :

Théorème 2.6 (Théorème de Mac Kinsey, Tarski). *Il y a une bijection naturelle entre les espaces topologiques T0-séparés finis et les ensembles ordonnés finis.*

On peut alors définir une relation d'ordre $\leq_B$ de la manière suivante : si E est un espace topologique T0-séparé, on définit la relation $\leq_B$ par $x \leq_B y$ si et seulement si tout voisinage de x contient y . Réciproquement si E est un ensemble ordonné, la base d'ouverts définissant la topologie est déterminée ainsi : pour chaque élément x de E , on définit un ouvert de la base par l'ensemble des successeurs de x . L'ensemble de tous ces ouverts ainsi définis forme cette base. On peut associer une notion de dimension de façon classique aux relations d'ordre (elle est égale à 0 pour les éléments minimaux, 1 pour les minimaux de l'espace obtenu après avoir enlevé les minimaux, etc). Cette relation est appelée la relation de bornage. C'est à partir de ce constat que V.A. Kovalevsky a proposé d'utiliser les complexes cellulaires abstraits.

La seconde approche consiste à chercher une structure discrète qui récupère le plus possible les propriétés de la géométrie et de la topologie classiques. L'idée est de décomposer un sous-espace de $\mathbb{R}^n$ en objets bien définis et dénombrables afin d'obtenir un ensemble discret quotient de $\mathbb{R}^n$. Cet ensemble discret correspond alors à une partition

⁴La littérature en géométrie discrète emploie le terme *complexe polyédral* que j'ai donc employé jusque ici. Néanmoins le terme de *complexe convexe* utilisé par S. Lefschetz [161] me semble plus approprié, notamment car les polytopes (et non des polyèdres qui restreindraient l'étude au cas 3d) considérés sont convexes. C'est pourquoi j'emploie de préférence le terme *complexe convexe* en ce qui concerne la star-topologie.

d'un sous-ensemble de $\mathbb{R}^n$, il suffit donc de lui associer la topologie quotient. Nous avons choisi de décomposer $\mathbb{R}^n$ en ensembles polytopes convexes. Ces polytopes forment les cellules de notre complexe cellulaire.

Définition 2.17 (Complexe convexe [161] d'après [113]). On appelle *complexe convexe* une famille finie $\mathcal{C}$ de polytopes ouverts P_i de $\mathbb{R}^n$ appelés cellules du complexe tels que :

- Si $P_i \in \mathcal{C}$, alors toute face de $P_i \in \mathcal{C}$.
- Si $P_i \in \mathcal{C}$ et $P_j \in \mathcal{C}, i \neq j$, alors $\overline{P_i} \cap \overline{P_j}$ est vide ou bien est une face commune de P_i et P_j .

La dimension du complexe est par définition le maximum des dimensions des cellules du complexe.

Définition 2.18 (Sous-complexe [161] d'après [113]). On appelle sous-complexe $\mathcal{S}$ d'un complexe $\mathcal{C}$ un ensemble de cellules de $\mathcal{C}$ ayant une structure de complexe ce qui équivaut à un ensemble de cellules telles que si c est une cellule de $\mathcal{S}$, toute face de c est une cellule de $\mathcal{S}$.

Définition 2.19 (Complexe engendré [161] d'après [113]). On appelle *complexe engendré* par un ensemble $\mathcal{K}$ de cellules d'un complexe donné, le plus petit sous-complexe, noté $\mathcal{C}(\mathcal{K})$, contenant ces cellules.

Définition 2.20 (Polyèdre d'un complexe [161] d'après [113]). Si E est une famille de cellules, on appelle *polyèdre* de E , et on note $|E|$ l'union des points des cellules de E .

2.4.6 Star-topologie

Un complexe convexe $\mathcal{C}$ est donc un ensemble fini. Une T0-topologie sur $\mathcal{C}$ est associée à une relation d'ordre (voir [Théorème de Mac Kinsey](#), [Tartski](#) présenté Section 2.4.5). Afin d'illustrer la démarche que nous avons choisie, je donne ci-dessous les principales définitions, propriétés de [1, 81] qui nous ont permis de démontrer le théorème 2.5 décrivant le passage à la topologie quotient de celle induite par $\mathbb{R}^n$, théorème fondamental de la *star-topologie*. Les preuves des propositions ou théorèmes donnés ici pourront être lues dans [1, 81].

Définition 2.21 (Relation de bornage et incidence). Étant donnés deux polytopes P_1 et P_2 , on dit que P_1 borne P_2 et on note $P_1 \leq_B P_2$ si P_1 est une face de P_2 ou si $P_1 = P_2$. Si P_1 borne P_2 ou P_2 borne P_1 , on dit que P_1 et P_2 sont *incidents*.

Proposition 2.2. La relation de bornage $\leq_B$ définie sur l'ensemble des polytopes d'un complexe convexe est une relation d'ordre.

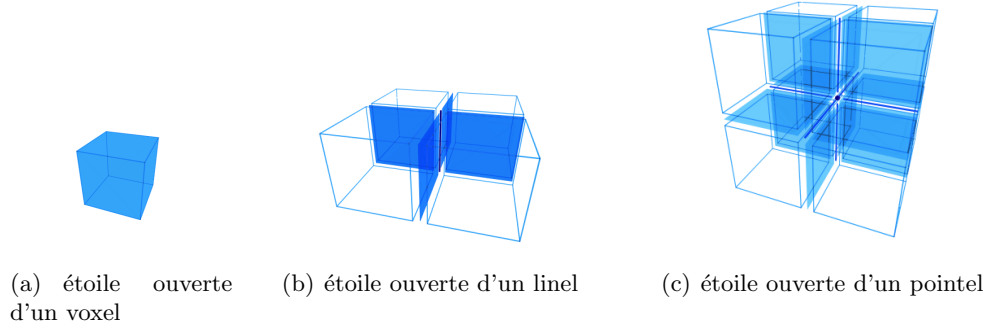


FIG. 2.11 – Étoiles ouvertes en dimension 3 pour un complexe cubique

Définition 2.22 (Étoile ouverte). On appelle *étoile ouverte* (respectivement *complexe étoile*) d'une cellule c dans le complexe $\mathcal{C}$, l'ensemble des cellules de $\mathcal{C}$ bornées par c (respectivement le plus petit complexe contenant l'étoile ouverte de c). On note $St(c) = \{\gamma \in \mathcal{C}, c \leq_B \gamma\}$, l'étoile ouverte de c (voir exemples en 3d à la figure 2.11).

Nous devons maintenant caractériser les ouverts du polyèdre $|\mathcal{C}|$ qui est la réunion des cellules de $\mathcal{C}$. Ceci nous permettra de définir sur $\mathcal{C}$ une topologie quotient.

Définition 2.23 (Relation *cell* et application *Cell*). On définit sur $|\mathcal{C}|$ une relation d'équivalence *cell* : x et y sont *cell-équivalents* s'ils appartiennent à la même cellule ouverte. L'espace quotient de $|\mathcal{C}|$ par cette relation est l'espace $\mathcal{C}$ des cellules. On dit qu'un ensemble $E \in |\mathcal{C}|$ est saturé pour *cell* s'il est une réunion de cellules de $\mathcal{C}$. On note *Cell* l'application qui à un point de $|\mathcal{C}|$ associe sa classe d'équivalence, c'est-à-dire la cellule à laquelle il appartient.

Proposition 2.3. Soit $\mathcal{C}$ un complexe convexe :

- (i) si $c \in \mathcal{C}$ et si un ouvert saturé du polyèdre $|\mathcal{C}|$ contient $|c|$, il contient $|St(c)|$;
- (ii) un sous-ensemble saturé de $\mathcal{C}$ est fermé ssi c'est un sous-complexe de $\mathcal{C}$;
- (iii) $|St(c)|$ est un ouvert.

Proposition 2.4. Pour la topologie de $|\mathcal{C}|$, si c est une cellule de $\mathcal{C}$ alors $|St(c)|$ est le plus petit polyèdre ouvert contenant c .

Nous pouvons maintenant définir la *topologie-étoile* sur l'espace des complexes convexes comme la topologie quotient de la topologie classique sur $\mathbb{R}^n$:

Définition 2.24 (Topologie-étoile de $\mathcal{C}$). Si $|\mathcal{C}|$ est muni de sa topologie classique (induite par celle de $\mathbb{R}^n$), la topologie quotient définie sur $\mathcal{C}$ a pour ouverts, par définition

de la topologie quotient, les images par Cell (voir Définition 2.23) des ouverts de $|\mathcal{C}|$ qui sont saturés pour la relation cell ⁵. Cette topologie est appelée la *topologie-étoile*.

Toujours afin d'illustrer l'intérêt d'une approche quotient, nous allons maintenant démontrer un théorème de Jordan pour l'espace des complexes convexes muni de la topologie-étoile. Pour cela on montrera d'abord le théorème fondamental suivant caractérisant la topologie-étoile :

Théorème 2.5 (Théorème fondamental de la topologie-étoile). *Sur l'espace $\mathcal{C}$, les éléments sont les cellules du complexe $\mathcal{C}$. La topologie-étoile sur $\mathcal{C}$ est la topologie engendrée sur $\mathcal{C}$ par les étoiles ouvertes de ses cellules.*

Démonstration. La topologie engendrée par les étoiles ouvertes des cellules a pour ouverts les unions des étoiles ouvertes de cellules (dans la définition générale, on prend les unions quelconques et intersections finies, mais ici, les intersections d'étoiles ouvertes de cellules sont des étoiles ouvertes de cellules ou bien sont vides). D'après les deux propositions 2.3 et 2.4, les ouverts saturés de $|\mathcal{C}|$ sont les unions d'étoiles ouvertes, donc les topologies sont confondues. $\square$

Ensuite nous caractérisons la notion de courbe dans les espaces de complexes convexes $2d$:

Définition 2.25 ($\mathcal{C}$ -courbe). Une $\mathcal{C}$ -courbe est une suite finie $c_0, \dots, c_{2n}$ de cellules d'un complexe convexe $\mathcal{C}$ de dimension 2 qui sont alternativement des pointels (cellules de dimension 0) et des linels (cellules de dimension 1), telles que c_0 et c_{2n} soient des sommets et que les cellules d'indices successifs soient incidentes. Elle est dite simple si toutes ses cellules sont distinctes. Elle est dite fermée si $c_0 = c_{2n}$.

On montre alors facilement (voir [1, 81]) la proposition suivante :

Proposition 2.6. *Une $\mathcal{C}$ -courbe (resp. fermée, simple) est l'image par Cell d'une courbe continue (resp. fermée, simple) de $\mathbb{R}^2$.*

Il nous faut maintenant définir la notion classique de connexité et montrer son lien avec la connexité sur $\mathbb{R}^n$:

Définition 2.26 (Ensemble connexe). Un ensemble de cellules d'un complexe est dit *connexe* si quelles que soient les cellules c et c' , il existe une suite $c = c_0, \dots, c_p = c'$ telle que deux cellules d'indices consécutifs soient incidentes.

Théorème 2.7. *Soit E un ensemble de cellules d'un complexe dans $\mathbb{R}^n$. Alors $|E|$ est connexe par arcs si et seulement si son image par Cell est connexe.*

⁵Les ouverts de $|\mathcal{C}|$ qui sont saturés pour la relation cell sont des unions de cellules ; les ouverts dans l'espace quotient ainsi obtenus sont donc bien des ensembles de cellules.

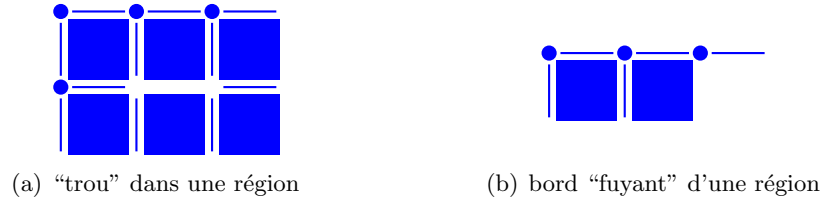


FIG. 2.12 – Exemples de problèmes posés par les cellules de dimension inférieure à n dans des régions connexes au sens de 2.26

On peut désormais s’attaquer au théorème de Jordan et le prouver très simplement :

Théorème 2.8 (Théorème de Jordan discret sur les complexes convexes). *Soit $\mathcal{C}$ un complexe convexe de dimension 2 et soit Γ une $\mathcal{C}$ -courbe fermée simple. Γ sépare le plan en deux composantes connexes distinctes dont une bornée est appelée l’intérieur.*

Démonstration. Une $\mathcal{C}$ -courbe fermée simple est l’image par $Cell$ d’une courbe fermée simple Γ' dans $\mathbb{R}^2$. On applique le théorème de Jordan classique à Γ' . L’intérieur et l’extérieur de la courbe réelle sont des unions de cellules, on peut donc appliquer le théorème 2.7 ce qui prouve le théorème. $\square$

La simplicité de cette démonstration est à mettre en parallèle avec celles n’utilisant pas la notion de topologie quotient (voir par exemple [131, 232]) et démontre à elle seule l’intérêt d’une topologie numérique qui soit une topologie quotient de la topologie classique sur les espaces euclidiens.

L’autre objectif que nous nous étions fixé était d’avoir une topologie adaptée à l’analyse d’images. Comme nous l’avons mentionné plus haut, afin de pouvoir être facilement utilisée dans les applications, il faut des définitions de région, d’adjacence, ou d’autres notions, qui sont basées sur les cellules de plus grande dimension (pixel en $2d$ et voxel en $3d$) afin de rester cohérent avec les données de l’application et de pallier ainsi l’inconvénient, souvent opposé, de l’utilisation de cellules “virtuelles” pour garantir la cohérence topologique de l’image. Or la définition de connexité d’un sous-ensemble de cellules (voir Définition 2.26) que nous avons donnée n’est pas satisfaisante en ce sens, voir Figure 2.12 pour des exemples de problèmes posés. Une région ne pourra donc pas être simplement un sous-ensemble de cellules connexe. Il faut lui adjoindre des propriétés supplémentaires nous assurant leur cohérence et une définition utilisable en pratique, c’est-à-dire telle que les cellules de dimensions inférieures (linel et pointel en dimension 2) servent à l’adjacence entre régions, à la définition des frontières ou des bords, mais n’aient pas à être considérées explicitement à l’intérieur des régions.

Dans un premier temps, nous définirons, dans le cadre des complexes convexes, les notions classiques en topologie numérique de κ -adjacence, κ -chemin et κ -connexité :

Définition 2.27 (κ -adjacence, κ -chemin, κ -connexité). Deux m -cellules e_1, e_2 de $E \subseteq \mathcal{C}$, $m \leq n$, sont κ -adjacentes, $\kappa < m$, si et seulement s'il existe une κ' -cellule $e \in E$ telle que $m \geq \kappa' \geq \kappa$ et $e_1, e_2 \in St(e)$.

Un κ, m -chemin dans E de e à e' est une suite $e = e_0, \dots, e_p = e'$ de m -cellules de E , telles que $\forall i = 1 \dots p$, e_i est κ -adjacente à e_{i-1} . Lorsqu'il n'y a pas d'ambiguïté sur la dimension m des cellules considérées, on parlera simplement de κ -chemin.

E , sous-ensemble de dimension m de $\mathcal{C}$ est κ -connexe si et seulement si pour tout couple de cellules (e, e') de dimension m de E , il existe un κ -chemin dans E de e à e' .

Remarque 2.1. Ces définitions, dans le cadre des complexes convexes, sont bien équivalentes aux définitions classiques de connexité et de chemin. Ainsi, par exemple en dimension 2, la 0-connexité correspond bien à la 8-connexité et la 1-connexité à la 4-connexité. D'autre part, elles restent conformes à la définition 2.26 de connexité générale et nous permettent donc toujours de nous placer dans le cadre de la topologie-étoile.

Pour traduire en terme de complexe convexe la notion de région, traditionnellement définie comme un ensemble connexe de pixels ou voxels (c'est-à-dire de cellules de dimension n dans un espace nd), il suffit d'ajouter à la propriété de connexité des propriétés assurant que :

- (i) il ne peut y avoir de "trous" de dimension non maximale (inférieure à n) dans la région ;
- (ii) nos régions étant définies à partir des cellules de dimension n , toute cellule de dimension inférieure à n , qui appartient à la région, doit borner une cellule de dimension n de la région.

La première propriété se traduit topologiquement par $\forall : x \in \mathcal{R}, (St(x) \setminus \{x\}) \subseteq R \Rightarrow x \in R$ et la seconde par toute m -cellule, $m < n$, de R borne une n -cellule de R . Les propositions 2.9 et 2.10 nous permettent alors de donner la définition 2.28 d'une κ -région :

Proposition 2.9. Soit R un ensemble de cellules κ -connexes de $\mathcal{C}$, les deux conditions suivantes sont équivalentes :

- (1) $\overset{\circ}{R} \subseteq R$
- (2) $\forall x \in \mathcal{C}, (St(x) \setminus \{x\}) \subseteq R \Rightarrow x \in R$

Proposition 2.10. Soit R un ensemble de cellules κ -connexes de $\mathcal{C}$, les deux conditions suivantes sont équivalentes :

- (1) $R \subseteq \overset{\circ}{R}$
- (2) toute m -cellule, $m < n$, de R borne une n -cellule de R .

Définition 2.28 (κ -région). R ensemble de dimension n de cellules de $\mathcal{C}$, est une κ -région si et seulement si R est κ -connexe et $\overset{\circ}{R} \subseteq R \subseteq \overset{\circ}{R}$. On appellera *région* une 0-région.



FIG. 2.13 – ‘Région 2d’ en 3d qui n’est pas intuitivement une surface : $\overset{\circ}{\overline{R}}$ ne contient que les surfels, en effet l’étoile ouverte des linels n’est pas incluse dans l’ensemble et donc eux-mêmes ne sont pas inclus dans l’ouvert de $\overline{R}$, cette condition ne nous permet pas de “boucher le trou”. Donc, bien qu’il manque intuitivement un linel, cette ensemble de cellules serait une κ -region sans la condition sur la dimension n . Le terme de région est donc plus approprié que le terme de surface.

Remarque 2.2. La condition sur la dimension n de R est indispensable et cette définition de région ne permet donc pas de définir des surfaces ou des ensembles bien formés de dimension $n - 1$ (voir Figure 2.13 pour un exemple de “région” de dimension $n - 1$ possédant un “trou”). C’est pourquoi nous n’avons pas repris le terme de k -hypersurface que nous avons employé dans [81].

Si la définition d’une région impose de rajouter la condition $\overset{\circ}{\overline{R}} \subseteq R \subseteq \overline{\overset{\circ}{R}}$ à la connexité, en revanche, les notions d’*extérieur*, de *bord* ou de *frontières* de la topologie classique restent les mêmes pour l’espace des complexes convexes munis de la topologie-étoile :

Extérieur : $Ext(R) = (\mathcal{C} \setminus \overset{\circ}{R})$

Bord : $\mathcal{B}d(R) = R \setminus \overset{\circ}{R}$

Frontière : $\partial R = \mathcal{B}d(R) \cup \mathcal{B}d(\mathcal{C} \setminus R) = Ext(R) \setminus \overset{\circ}{R}$

De la même façon, la notion d’adjacence entre régions se définit comme pour les topologies classiques :

Définition 2.29 (Adjacence entre régions). Deux régions R_1, R_2 de $\mathcal{C}$ sont adjacentes si et seulement si $\partial R_1 \cap \partial R_2 \neq \emptyset$

En revanche, autorisant la 0-connexité pour les régions, cette définition de l’adjacence pourrait nous amener à des aberrations dans les traitements en analyse d’images. Ainsi, par exemple, sur la Figure 2.14, les régions A et C sont adjacentes et leur fusion serait donc possible pour un processus de segmentation en régions. Mais la région AC ainsi créée traverserait la région B ! Pour éviter cela, il faut donc tout naturellement tenir

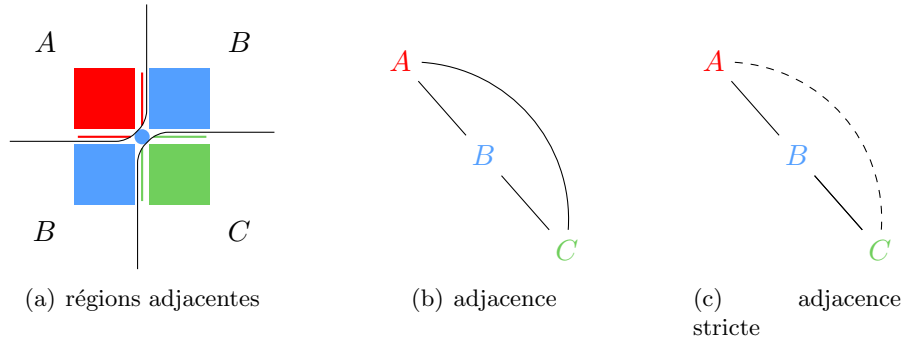


FIG. 2.14 – Adjacence et adjacence stricte

compte des éléments interpixels de bords, comme l'a déjà montré V.A. Kovalevsky [146]. Nous définissons donc la notion de *stricte adjacence* et c'est elle qui doit être utilisée en analyse d'images.

Définition 2.30 (Stricte adjacence). Deux régions R_1, R_2 de $\mathcal{C}$ sont *strictement adjacentes* si et seulement si elles sont adjacentes et si $\mathcal{Bd}(R_1) \cap \partial R_2 \neq \emptyset$ ou $\mathcal{Bd}(R_2) \cap \partial R_1 \neq \emptyset$.

Remarque 2.3. Cette condition de stricte adjacence impose tout simplement que les cellules de la frontière, communes à deux régions adjacentes, appartiennent au bord de l'une ou de l'autre. Ainsi dans le cas de la Figure 2.14, le pointel du milieu, seule cellule de la frontière commune à A et C n'appartient ni au bord de l'une, ni au bord de l'autre ; ces deux régions ne sont donc pas strictement adjacentes et pour cause puisque la région C les sépare. Les courbes dessinées donnent une interprétation continue de la frontière entre ces régions. On peut voir que dans cette interprétation continue, A et C ne sont pas adjacentes. C'est ce que traduit la stricte adjacence dans le cas discret.

La topologie-étoile sur les complexes convexes reprend donc l'idée d'interpixels suggérée par V.A. Kovalevsky [146] mais s'appuie fortement sur la notion d'espace quotient pour faciliter la démonstration de certains théorèmes. De plus, les définitions de région et d'adjacences strictes basées sur des propriétés topologiques (intérieur, ouvert, fermeture, frontières et bords) nous permettent de garantir de bonnes propriétés à nos ensembles de pixels ou voxels et donc garantissent des traitements topologiquement cohérents, notamment en segmentation en régions. Cette approche interpixels a été utilisée dans des algorithmes liés à la segmentation d'images, que ce soit en $2d$ (voir par exemple [26]) ou $3d$ (voir par exemple [57]). A. Galton [99] propose même une approche topologique du mouvement dans les espaces discrets en se basant principalement sur les travaux de M.B. Smyth [214] sur les espaces topologiques clos mais en citant nos travaux [1] et ceux de [130, 146] comme exemple d'espaces topologiques dédiés aux images numériques et basés sur la notion d'ouverts. P. Desbarats et S. Gueorguieva [68], quant à eux, re-

visitent ces “topologies interpixels” pour proposer un cadre général de représentation topologique des objets discrets. Néanmoins, si cette approche interpixels apporte certains avantages indéniables du point de vue topologique, certains préfèrent s’appuyer sur des structures pixels plus simples selon eux à manipuler. Ainsi J. Marchadier, D. Arqués et S. Michelin [168] indiquent que les topologies telles que la star-topologie [1, 81] ou celle de V.A. Kovalevsky [146] ont l’inconvénient de l’espace mémoire (codage de toutes les cellules) ou celui de rendre difficile certaines opérations, comme reconstruire la partition à partir des bords. Pour l’espace mémoire, si ces espaces, comme celui de Khalimsky [132] nécessitent de stocker plus d’information, des structures de données moins coûteuses en mémoire, et permettant d’implanter des opérations de base comme le remplissage ou même des opérateurs topologiques, ont été proposées (voir par exemple [82, 24, 147, 22, 145, 155]). J. Lamy [156] a même proposé une implantation C++ d’une librairie de traitement d’images intégrant des éléments de topologie numérique et notamment l’interpixels.

Si cette topologie (celle de V.A. Kovalevsky ou la nôtre) a bien été utilisée dans le cadre du traitement d’images, c’est surtout dans le cadre de la modélisation des bords des régions. L’intégration des éléments interpixels dans les processus de décision, notamment dans le critère de fusion de régions en segmentation d’images, est rarement fait car dans ces processus l’élément de référence reste le pixel ou le voxel. Or si les définitions que nous avons proposées apportent une solution à ce problème, elles restent difficiles à appréhender en pratique car elles sont globales alors que la décision se fait généralement localement. À ce sujet le cadre de travail proposé par R. Ayala, E.Dominguez, A.R. Frances et A. Quintero [14, 15], et que nous présentons ci-dessous section 2.4.7, semble être une solution intéressante.

2.4.7 Fonction d’illumination

R. Ayala, E.Dominguez et A.R. Frances [73] proposent une architecture multi-niveaux permettant d’établir un lien entre le monde discret des images numériques et le monde continu d’un espace euclidien. Comme proposé initialement par V.A. Kovalevsky [146], puis par nous [1, 81], l’espace discret est représenté par un complexe polyédrique dans lequel les cellules de dimension maximale représentent les pixels de l’image et les cellules de dimension inférieure servent à définir les adjacences. La différence fondamentale entre l’approche proposée par R. Ayala, E.Dominguez et A.R. Frances et celle de V.A. Kovalevsky ou la nôtre est que l’adjacence est définie globalement par des fonctions particulières, appelées *fonction d’éclairage* [14, 15], qui s’appuient sur des propriétés locales des objets considérés. Donc contrairement à nous, deux configurations identiques de pixels produiront toujours la même adjacence entre les cellules de cet objet et donc le même analogue continu. Dans notre modèle, l’adjacence n’est pas définie par un modèle de

configuration locale des pixels, mais est particulière à chaque configuration, l'idée étant que l'adjacence n'est pas décidée globalement mais localement par un processus d'analyse pouvant prendre en compte des informations de plus haut niveau. Ainsi, avec notre approche, deux régions différentes avec la même configuration de pixels que l'objet de la Figure 2.15(a) pourront donner l'une l'analogue continu 2.15(e), et l'autre l'analogue continu 2.15(i), alors que dans l'approche de R. Ayala, E.Dominguez et A.R. Frances, c'est la fonction d'éclairage, définie pour toute l'image, qui déterminera quel est l'analogue continu de cette configuration de pixels ; deux régions différentes de la même image correspondront donc au même analogue continu. Si cela peut apparaître comme plus cohérent, il n'en est rien dans le cadre de l'analyse d'images, en particulier pour les images médicales, où la résolution n'est pas toujours suffisante pour garantir que deux configurations identiques de pixels dans la même image correspondent forcément au même analogue continu. Nous pensons donc que c'est à l'analyse de déterminer quelle est la forme de l'objet, mais en respectant un modèle garantissant une topologie cohérente des objets ainsi définis.

Néanmoins, cette approche présente un intérêt certain, notamment dans la manière de définir les différents niveaux et dans la méthode pour passer du niveau discret au niveau continu. De plus, si l'on considère que la définition des objets ne se fait qu'à partir des cellules de dimension maximale sans se préoccuper de l'adjacence entre les pixels, comme le font en pratique la grande majorité des algorithmes de segmentation, alors cette approche est parfaitement adaptée et fournit un cadre de travail riche et bien formalisé.

L'espace discret est construit sur le premier niveau de cette architecture et sa définition conditionne l'efficacité de tout le modèle. Ce premier niveau est, en effet, un complexe particulier, dont les cellules de dimension maximale sont utilisées pour représenter les *spels*⁶ de l'image. Les cellules de dimensions inférieures servent à décrire les liens entre les spels. Définir un espace discret sur ce complexe consiste à ajouter une information au complexe pour expliciter les relations d'adjacence entre les cellules maximales. Cette information est donnée par une fonction d'éclairage (voir définition 2.12 ci-dessous). Les autres niveaux du modèle construisent progressivement un analogue continu de l'image considérée en se fondant sur les relations d'adjacence choisies. Les cinq niveaux du modèles sont les suivants :

niveau physique (*device level*) est un complexe polyédrique de dimension n homogène⁷ et localement fini ;

niveau logique (*logical level*) est un graphe non orienté représentant les adjacences entre

⁶On appelle spel les points de l'image en dimension n : un pixel est un spel en dimension 2 et un voxel est un spel en dimension 3.

⁷Un complexe homogène est un complexe dont chaque cellule est la face d'au moins une cellule de dimension maximale.

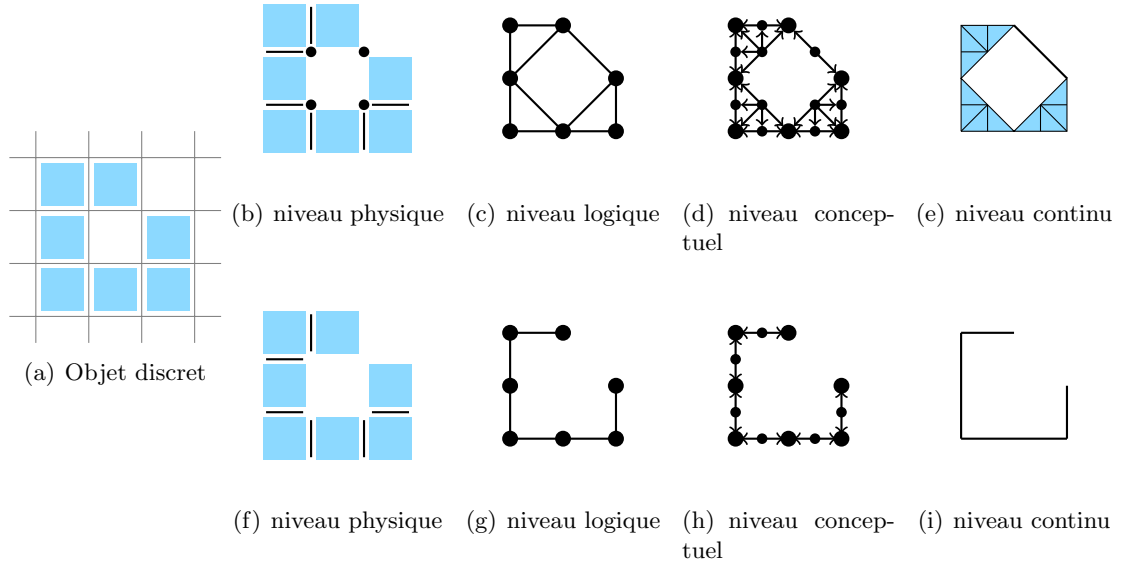


FIG. 2.15 – Architecture multi-niveaux et fonction d'éclairage

cellules de dimension maximale ;

niveau conceptuel (*conceptual level*) est un graphe orienté représentant les relations d'incidence entre les cellules du complexe impliquées dans les relations d'adjacence ;

analogue simplicial (*simplicial analogue*) est un complexe simplicial construit sur les chemins maximaux du graphe du niveau précédent ;

analogue continu (*continuous analogue*) est l'analogue continu du complexe simplicial : son polyèdre sous-jacent, sous-ensemble de l'espace euclidien.

L'originalité de cette approche est double. D'abord par la définition d'une architecture multi-niveaux, premièrement introduite dans [73], mais surtout par l'ajout au modèle d'une *fonction d'éclairage*, introduite dans [14] et développée dans [15] et [16], qui permet d'éviter de restreindre leur modèle aux topologies classiques définies par l'adjacence entre pixels.

Avant de pouvoir définir les fonctions d'illumination, il est nécessaire de préciser certaines notations. On notera K_n les cellules de dimension n du complexe K . Soit un objet $O \subseteq K_n$, on appelle *étoile de α dans O* l'ensemble $st_n(\alpha, O) = \{\sigma \in O, \sigma \in St(\alpha)\}$ et *étoile étendue de α dans O* l'ensemble $st_n^*(\alpha, O) = \{\sigma \in O, \sigma \cap \bar{\alpha} \neq \emptyset\}$ ⁸. Enfin on note

⁸La définition que nous donnons ici est différente de celle d'origine publiée dans [14, 15] où $st_n^*(\alpha, O) = \{\sigma \in O, \alpha \cap \sigma \neq \emptyset\}$. En effet, de la même manière que pour la note de bas de page suivante, cette définition ne correspond pas aux exemples donnés par les auteurs. En particulier, selon cette définition, sur la figure 2.16(c), le pixel le plus à droite ne devrait pas appartenir à $st_n^*(\alpha, O)$.

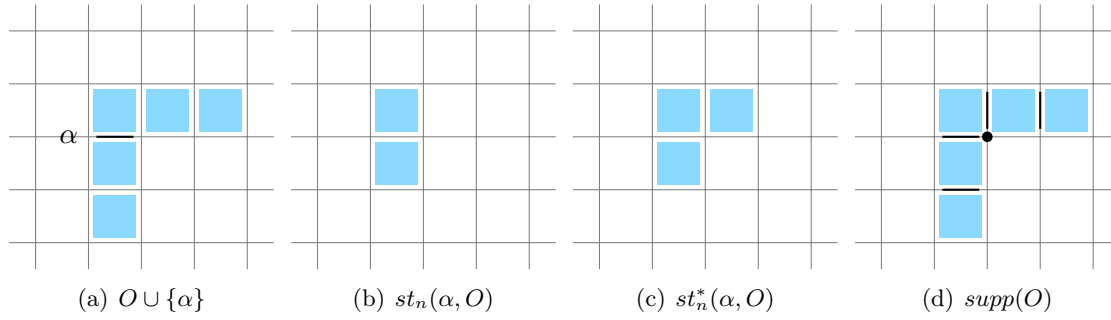


FIG. 2.16 – Exemples illustrant les notations de base nécessaires à la définition des fonctions d'éclairage.

$supp(O) = \{\alpha \in K, \bar{\alpha} = \cap \{\bar{\sigma}, \sigma \in st_n(\alpha, O)\}\}$ ⁹, le support de O , c'est-à-dire l'ensemble des cellules de O et des cellules de dimension inférieures nécessaires à l'adjacence deux à deux des cellules de O .

Définition 2.12 (Définition 3 de [16]). Soit un complexe polyédrique K représentant le modèle physique de l'image, une *fonction d'éclairage faible* (w.l.f. pour « weak lightning function ») sur K est une application $f : \mathcal{P}(K_n) \times K \rightarrow \{0, 1\}$ où $\mathcal{P}(K_n)$ est l'ensemble des parties de K_n et satisfaisant les cinq axiomes suivants pour tout objet $O \in \mathcal{P}(K_n)$ et toute cellule $\alpha \in K$:

1. si $\alpha \in O$ alors $f(O, \alpha) = 1$ (*axiome de l'objet*) ;
2. si $\alpha \notin supp(O)$ alors $f(O, \alpha) = 0$ (*axiome du support*) ;
3. $f(O, \alpha) \leq f(K_n, \alpha)$ (*axiome de faible monotonie*) ;
4. $f(O, \alpha) = f(st_n^*(\alpha, O), \alpha)$ (*axiome de faible localité*) ;
5. si $O' \subseteq O \subseteq K_n$ et $\alpha \in K$ sont tels que $st_n(\alpha, O) = st_n(\alpha, O')$, $f(O', \alpha) = 0$ et $f(O, \alpha) = 1$ (*axiome de connexité du complémentaire*) alors :
 - (a) l'ensemble de cellules $\alpha(O, O') = \{\omega < \alpha, f(O', \omega) = 0, f(O, \omega) = 1\}$ n'est pas vide ;
 - (b) l'ensemble $\bigcup \left\{ \overset{\circ}{\omega}, \omega \in \alpha(O, O') \right\}$ est connecté dans $\partial\alpha$;
 - (c) si $O \subseteq \bar{O} \subseteq K_n$ alors $f(\bar{O}, \omega) = 1, \forall \omega \in \alpha(O, O')$.

Si $f(O, \alpha) = 1$, on dit que f éclaire la cellule α pour l'objet O .

⁹Nous reprenons ici la définition proposée par S. Alayrangues et J.-O. Lachaud dans [5] qui est différente de la définition d'origine donnée dans [14, 15] où $supp(O) = \{\alpha \in K, \alpha = \cap \{\sigma \in st_n(\alpha, O)\}\}$. En effet cette dernière n'est pas exacte au sens où elle ne correspond pas à tous les exemples donnés par les auteurs, en particulier celui de la figure 2.16(d) où, selon cette définition, les cellules α doivent être égales à l'intersection de cellules σ de l'objet, or celles-ci sont des ouverts et leur intersection est donc vide !

Une w.l.f. est dite *fortement locale* à une cellule $\alpha \in K$ si $f(O, \alpha) = f(st_n(\alpha, O), \alpha)$ pour tout objet $O \in K_n$ et f est appelée une *fonction d'éclairage fortement locale* si elle est fortement locale à chaque cellule $\alpha \in K$. La propriété de forte localité implique les axiomes de faible localité et de connexité du complémentaire (axiomes 4 et 5 de la définition 2.12). Il suffira donc pour définir une fonction d'éclairage fortement locale de remplacer ces deux axiomes par l'*axiome de forte localité* : $f(O, \alpha) = f(st_n(\alpha, O), \alpha)$.

Ces différents axiomes permettent de garantir les propriétés recherchées et notamment le fait que l'analogue continu reste une interprétation de l'objet discret O , représenté visuellement uniquement par les pixels, ou plus précisément par les n -cellules, c'est-à-dire les α -terminaux. Ainsi l'axiome 1 garantit que les n -cellules sont toujours représentées dans l'analogue continu ; l'axiome 2 garantit que seules les cellules du support, c'est-à-dire l' α -noyau de $|\beta(O)|$, mais pas nécessairement toutes, peuvent apparaître dans l'analogue continu, garantissant ainsi que toute cellule de dimension inférieure à n est face de deux cellules de dimension n de l'objet O ; de la même manière, si une cellule semble connecter deux pixels dans l'objet O , alors l'axiome 3 garantit qu'elle est présente dans l'analogue continu ; l'axiome 4, quant à lui, nous assure que notre perception de l'analogue continu reste locale, et que la présence ou non des cellules dans l'objet est décidée uniquement en fonction du voisinage local ; enfin, l'axiome 5 garantit que l'analogue continu, ainsi défini, a une interprétation cohérente en regard de la connexité du complémentaire.

2.5 Synthèse et perspectives

L'approche « E. Khalimsky, R. Kopperman, P.R. Meyer » (KKM) reste la plus étudiée et la plus citée. De mon point de vue c'est principalement dû au fait que les éléments de l'espace restent des points discrets, l'espace est donc homogène est plus facile à représenter, plus proche de sa représentation informatique immédiate par une matrice de points. De plus leur approche est dans la suite logique de l'idée de deux relations d'adjacence, introduite dans le cadre des images binaires par A. Rosenfeld [197], en la formalisant dans un cadre général. Néanmoins, l'approche par les complexes cellulaires me semble plus riche et tend d'ailleurs à se généraliser, notamment dans le cadre des images $3d$. En effet elle permet d'avoir une modélisation proche des outils utilisés en imagerie numérique $3d$, alors que l'approche KKM oblige à utiliser uniquement des voxels. De plus, si l'espace proposé par E. Khalimsky, R. Kopperman et P.R. Meyer est homogène dans sa forme, les espaces topologiques basés sur les complexes cellulaires apportent une notion de dimension des éléments de l'espace permettant une approche plus intuitive et plus proche des notions continues de surfaces et de bords. Cette conviction déjà évoquée dans ma thèse [81] a été confirmée par la suite par T.Y. Kong [140] et M. Cou-

prie, G. Bertrand et Y. Kenmochi [50] qui ont montré que, si les espaces topologiques de E. Khalimsky ont les bonnes propriétés pour les images numériques, la notion de dimension apportée par les complexes cellulaires justifie les différentes notions d'adjacence dans ces espaces. M. Couprie et G. Bertrand, associés à X. Daragon ont continué à explorer cette voie en associant espaces topologiques basés sur une relation d'ordre partiel à la *KKM* et complexes cellulaires. Dans [61] ils montrent des propriétés sur les notions de bords, de frontières et de surfaces, très intéressantes. Un fait important notamment, est la récursivité de la définition de surface donnée par [50], originellement de [77] dans le cadre des graphes, car comme indiqué dans [61], cette définition peut facilement s'entendre en dimension n . Néanmoins pour avoir une notion de bord symétrique qui est essentielle, ils proposent dans [60] une notion de *voisinage dérivé* en subdivisant encore l'espace, ce qui complique donc la modélisation. De plus, la notion de rang associé aux cellules, et utilisé pour définir leur ordre partiel, ne correspond pas à la dimension naturelle des cellules, et les objets définis sont hétérogènes au sens de la dimension naturelle des cellules (une courbe n'est pas uniquement un ensemble de cellules de dimension 0 et 1).

Une telle approche, basée sur les complexes cellulaires et une notion de dimension des cellules, implique de dissocier topologie, géométrie et affichage. Cela est souvent présenté comme un inconvénient majeur, les cellules de dimension inférieure à n n'étant que virtuelles par rapport aux données initiales. À mon avis, c'est plutôt une bonne chose et un avantage, car l'affichage n'est qu'une représentation graphique des objets modélisés, en aucun cas il en est la modélisation. De plus, un même objet peut être affiché de multiples façons; le choix de la représentation doit être laissé à l'outil de visualisation et non pas imposé par le modèle. V.A. Kovalevsky dans [148] dit d'ailleurs

The objection because of visibility is not pertinent since the visibility has nothing to do with topology.

et il cite l'exemple des images 3d pour lesquelles nous raisonnons avec des voxels alors que les outils de visualisation ne les affichent jamais en tant que tels. Ce sont leurs faces ou même des triangles qui sont affichés.

Néanmoins, l'imagerie numérique reste une représentation du monde réel et, que ce soit pour l'analyse d'images ou pour la synthèse d'images, un lien entre les mondes réel et virtuel doit être établi. Il l'est souvent par la notion de discrétisation. Il existe plusieurs définitions de discrétisation, toutes tentant de se rapprocher de ce qu'un capteur numérique fait. Certains modèles proposés en géométrie discrète s'appuient fortement sur cette notion, tel que le modèle de *supercouverture* proposé par É. Andres [8]. E.H. Kronheimer a développé dans [149] une approche des espaces numériques discrets par la topologie, en associant à chaque élément discret un sous-ensemble ouvert correspondant à la réponse du capteur à un élément de l'image. Il intègre donc directement cette notion de dis-

crétisation dans son modèle d'image numérique et comme base de la topologie de cette espace. Ce faisant, il définit clairement l'espace topologique numérique comme un espace quotient de l'espace euclidien. Finalement un espace numérique discret est pour lui la partition d'un espace euclidien en sous-ensembles ouverts, la partition canonique étant celle en carré unité. On voit donc, dès 1992, que des auteurs approchent la topologie numérique par la notion de quotient.

Je pense donc que l'approche quotient associée à celle d'interpixels et de complexe cellulaire permet de définir correctement une topologie adaptée aux espaces des images numériques, tout en permettant de modéliser les différents problèmes liés à l'analyse d'images. Il est d'ailleurs intéressant de noter que de nombreuses propositions allant dans ce sens ont été faites, toutes assez proches les unes des autres, sans pour autant qu'une synthèse puisse se dégager et être un modèle adopté majoritairement. On trouve ainsi les notions d'*ensemble régulier* (la fermeture d'un ensemble régulier est la fermeture de son intérieur et l'intérieur d'un ensemble régulier est l'intérieur de sa fermeture), de *scènes robustes*, d'*écrans* et d'*affichage* (voir la définition 2.8 page 25) proposées dans [131, 149] qui sont à rapprocher des notions de κ -régions (cf la définition 2.28 page 37), de *stricte adjacence* (cf la définition 2.30 page 39) de la *topologie-étoile* que j'ai proposées dans ma thèse [1, 81], sans pourtant être tout à fait équivalentes. On notera bien sûr les travaux de V.A. Kovalevsky sur les *complexes cellulaires abstraits* [146, 144, 147, 145] et bien sûr le *framework multi-niveaux*, du continu au discret, proposé par E. Dominguez, E. and Frances et A. Marquez [73] associé aux *fonctions d'éclairage* proposées avec A. Quintero [13, 14, 15]. S. Alayrangues et J.-O. Lachaud dans [5] et [4] ont étudié les liens entre relation d'ordre et complexe cellulaire en proposant les notions de *complexe fortement normal*, à rapprocher de la notion de *forte normalité* [206], de *support*, à rapprocher des ensembles réguliers et des κ -régions, de *fortes fonctions d'éclairage faible* (strong weak lightning function), extension des fonctions d'éclairage de [13] mais rajoutant des propriétés proches de la stricte adjacence et des κ -régions que j'avais définies [1, 81].

Sans doute faudrait-il approfondir tous ces travaux et proposer une topologie, basée sur les notions d'espace quotient et de complexe cellulaire, dont la relation d'ordre de spécialisation serait associée à la dimension, et donnant les outils pour faire le lien avec les représentations continues des objets discrets. Toutes les notions évoquées ici, telles que voisinage, région, adjacence, ensemble régulier, forte normalité, affichage, écran, dessin, seraient à préciser. Enfin, un lien avec la géométrie discrète doit être établi, peut-être en adaptant les définitions, désormais classiques, de la géométrie discrète arithmétique à l'interpixels et à l'hétérogénéité d'un espace cellulaire.

Chapitre 3

Analyse combinatoire des images

Préambule

« Analyse combinatoire de l'imagerie » est le sujet initial de mes travaux et en reste le fil conducteur. Mon sujet initial était donc de trouver une modélisation à base de graphes des images, plus performante que le graphe d'adjacence des régions, et d'utiliser au mieux l'algorithmique de graphe afin d'améliorer la performance des algorithmes de segmentation. Je me suis donc intéressé aux cartes combinatoires, présentées à la section 3.2, qui elles-mêmes m'ont mené aux modèles topologiques des images numériques (voir chapitre 2). Le thème de la modélisation à base topologique m'a amené à collaborer avec P. Lienhardt et Y. Bertrand à Strasbourg avec qui j'ai ensuite co-encadré mon premier thésard Guillaume Damiaud. J'ai alors noué des contacts avec le laboratoire SIC de Poitiers et le LaBRI de Bordeaux. Après mes premiers résultats liés aux modèles de représentation à base topologique, je suis revenu sur le sujet initial et en collaboration avec Jens Gustedt, que j'ai rejoint en Post-Doc à Berlin, actuellement Directeur de Recherche INRIA au LORIA à Nancy, nous avons développé des algorithmes de segmentation basés sur l'Union-Find (voir la section 3.1.1). Ces algorithmes ont ensuite donné lieu à plusieurs développements par d'autres auteurs et m'ont permis d'établir plusieurs collaborations. Notre principale publication sur le sujet ([84]) reste encore très citée.

3.1 Approche algorithmique de la segmentation

Dans un processus complet de traitement de l'information visuelle, plusieurs étapes sont nécessaires pour arriver à analyser l'image qualitativement et quantitativement. Plusieurs méthodologies ont été définies. La première, sans doute celle de référence, est celle proposée par D. Marr [169] qui décompose l'analyse d'images en trois grandes étapes :

la segmentation qui consiste à extraire de l'image des éléments caractéristiques, souvent sans connaissance *a priori* ;

la reconstruction qui a pour but de calculer les propriétés (barycentre, surface, orientation, distribution des niveaux de gris, ...) des éléments identifiés à l'étape précédente, et d'en extraire les relations spatiales et topologiques les reliant ;

la reconnaissance qui cherche à retrouver l'organisation perceptuelle de ces entités et à les mettre en correspondance avec le modèle recherché, la description symbolique de l'objet.

D'autres méthodologies ont ensuite été proposées, détaillant davantage ces différentes étapes, et notamment l'étape de segmentation qui nous intéresse ici. On citera en particulier R.M. Haralick and L.G. Shapiro [116] qui décompose le processus complet en

cinq étapes, les trois premières correspondant à la segmentation. Ces trois étapes du processus de segmentation sont les suivantes :

le conditionnement qui consiste à supprimer le bruit introduit par le capteur ; il fait appel à des techniques de filtrage, en utilisant éventuellement la morphologie mathématique ;

l'étiquetage qui associe à chaque pixel une étiquette précisant quel type d'information il véhicule (processus de seuillage, de détection d'arêtes, de coins, d'appartenance à une forme primitive, ...) ;

le groupement qui identifie des ensembles maximaux de pixels connectés ayant la même étiquette.

On voit à travers cette dernière description que l'approche proposée est orientée traitement du signal, chaque point est considéré comme un échantillon devant être filtré puis analysé. En fait, avant d'aborder le problème de l'analyse d'images, il est primordial de définir le cadre formel des concepts que l'on étudie, d'exprimer ces concepts sous forme mathématique, et enfin de proposer un ensemble de solutions permettant à la fois de vérifier la validité de la théorie et d'apporter des solutions concrètes au problème posé. C'est ici que nous avons divergé par rapport aux approches classiques de l'analyse d'images. En effet, à part les travaux en morphologie mathématique, pratiquement tous les travaux en analyse d'image proposaient de considérer l'image comme un signal échantillonné. Pour ma part, je me place après le conditionnement et je traite donc l'image comme un espace discret. L'étape de segmentation en régions consiste alors à partitionner l'image en sous-espaces connexes maximaux pour un critère d'homogénéité donné. Bien sûr, cette approche n'était pas complètement nouvelle, plusieurs auteurs avaient déjà proposé des algorithmes de segmentation (voir par exemple [189, 122, 201, 240] pour des résultats bien antérieurs à ma thèse), et même des structures combinatoires comme le graphe d'adjacence des régions (RAG en abrégé) [203, 12] pour représenter l'information. Néanmoins ces résultats souffraient de ne pas s'appuyer sur un modèle formel adapté. Ainsi les résultats en segmentation des régions étaient principalement des méthodes décrivant le processus de segmentation et non des solutions algorithmiques garantissant le résultat et proposant une mesure de l'efficacité du processus. De même, comme il a été démontré par plusieurs auteurs, en particulier V.A. Kovalevsky [146], le graphe d'adjacence des régions n'est pas une structure satisfaisante permettant de représenter correctement les relations topologiques entre les régions. Ma proposition était donc de proposer un modèle combinatoire cohérent des images, d'où les études menées sur les espaces topologiques (voir chapitre 2), sur lequel m'appuyer pour proposer des méthodes combinatoires d'analyse d'images.

Je me suis donc d'abord intéressé au problème algorithmique posé par la segmentation d'image. Avec Jens Gustedt nous avons proposé une solution basée sur l'Union-Find

(voir la section 3.1.1 et l'annexe A). J'ai ensuite cherché à proposer un nouveau critère d'homogénéité performant et tenant compte des propriétés algorithmiques permettant de calculer celui-ci. Ces travaux ont été menés en collaboration avec Richard Nock (voir la section 3.1.2 et l'annexe B). Enfin je me suis attaché à proposer une structure combinatoire permettant de modéliser les images tout en respectant le cadre formel proposé, c'est-à-dire permettant de coder la topologie des régions définies par notre processus de segmentation. Une partie de ces travaux a été réalisée pendant ma thèse, l'autre partie l'a été en collaboration avec Guillaume Damiani que je co-encadrais alors (voir la sections 3.2 et l'annexe C)

3.1.1 Union-Find

La segmentation d'images a été, et est toujours, un domaine de recherche très étudié et de nombreuses méthodes de segmentation ont été proposées dans la littérature (voir par exemple les survols [240, 115, 188, 179]). La plupart des méthodes de segmentation sont basées sur les deux premières propriétés relatives à l'information niveau de gris (ou couleur) portée par un pixel et ceux de son voisinage : *similarité* et *discontinuité*. La seconde propriété est directement exploitée par les méthodes de traitement du signal qui cherchent des fortes variations du signal image. Elles correspondent à ce qu'on appelle la détection de contours. Celles s'appuyant sur la similarité des niveaux de gris correspondent aux méthodes de segmentation en régions. Ce sont ces dernières qui nous intéressent ici car elles sont généralement algorithmiques¹. On ne s'attardera pas dans cette section sur la problématique du critère évaluant cette similarité, on considérera qu'on dispose d'un prédicat d'homogénéité permettant de répondre à la question : cet ensemble de pixels est-il homogène, i.e. les pixels le composant sont-ils similaires ? Pour une discussion sur les problèmes liés à la détermination du critère d'homogénéité, on se référera à la section 3.1.2. Une segmentation en régions est communément définie (voir par exemple [178, 189, 122, 240]) comme :

Définition 3.1 (segmentation en régions). Une segmentation en régions d'une image I munie d'un prédicat d'homogénéité $\mathcal{P}$ est une partition $R = \{R_1, R_2, \dots, R_n\}$ de I en n

¹On trouvera également des méthodes de classification partitionnant l'image en ensembles de pixels. Mais ces méthodes ne se préoccupent pas de la connexité des régions, on ne les classera donc pas dans les méthodes de segmentation en régions.

régions telle que :

$$I = \bigcup_{i \in \{1, \dots, n\}} R_i \quad (3.1)$$

$$\forall i \in \{1, \dots, n\}, R_i \text{ est connexe} \quad (3.2)$$

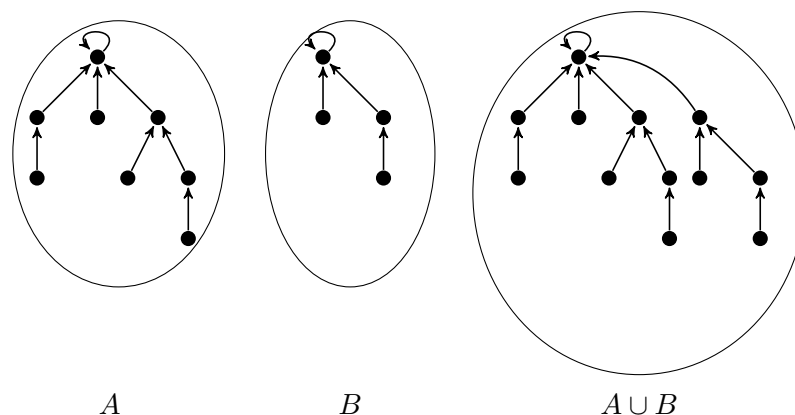
$$\forall i \in \{1, \dots, n\}, \mathcal{P}(R_i) = \text{vrai} \quad (3.3)$$

$$\forall i, j \in \{1, \dots, n\}, i \neq j, \mathcal{P}(R_i \cup R_j) = \text{faux} \quad \forall R_i, R_j \text{ adjacentes} \quad (3.4)$$

Bien sûr, avec de telles conditions, une segmentation ne sera jamais unique. On peut essayer de réduire cette indétermination en transformant la segmentation en un problème d'optimisation d'une mesure $\mathcal{Q}$ de la qualité de la segmentation. Outre qu'une telle mesure est difficilement définissable, O. Monga [175] a démontré que ce problème d'optimisation était NP-difficile. En fait, pour ces deux raisons, les algorithmes de segmentation se contentent de regrouper les pixels en régions satisfaisant les quatre critères énoncés par les équations (3.1), (3.2), (3.3) et (3.4). Cette définition d'une segmentation en régions amène naturellement à comparer ce problème à celui de l'Union d'Ensembles Disjoint (Union-Find en abrégé). Ce problème ne possède pas dans le cas général de solution en temps linéaire. La meilleure complexité connue a été obtenue pour la première fois par R.E. Tarjan [218]. Elle est en $O(m \cdot \alpha(m, n))$ où $\alpha(m, n)$ est l'inverse de la fonction d'Ackerman² et n et m ($n < m$) sont respectivement le nombre d'appels aux fonctions *Union* et *Chercher*. Cette fonction croît très lentement et on considère donc la solution comme quasi-linéaire. Il a de plus été prouvé (voir [219, 220]) qu'une telle complexité était optimale pour les opérations sur des structures de données satisfaisant certaines conditions techniques. H.N. Gabow et R.E. Tarjan [97] ont également présenté un algorithme linéaire suivant certaines restrictions. Cette question algorithmique a beaucoup été étudiée et le lecteur pourra se référer par exemple à [172, 98, 49] pour de plus amples informations, et par exemple à [221] pour des résultats récents.

Le problème de l'*Union-Find* consiste essentiellement à regrouper n éléments distincts dans une collection d'ensembles disjoints. Les deux opérations utilisées sont l'*Union* qui réalise l'union de deux ensembles et *Find* qui recherche l'ensemble auquel appartient un élément. L'analogie avec notre problème de départ devient évidente : les éléments à regrouper sont les pixels et les ensembles disjoints sont les régions. À notre connaissance, ceci n'avait été observé avant nous que par M.B. Dillencourt, H. Samet et M. Tamminen [71] qui avaient utilisé l'Union-Find pour étiqueter les régions d'images binaires. On peut donc considérer que nous avons été les premiers à proposer d'utiliser l'Union-Find dans le cadre d'images multi-niveaux (initialement en niveaux de gris) pour réaliser une

²L'inverse de la fonction d'Ackerman $\alpha(n, m)$ est définie pour $m, n \geq 1$ par $\alpha(m, n) = \min \{i \geq 1 \mid A(i, 4 \lfloor \frac{m}{n} \rfloor) \geq \log n\}$.

FIG. 3.1 – Opération *Union*.

segmentation en régions. Dans [84], J. Gustedt et moi-même avons proposé deux algorithmes de segmentation en régions, linéaires en le nombre de pixels, basés sur l'Union-Find. Nous avons poursuivi l'étude théorique d'un point de vue complexité mémoire afin de garantir une utilisation optimale de celle-ci et d'améliorer les performances pratiques des algorithmes basés sur l'Union-Find (voir [85]), ce qui nous a permis de proposer une version de nos algorithmes pour la segmentation d'images 3d (voir [86]).

Pour les preuves de complexité et les détails de ces deux algorithmes, l'auteur pourra se référer à l'annexe A qui présente notre publication [84]. Nous présenterons ici la modélisation utilisée dans le cadre des images afin de pouvoir définir ces algorithmes linéaires.

Rappelons dans un premier temps les éléments fondamentaux de la solution de R.E. Tarjan [219] pour obtenir la complexité optimale dans le cas général en $O(m.\alpha(m, n))$. Les ensembles sont représentés par une structure de données d'arbres « inversés » dans lesquels la racine joue le rôle d'élément représentatif de l'ensemble. Par « inversé », on entend qu'au lieu de lier les éléments de la racine vers les feuilles (un père connaît ses fils dans l'arbre), on lie les éléments des feuilles vers la racine (un fils connaît son père et non l'inverse), comme montré sur les exemples des figures 3.1 et 3.2. L'opération *Union* se réalise alors en temps constant en liant une racine d'un des deux arbres représentant un des deux ensembles dont on doit faire l'union, à l'autre racine de l'arbre représentant l'autre ensemble. Pour obtenir la bonne complexité, il suffit de réaliser une *Union par rang* qui consiste à lier l'arbre le moins haut sous l'arbre le plus haut (voir la figure 3.1 pour un exemple). L'idée de cette heuristique est qu'ainsi la hauteur de l'arbre final n'est pas plus grande que celle du plus haut des deux arbres ainsi liés. On n'augmente donc pas le coût de l'opération *Find*. L'opération *Find*, quant à elle, consiste à remonter récursivement jusqu'à la racine afin de trouver l'élément représentatif de l'ensemble (voir la figure 3.2 pour un exemple). L'heuristique à appliquer afin d'obtenir la complexité en

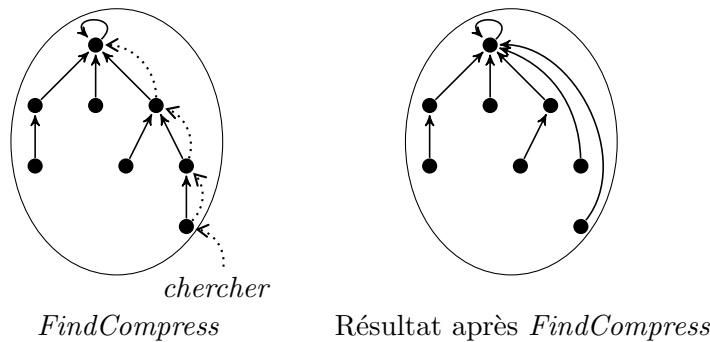
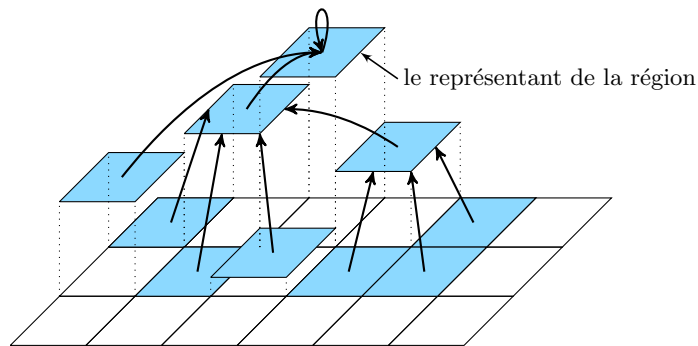


FIG. 3.2 – Compression de chemin

FIG. 3.3 – Une région dans le modèle *Union-Find*

$O(m.\alpha(m,n))$ consiste à remplacer l'opération *Find* par l'opération *FindCompress* qui une fois la racine trouvée en profite pour lier tous les éléments sur le chemin de la racine sous la racine elle-même, réduisant ainsi le coût d'une future opération *Find* sur ces éléments (voir l'exemple de la figure 3.2).

Pour utiliser l'Union-Find pour la segmentation d'images, il faut donc adopter cette représentation des régions par des arbres « inversés ». Un pixel d'une image ne sera plus seulement une valeur de couleur, mais sera représenté par une structure contenant les informations de chrominance, la référence à son parent dans l'arbre représentant la région à laquelle il appartient, ainsi que les informations de la région elle-même lorsque le pixel est le représentant de son ensemble, c'est-à-dire de sa région (voir l'exemple de la figure 3.3).

Du côté des opérations *Union* et *Chercher*, le seul changement intervient dans l'opération *Union*. En plus de l'*Union par rang*, il faut aussi garantir la connexité de la région, propriété propre aux régions qui n'est pas prise en compte pour le problème général de l'union d'ensembles disjoints. Il ne faut donc autoriser l'opération *Union* que pour des régions adjacentes. Ceci sera simplement réalisé en partant des pixels et en ne proposant

la fusion de régions que pour deux pixels adjacents : si ils sont adjacents, leurs régions le sont forcément.

L'autre précaution à prendre concerne le prédicat d'homogénéité. Il faut absolument que celui-ci possède la propriété 3.1 suivante :

Propriétés 3.1. (Efficacité du prédicat d'homogénéité)

- (i) interrogation en temps constant ;
- (ii) incrémentalité de la mise à jour.

En effet, l'union de deux régions ne sera décidée qu'après interrogation du prédicat d'homogénéité. Il faut donc garantir qu'il se calcule en temps constant si l'on veut garder la complexité linéaire de nos algorithmes *Scanline* et *MergeSquare* (voir l'annexe A), ou même la complexité en $O(m.\alpha(m,n))$ pour tout autre algorithme générique basé sur l'Union-Find. Il faut en plus garantir l'incrémentalité de la mise à jour des propriétés nécessaires au calcul de ce prédicat. Par incrémentalité, on entend que ces propriétés se mettent à jour en temps constant lors de l'union de deux régions. Ainsi, pour prendre un exemple simple, si le prédicat d'homogénéité consiste à comparer les moyennes de niveaux de gris des régions et répondre oui uniquement si la différence des moyennes est inférieure à un seuil, on peut garantir la propriété 3.1 simplement : il suffit de garder pour chaque région le nombre de pixels la composant et la somme des niveaux de gris de ses pixels. Calculer le prédicat revient à calculer la moyenne à partir de la somme des niveaux de gris et du nombre de pixels, opération triviale en temps constant : mettre à jour la somme des niveaux de gris et le nombre de pixels lors de l'union de deux régions revient à faire la somme des propriétés des régions fusionnées, cette mise à jour est donc bien incrémentale. Une fois ces garanties données, union de régions adjacentes et propriété 3.1 du prédicat d'homogénéité, on est sûr d'obtenir une segmentation efficace linéaire ou quasi-linéaire selon l'algorithme choisi.

Ces changements ne sont pas si anodins qu'ils paraissent. En effet, d'un point de vue image, deux changements fondamentaux interviennent :

Propriétés 3.2. (Segmentation Union-Find)

- (i) algorithmes de segmentation en régions simples, génériques et efficaces ;
- (ii) approche région guidée par les pixels permettant une approche par collaboration régions-contours.

Le premier point est majeur. En effet, si l'on regarde les solutions de segmentation existantes (cf les survols [240, 115, 188]) avant notre proposition [84], non seulement aucune n'avait d'étude de complexité, mais surtout elles consistaient essentiellement en des méthodes de calcul pour décider de la partition en régions, et non pas d'algorithmes réalisant cette partition. Il n'y avait donc pas d'analyse de complexité mais simplement

une présentation des résultats obtenus et une évocation des temps de calcul sans que l'on connaisse l'algorithme utilisé. D'ailleurs si l'on regarde d'autres survols pour des résultats plus récents [179, 238], ou même des articles récents comme [192, 124], on se rend compte que c'est encore le plus souvent le cas, même si la préoccupation de complexité intervient de plus en plus dans les publications en analyse d'images, comme par exemple le récent [216]. De plus la contrainte de complexité imposée sur les opérations oblige à réfléchir à des critères de segmentation particuliers permettant de garantir l'efficacité de calcul. Il faut donc changer son approche de l'analyse de l'homogénéité.

Le second point de la propriété 3.2 est également d'importance. En effet, généralement les algorithmes de segmentation en régions ont des structures de données basées sur des listes ou des graphes de régions, les algorithmes sont donc guidés par les régions et il devient difficile d'utiliser des critères locaux liés aux pixels en même temps que les critères régions. De fait, les algorithmes dit de coopération régions-contours sont alors difficiles à mettre en œuvre. Avec notre proposition, ils deviennent très faciles à implémenter puisque l'information part du pixel. De plus, rien n'empêche de lister les représentants des régions et de maintenir cette liste, sans dégrader la complexité, afin de pouvoir travailler au niveau des régions.

Grâce à cette modélisation, nous avons pu proposer deux algorithmes génériques de segmentation, *Scanline* et *MergeSquare* [84] (voir l'annexe A), en temps linéaire. Ces algorithmes sont génériques au sens où ils n'imposent pas de prédicat d'homogénéité et qu'ils sont utilisables avec n'importe quel prédicat respectant les propriétés 3.1. En fait, plus que l'intérêt théorique de la complexité linéaire obtenue (un algorithme quasi-linéaire en $O(m.\alpha(m, n))$ était suffisant d'un point de vue pratique), l'intérêt majeur de notre résultat réside dans la propriété 3.2. Ces deux caractéristiques fondamentales de notre approche ont permis d'envisager de nouveaux algorithmes de segmentation dont la complexité aurait été rédhibitoire sinon. C'est finalement le lien entre deux communautés, la communauté algorithmique et la communauté image qui a été le point le plus important de ces travaux. Nos différents articles, essentiellement [84] mais aussi [88, 90] (voir section 3.1.2), sont régulièrement cités pour cela.

Ainsi localement, j'ai pu établir depuis quelques années une collaboration avec le CEMAGREF et Gilles Rabatel sur des problèmes liés à la segmentation d'images. Cette collaboration a débuté en 2000 par le co-encadrement de Benoît de Mezzo, d'abord en DEA, puis en thèse [63]. Cette collaboration nous a amenés à travailler sur la problématique de la segmentation par collaboration régions-contours. La solution proposée s'appuyait sur l'*Union-Find* mais également sur le *Graphe Topologique des Frontières* (voir la section 3.2.2) afin de détecter et reconnaître des feuilles dans des images naturelles [64]. La collaboration s'est depuis approfondie avec notamment le co-encadrement de Nathalie Gorretta en DEA [107] puis en thèse (en cours) sur la problématique de la

segmentation d'images combinant information hyperspectrale et spatiale [17, 108, 110]

Bien entendu j'ai poursuivi l'étude même de la segmentation en régions en m'intéressant plus particulièrement au critère d'homogénéité. Ces travaux [88, 89, 90] ont été initiés avec Richard Nock et sont décrits plus en détail à la section 3.1.2. Mais ils ont également donné lieu à des suites, en premier lieu bien sûr par Richard Nock pour la segmentation d'images couleurs (voir par exemple [185, 184, 187]) mais également d'autres auteurs (voir par exemple [46]).

Mais l'élargissement à des domaines de l'imagerie, autres que la pure segmentation d'images en régions, est sans doute le plus intéressant et confirme bien l'aspect générique de notre approche. On pourra ainsi noter récemment R.D. Yapa et K. Harada [236] qui utilisent nos algorithmes comme outils de pré-traitement dans le cadre de l'imagerie médicale. Nos algorithmes sont utilisés ici comme outils d'étiquetage des composantes connexes. On retrouve plusieurs travaux utilisant ainsi nos algorithmes, que ce soit pour des images binaires [217], pour du suivi de contours [43] ou comme algorithmes de référence pour la mesure d'efficacité [117].

Ces algorithmes basés sur l'*Union-Find* ont aussi permis de faire découvrir l'*Union-Find* à d'autres domaines de l'imagerie comme la morphologie mathématique. Ainsi nos algorithmes sont-ils réutilisés et adaptés pour rendre plus efficaces les algorithmes de calcul des attributs morphologiques. On citera par exemple les résultats suivants [234, 173, 41, 182, 51] qui utilisent ou adaptent nos algorithmes de [84].

J'ai été le premier à utiliser notre résultat pour rendre plus efficace le calcul d'une représentation à base topologique des images segmentées en régions (voir [82] et la section 3.2). Naturellement d'autres auteurs ont fait de même [27, 40] ainsi bien sûr que Guillaume Damiani, un ancien étudiant que j'ai encadré en thèse et qui a poursuivi dans cette voie (voir par exemple [53, 21, 57, 111]). Jens Gustedt a quant à lui continué l'étude dans le cadre de l'algorithmique de graphes [114, 106].

Toutes ces suites, ainsi que les différents domaines abordés confirment l'impact sur le domaine de l'imagerie de notre approche algorithmique de la segmentation basée sur l'*Union-Find*.

3.1.2 Critères d'homogénéité pour l'Union-Find

D'un point de vue purement segmentation d'images, le reproche nous était souvent fait de ne pouvoir montrer de résultats pertinents de segmentation autrement qu'en temps de calcul. Ceci était lié au fait que nous propositions une solution algorithmique générique sans nous attarder sur la problématique du prédicat d'homogénéité qui lui décide de la qualité de la segmentation. C'est pourquoi, avec Richard Nock, nous nous sommes attaqués à ce problème, en nous attachant à proposer un critère d'homogénéité respectant la propriété 3.1 afin de garantir, non seulement la qualité du résultat par notre

proposition de prédicat, mais également son efficacité grâce aux algorithmes de [84].

Les résultats que nous présentions jusqu'alors utilisaient un critère d'homogénéité très simple : la différence de moyenne des niveaux de gris des deux régions examinées. Nous avons remarqué que ce critère très simple était aussi un bon critère. Certes, des critères plus performants existaient (voir par exemple [25, 194, 193, 123, 213, 230]), mais ceux-ci étaient beaucoup plus complexes et difficiles à mettre en œuvre. Et surtout, ils ne permettaient pas de garder la linéarité, et donc l'efficacité, de nos algorithmes. Nous avons donc décidé de chercher un critère plus performant que la simple différence de niveaux de gris et restant simple à mettre en œuvre. Pour cela, fidèle à la démarche qui m'avait amené à étudier la topologie des images numériques, nous avons d'abord cherché à définir un modèle théorique des images en niveaux gris, modèle sur lequel nous pourrions nous appuyer pour proposer des solutions prouvées aux caractéristiques bien définies.

Dans un premier temps je présenterai le modèle statistique d'image numérique que nous avons adopté. Puis, très brièvement, je donnerai un aperçu du critère original que nous avons proposé avec R. Nock. Le lecteur pourra trouver à l'annexe B deux articles [89, 90] présentant notre modèle. Enfin je citerai les suites qu'a suscitées notre proposition et je présenterai en perspective de nouveaux travaux entrepris en collaboration avec André Mas.

Modèle statistique d'image numérique

Dans le domaine de la segmentation d'images, de nombreux critères de segmentation déjà définis utilisent des modèles mathématiques et statistiques basés sur des hypothèses faites *a priori* sur les images [233, 239]. La plupart de ces hypothèses concernent la distribution des niveaux de gris dans l'image. Les contraintes imposées aboutissent souvent à des temps de calcul importants. A contrario, des solutions algorithmiques efficaces existent, telle que celle que nous avons proposée [84] mais font l'impasse sur le critère de segmentation et les résultats obtenus ne sont pas alors au niveau des méthodes statistiques. Nous avons donc cherché à combiner notre approche algorithmique, afin de garantir l'efficacité du processus de segmentation, et une approche statistique afin de garantir la pertinence du résultat. Notre proposition est basée sur un modèle statistique de génération d'images, ne faisant donc pas d'hypothèse sur la distribution des niveaux de gris. Les faibles contraintes du modèle nous permettent de définir un critère de segmentation calculable en temps constant et conservant les bonnes propriétés de complexité de notre solution.

Notre modèle est basé sur l'hypothèse qu'une image I est l'observation d'une scène idéale I^* , capturée sous des conditions particulières³. Formellement, dans I^* , les pixels

³Par la suite, on notera par l'exposant $*$, tous les objets issus de cette scène idéale, comme par

sont représentés par des distributions permettant de générer l'image observée. Une région homogène idéale serait alors telle que l'espérance mathématique de ses pixels serait égale. Celle-ci serait différente pour deux régions distinctes adjacentes. Ce modèle donne une intuition de ce que peut être une *segmentation optimale* : elle peut être obtenue par la moyenne d'un nombre infini d'échantillons, capturés sous un nombre infini de conditions d'observation. Les pixels ainsi obtenus ont alors comme niveau de gris leur espérance mathématique théorique et on peut donc observer les régions idéales.

Soit I l'image observée et $|I|$ le nombre de pixels de l'image. On note g , le nombre théorique de niveaux de gris dans l'image (généralement 256), g_I le nombre de niveaux de gris effectivement présents dans l'image (donc inférieur à g) et $|I|$ le nombre de pixels de l'image. En faisant l'hypothèse que les pixels sont échantillonnés indépendamment les uns des autres, notre modèle suppose que I est l'observation d'une scène idéale I^* dont nous ne connaissons rien et dans laquelle les pixels sont complètement représentés par une famille de *distributions* à partir de laquelle les niveaux de gris observés sont échantillonnés. Le $j^{\text{ème}}$ pixel $p_{i,j}$ de la $i^{\text{ème}}$ région R_i de l'image I est alors décrit par une variable aléatoire $p_{i,j}^* \in I^*$, elle-même somme de Q variables aléatoires indépendantes $p_{i,j,1}^*, p_{i,j,2}^*, \dots, p_{i,j,Q}^*$ de I^* , chacune prenant ses valeurs dans l'intervalle $[0, \frac{g}{Q}]$. Cette hypothèse est illustrée par la figure 3.4. Q sert essentiellement à s'assurer de la pertinence pratique de notre test d'homogénéité. En fait, plus l'image est grande et plus les régions sont testées, plus Q peut être choisi petit.

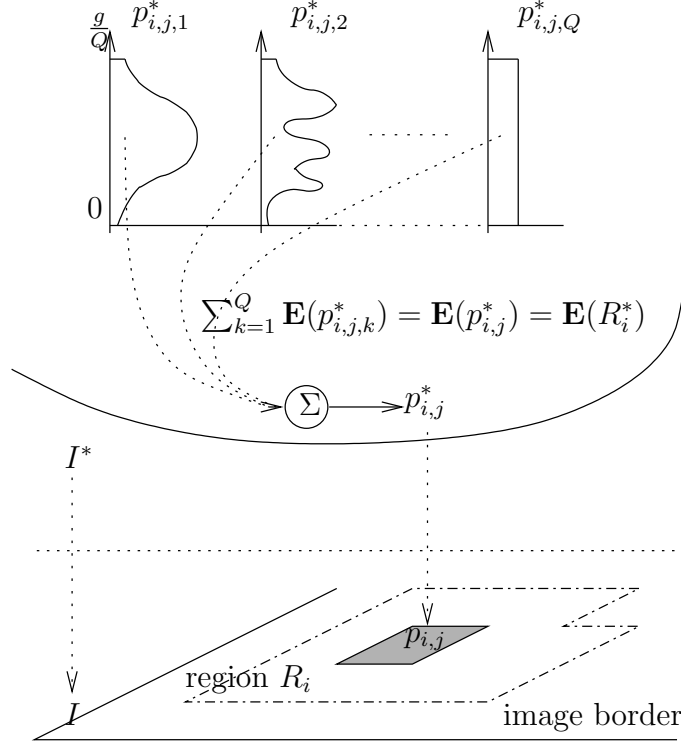
Comme le veut notre modèle, chaque région R_i^* de la scène idéale I^* satisfait à la propriété d'homogénéité 3.1.

Propriété 3.1 (homogénéité). $E(\cdot)$ dénotant l'espérance mathématique, une région R_i^* homogène de I^* satisfait :

1. $\forall i \in \{1, 2, \dots, |I|\}, \forall j \in \{1, 2, \dots, |R_i^*|\}, E(p_{i,j}^*) = E(R_i^*)$ (les espérances de tous les pixels de R_i^* sont égales) ;
2. tout couple (R_i^*, R_j^*) de régions adjacentes satisfait l'inégalité $E(R_i^*) \neq E(R_j^*)$ (deux régions adjacentes ont des espérances différentes).

Notre modèle statistique d'image repose donc sur le fait que chaque pixel est supposé être échantillonné à partir d'une famille de distributions représentant les propriétés intrinsèques de luminosité de la zone qu'il décrit, et que chaque région est homogène au sens où l'espérance de chaque pixel de la région est la même. Cette homogénéité suppose donc que la variance est nulle. Afin de réaliser la segmentation de l'image observée, nous allons nous appuyer sur cette dernière hypothèse et mesurer l'augmentation de la variance lorsqu'on agrège de nouveaux pixels à une région donnée.

exemple R^* pour une région de I^* à laquelle correspond une région observée R de I .

FIG. 3.4 – Un pixel dans I^* et I (on note $\mathbf{E}(\cdot)$ l'espérance).

Critère d'homogénéité et inégalité de concentration

Notre prédicat d'homogénéité est donc basé sur l'augmentation de la variance intra-région provoquée par la fusion éventuelle des deux régions considérées. Ce que nous avons cherché à évaluer est la probabilité de rejeter faussement la fusion de deux régions qui en fait seraient des parties de la même région R^* de I^* . $\Delta_i(R, R')$, l'augmentation de la variance intra-région, est donnée par le lemme 3.1. Grâce à l'inégalité de concentration de M. Diarmind (voir le théorème 3.1), on peut borner la probabilité que $\Delta_i(R, R')$ dépasse un seuil fixé. Cette borne maximum est donnée par le théorème 3.1 rappelé ci-dessous et proposé dans [89].

Lemme 3.1 (Lemme 1 de [89]). $\Delta_i(R, R') = \frac{|R| \times |R'|}{Q(|R| + |R'|)} (\bar{R} - \bar{R}')^2$

Théorème 3.1 (independent boudned difference inequality [170]). Soit $X = (X_1, X_2, \dots, X_n)$ une famille de n variables aléatoires indépendantes, $X_k \in A_k$, en supposant que la fonction à valeurs réelles f définie sur $\prod_k A_k$ satisfait l'inégalité $\|f(\mathbf{x}) - f(\mathbf{x}')\| \leq c_k$ pour des vecteurs $\mathbf{x}$ et $\mathbf{x}'$ ne différant que sur la $k^{\text{ième}}$ coordonnée, et soit μ

l'espérance de la variable aléatoire $f(X)$, alors $\forall \tau \geq 0$,

$$\Pr[f(X) - \mu \geq \tau] \leq e^{\left(\frac{-2\tau^2}{\sum_k (c_k)^2}\right)} \quad (3.5)$$

Théorème 3.1 (Théorème 1 de [89]). Soient R et R' deux régions adjacentes de I , parties de la même région R^* de I^* . Soit

$$m = \frac{6\sqrt{2}g^2 \sqrt{\max\{|R|, |R'|\}(|R|(|R| + 2|R'|)^2 + |R'|^3)}}{Q^3 |R| \cdot |R'| (|R| + |R'|)}$$

et supposons que $g, Q, |R|, |R'|$ sont tels que $m < 1/2$, alors soit τ , tel que $0 < \tau < 1$, on a :

$$\Pr[\Delta_i(R, R') \geq \tau] < \frac{8\#R\#R'}{1 - 2m} \times e^{-\frac{2Q^2(|R|+|R'|)^2\tau}{g^2|R||R'|}} \quad (3.6)$$

D'un point de vue pratique, il est assez difficile d'utiliser tel quel ce résultat, d'abord parce que cette probabilité reste difficile à calculer et ensuite parce que nous ne disposons pas encore d'un prédicat nous permettant de répondre directement à la question de la pertinence de la fusion potentielle de deux régions. Aussi avons-nous proposé des approximations pratiques permettant, à partir de l'équation (3.6) du théorème 3.1, de proposer une fonction $\tau(R, R')$ de seuil adaptatif en fonction des régions à fusionner (pour plus de détail sur cette fonction et son calcul, voir [89] présenté à l'annexe B). Le prédicat d'homogénéité est alors le suivant :

Définition 3.2 (Prédicat d'homogénéité). $\mathcal{P}(\mathcal{R}, \mathcal{R}') = \text{VRAI}$ si $\Delta_i(R, R') < \tau(R, R')$

L'originalité de notre approche tient à deux points clefs :

1. nous considérons la segmentation en régions comme la reconstruction d'une image idéale à partir d'une observation : c'est le sens de notre modèle statistique d'images présenté à la section 3.1.2 ;
2. nous nous appuyons sur les inégalités de concentration afin de proposer un critère statistique pertinent et efficace combiné à des algorithmes dont la complexité linéaire en temps a été démontrée.

Le résultat obtenu est une famille d'algorithmes efficaces qui ont une forte probabilité de donner un résultat ne présentant qu'une seule source d'erreur : la sur-segmentation, c'est-à-dire le fait qu'une région obtenue contienne en fait plusieurs régions R^* de l'image idéale I^* . En général, on n'observe pas de sous-segmentation, ni le cas d'erreur le plus fréquent et le pire où des régions recouvrent partiellement plusieurs régions idéales R^* .

On trouvera à l'annexe B les deux articles [89, 90] présentant ces résultats, le premier s'intéressant essentiellement à la partie statistique alors que le second se concentre sur les

aspects algorithmiques. J'ai alors commencé l'encadrement en thèse de Guillaume Damiani et j'ai donc concentré mes efforts sur les représentations à base topologique (voir la section 3.2) et par manque de temps j'ai laissé de côté ce sujet. Mais mon co-auteur Richard Nock a continué à explorer cette voie, proposant d'abord une extension de nos résultats à la couleur [185], puis, avec l'aide de Franck Nielsen, il a continué à améliorer ces résultats [184]. R. Nock et F. Nielsen ont alors proposé dans [186] une version simplifiée et améliorée de nos résultats de [89, 90] ainsi qu'une synthèse de leurs propres résultats. D'autres travaux se sont ensuite inspirés de cette approche en l'appliquant à d'autres types de segmentation, notamment semi-supervisée (voir par exemple [187, 128] ou le survol [238]).

Analyse critique et perspectives

En fait, la problématique du prédicat d'homogénéité est relative à un test s'appuyant sur les deux hypothèses suivantes :

$$\begin{cases} H_0 : \text{Les régions } R \text{ et } R' \text{ doivent être fusionnées} \\ H_1 : \text{Les régions } R \text{ et } R' \text{ ne doivent pas être fusionnées} \end{cases} \quad (3.7)$$

Or deux types d'erreur peuvent apparaître lorsqu'on implémente un test basé sur ces hypothèses :

type I : rejeter faussement H_0

type II : rejeter faussement H_1

Si l'on note α la probabilité de l'erreur de type I et β celle de l'erreur de type II, il est bien connu qu'on ne peut pas rendre, en même temps, α et β arbitrairement petites [162].

Habituellement les prédicats d'homogénéité sont basés sur le contrôle de l'erreur de type I sans se préoccuper de l'erreur de type II. C'est ce que nous avons proposé avec R. Nock (voir la section 3.1.2). L'inconvénient principal de cette démarche est clairement une sur-segmentation, c'est-à-dire qu'on n'accepte pas suffisamment souvent l'hypothèse H_1 et que l'on fusionne donc des régions qui n'auraient pas dû l'être.

Avec A. Mas, nous avons commencé à chercher une procédure adaptative qui permet de contrôler les deux types d'erreur. Par adaptative, on entend la même chose que le critère que nous avons proposé avec R. Nock, c'est-à-dire que le seuil du test dépend des caractéristiques de second ordre (c'est-à-dire la variance des régions) des régions considérées. Il n'est pas global pour l'image.

Pour arriver à cela, nous avons regardé d'autres inégalités de concentration que celle de Mc Diarmid [170] utilisée dans [89, 90, 88, 184, 185, 186, 187]. En fait, l'inégalité de M. Diarmid est extrêmement générale. Comme le test d'homogénéité est basé sur une comparaison des moyennes ou des variances, on devrait pouvoir arriver à de meilleurs

résultats et surtout à contrôler les deux types d'erreur en utilisant une inégalité plus spécifiquement conçue pour les sommes de variables indépendantes. Ainsi l'inégalité de Bennett prend en compte la variance des variables aléatoires et nous pouvons donc raisonnablement espérer qu'elle soit plus précise dans notre cas d'utilisation. Nous donnons ici l'inégalité dans sa formulation générale avec les notations de Ledoux-Talagrand [160] :

$$\mathbb{P}\left(\left|\sum_i V_i\right| > t\right) \leq 2e^{\left(\frac{t}{a} - \left(\frac{t}{a} + \frac{b^2}{a^2}\right) \log\left(1 + \frac{at}{b^2}\right)\right)} \quad (3.8)$$

Où les variables V_i aléatoires réelles indépendantes et **centrées** sont telles que $|V_i| \leq a$ et $b^2 = \sum_i \mathbb{E}V_i^2$. Puisque $\forall i, \mathbb{E}V_i^2 = \sigma^2$ alors $b^2 = n\sigma^2$ et en posant $x = at/\sigma^2$

$$\mathbb{P}\left(\left|\sum_i V_i\right| > nt\right) \leq 2e^{\left(-\frac{n\sigma^2}{a^2}((1+x)\log(1+x)-x)\right)} \quad (3.9)$$

Notre but est donc de déterminer le seuil minimum x_t^* tel que $\mathbb{P}(|\sum_i V_i| > nt) < \alpha$ (α étant l'erreur de type I). La solution est donc

$$x_t^* = \min \left\{ x > 0 : 2e^{\left(-\frac{n\sigma^2}{a^2}\varphi(x)\right)} \leq \alpha \right\} \quad (3.10)$$

où $\varphi(x)$ est la fonction de concentration donnée par l'équation (3.11) suivante :

$$\varphi(x) = (1+x)\log(1+x) - x. \quad (3.11)$$

La solution est donc de prendre $x_t^* = \varphi^{-1}\left(\frac{a^2}{n\sigma^2} \log\left(\frac{2}{\alpha}\right)\right)$. Malheureusement φ^{-1} n'est pas connue. On déterminera donc une approximation $\varphi^\dagger \geq \varphi^{-1}$ par une variante d'un algorithme de Newton :

$$\varphi^\dagger(y) = \begin{cases} \varphi_1^\dagger(y) = \frac{y}{3} + \sqrt{\frac{y^2}{9} + 2y} \text{ si } 0 \leq y \leq 3(\log 3) - 2 \\ \varphi_2^\dagger(y) = \frac{y + 2.3}{\log 3.3} - 1 \text{ si } 3(\log 3) - 2 \leq y \leq 4(\log 4) - 3 \\ \varphi_3^\dagger(y) = \frac{y}{\log\left(\frac{y}{\log y} + 1\right)} \left(1 + \frac{1}{\log y}\right) - 1 \text{ si } 4(\log 4) - 3 \leq y \end{cases} \quad (3.12)$$

Sachant (3.9), (3.10), (3.12), et en considérant une région comme une variable aléatoire discrète bornée par $a = 255$, nous sommes à même de démontrer le théorème 3.2 sur l'erreur de type I et le théorème 3.3 sur l'erreur de type II.

Théorème 3.2. *Soit*

$$\begin{aligned} M &= \frac{a^2}{|R| \sigma^2} \log \left(\frac{4}{\alpha} \right), \\ M' &= \frac{a}{|R'| \sigma'^2} \log \left(\frac{4}{\alpha} \right), \\ t_I(\alpha) &= \frac{\sigma^2}{a} \varphi^\dagger(M) + \frac{\sigma'^2}{a} \varphi^\dagger(M'), \end{aligned}$$

Si les deux régions doivent être fusionnées (i.e. si $m = m'$), alors $\mathbb{P}(|S - S'| > t_I(\alpha)) \leq \alpha$ et donc si $|S - S'| \leq t_I(\alpha)$ on peut fusionner les régions.

Soit $\Delta \in \mathbb{N}$ la différence minimale de niveaux de gris entre deux régions idéales en delà de laquelle on doit toujours refuser la fusion⁴, on peut démontrer le théorème suivant sur l'erreur de type II :

Théorème 3.3. *Si $|\mathbb{E}X - \mathbb{E}X'| \geq \Delta$ (i.e. R et R' doivent donc être considérées comme des régions n'appartenant pas à la même région idéale), $\mathbb{P}(|S - S'| < \Delta - t_I(\beta)) < \beta$ et donc si $|S - S'| \geq \Delta - t_I(\beta)$ on ne doit pas fusionner les régions.*

De ces deux théorèmes, on peut en déduire la procédure de test suivante pour notre processus de segmentation :

Proposition 3.4 (Prédicat d'homogénéité).

- Si $|S - S'| \leq t_I(\alpha)$ alors fusionner R et R' ;
- Si $|S - S'| \geq \Delta - t_I(\beta)$ alors ne pas fusionner R et R'

Nous sommes également capables d'apporter une réponse lorsque les régions à fusionner tombent dans la zone d'incertitude, c'est-à-dire lorsque $t_I(\alpha) < |S - S'| < \Delta - t_I(\beta)$. Plus de détails sur ces nouveaux travaux pourront être trouvés dans l'article complet, encore en cours de finalisation.

On peut donc voir à travers ces derniers résultats que le modèle d'image que nous avons proposé avec R. Nock permet de nouveaux développements et qu'il permet de proposer des critères de segmentation pertinents et performants.

⁴Clairement Δ sera un paramètre du processus de segmentation

3.2 Représentation à base topologique des images de régions

Une fois une première segmentation effectuée, basée uniquement sur des critères liés à l'homogénéité chromatique des régions, le seul moyen d'améliorer l'analyse d'une image est d'introduire de la connaissance dans le processus de reconnaissance. Les algorithmes de reconnaissance nécessitent alors de s'appuyer sur des structures de représentation évoluées, généralement à base de graphes. J'ai donc eu, dès mes premiers travaux en thèse, la volonté de modéliser les images par une structure combinatoire adaptée aux traitements algorithmiques. La structure habituellement utilisée en imagerie est le graphe d'adjacence des régions (R.A.G en abrégé) [198, 190]. Or en 1989, V.A. Kovalevsky a montré [146] que les graphes de voisinage, famille dont fait partie le R.A.G, ne respectent pas les axiomes de base d'une topologie. La structure que proposait alors V.A. Kovalevsky n'était pas non plus satisfaisante, puisque le squelette 1D du complexe dual d'un *Block Cells* est équivalent au R.A.G. Nous avons donc cherché à proposer une structure combinatoire à base topologique permettant de représenter les images segmentées en régions. Trois points importants devaient donc être respectés :

- coder les frontières des régions, et pas seulement l'adjacence ;
- gérer les inclusions de régions les unes dans les autres ;
- être une structure à base de graphes.

Nous nous sommes alors intéressés aux structures des graphes planaires puis ensuite tout naturellement aux cartes combinatoires (voir la section 3.2.1), sortes de multi-graphes planaires codant la topologie des objets. Nous avons alors proposé le *Graphe Topologique des Frontières* (T.G.F en abrégé) [81, 3] dont nous rappellerons les principes à la section 3.2.2. En collaboration avec Yves Bertrand, j'ai ensuite encadré un étudiant de DEA sur un nouveau modèle extensible en dimension n [24]. Dans le même temps nous co-encadrions Guillaume Damiand dont le sujet était la modélisation topologique des images segmentées 2d et 3D [21, 22, 53]. Nous présentons brièvement ces travaux à la section 3.2.2. On pourra trouver une longue synthèse de nos résultats sur la modélisation topologique des images 2d dans l'article [55] joint à l'annexe C. Je terminerai ce chapitre par deux sections, la section 3.2.2 présentant les travaux concomitants d'autres auteurs 3.2.2 alors que la section 3.2.3 présente quelques travaux qui se placent dans la lignée de ce que nous avons proposé et notamment les travaux que Guillaume Damiand a poursuivi en tant que Maître de Conférence et maintenant Chargé de Recherche au CNRS.

3.2.1 Introduction aux cartes combinatoires

Les *cartes combinatoires* s'inscrivent dans le cadre des modèles de représentation par contours. Dans ce type de représentation les objets (les solides) sont définis par une subdivision en faces (et ces faces en sommets et arêtes⁵). Ce modèle topologique est complété par un modèle de plongement définissant la projection de l'objet dans $\mathbb{R}^n$. Y. Bertrand et J.F. Dufourd affirment dans [23] que les cartes et leurs extensions offrent un avantage décisif car elles sont définies à partir d'un seul élément de base, le *brin*, et qu'elles utilisent des opérateurs simples à mettre en œuvre : des involutions et des permutations. Ce qui implique une grande homogénéité dans la représentation et une généralisation immédiate à une dimension n . En fait la notion de carte est apparue en 1960 [76], puis a été reprise en 1970 en tant que graphe topologique [126]. R. Cori [48] en a le premier donné une définition algébrique [48]. Ce modèle a ensuite été repris et développé, essentiellement par P. Lienhardt en tant que modélisation géométrique à base topologique [101, 163, 164, 165, 166].

Les cartes combinatoires sont un modèle mathématique de représentation des subdivisions de l'espace. Une subdivision de l'espace est une partition d'un espace topologique de dimension n en $(n + 1)$ parties dont les éléments sont des *cellules* de dimension $0, 1, 2, 3, \dots, n$ (respectivement appelées sommets, arêtes, faces et volumes). Des relations d'incidence sont définies entre ces cellules. En fait l'ensemble des $(j < i)$ -cellules⁶ incidentes à une i -cellule forme le bord de la i -cellule. On dira que deux i -cellules sont adjacentes si elles sont incidentes à la même $(j < i)$ -cellule. Une carte combinatoire code toutes les subdivisions et les relations d'incidences entre les cellules de l'espace et code donc bien la topologie de cet espace. Formellement, une carte combinatoire de dimension n (n -carte) peut représenter toute variété orientable d'un espace de dimension n . P. Lienhardt a généralisé ce modèle afin de pouvoir représenter toute subdivision, orientable ou non, d'un espace de dimension n . On pourra trouver dans [165] la définition des *cartes généralisées*, nous donnons ici celle des cartes combinatoires en dimension n .

Définition 3.3 (carte combinatoire de dimension 3). Soit $n \geq 0$. Une n -carte combinatoire est un $(n + 1)$ -tuple $M = (D, \beta_1, \dots, \beta_n)$ tel que :

1. B est un ensemble fini de brins ;
2. β_1 est une *permutation* sur B ;
3. $\forall i$ tel que $2 \leq i \leq n$, β_i est une *involution* sur B ;
4. $\forall i$ tel que $1 \leq i \leq n$, $\forall j$ tel que $i + 2 \leq j \leq n$, $\beta_i \circ \beta_j$ est une *involution*.

⁵On retrouve donc les notions de cellules de dimension 0, 1, 2 et 3 utilisées dans la topologie-étoile

⁶On note i -cellule une cellule de dimension i .

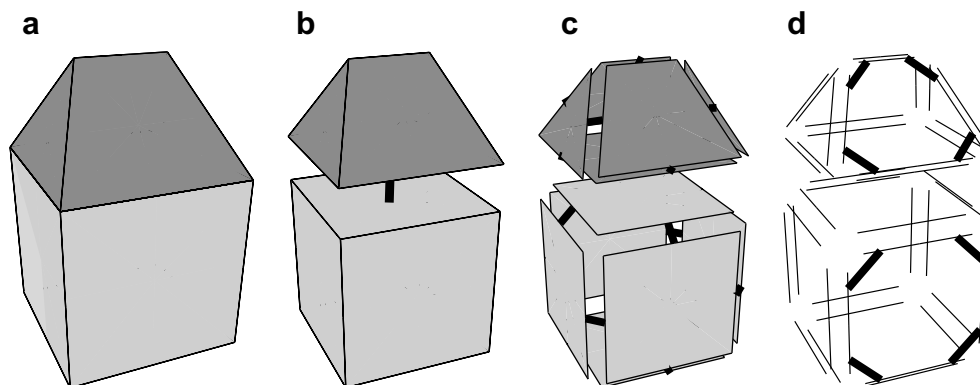


FIG. 3.5 – Exemple de subdivision d'un objet de dimension 3

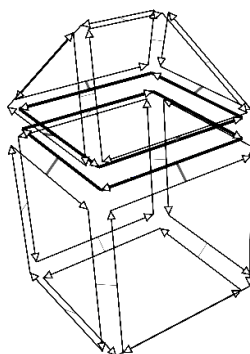


FIG. 3.6 – Carte combinatoire correspondant à l'objet de la figure 3.5

Lorsqu'on a deux brins, d_1 et d_2 tels que $\beta_i(d_1) = d_2$ ($1 \leq i \leq n$), on dit que d_1 est *i-cousu* à d_2 . La figure 3.5 montre un exemple de décomposition en cellules d'un objet 3d et la figure 3.6 montre la 3-carte correspondante.

L'autre notion importante dans les cartes, est la notion d'*orbite* donnée par la définition 3.4. Alors qu'une carte combinatoire est définie à partir d'éléments abstraits, la notion d'orbite permet de définir les cellules qui correspondent aux subdivisions topologiques de l'espace :

Définition 3.4 (Orbite). Soit $\Phi = \{f_1, \dots, f_k\}$ des permutations sur B , on note $\langle \Phi \rangle$ le groupe de permutations engendré par Φ : c'est l'ensemble des permutations pouvant être obtenues à partir de compositions et d'inversions des permutations de Φ . On appelle $\langle \Phi \rangle (b) = \{\phi(b) | \phi \in \Phi\}$ l'*orbite* de b relativement à Φ .

Par exemple, en dimension 2, les sommets sont définis par l'orbite $\langle \beta_{21} \rangle$, les arêtes par l'orbite $\langle \beta_2 \rangle$ et les faces par l'orbite $\langle \beta_1 \rangle$.

Les cartes combinatoires représentent seulement la topologie des objets, pas leur

géométrie. Mais il est facile d'associer à certaines (ou toutes) des orbites une géométrie. On appelle cette opération le *plongement*. Dans notre cas, le plongement correspondra à la description du contour associé à chaque arête. Un simple code de Freeman [96] pourra faire l'affaire.

Cette séparation de la topologie et de la géométrie est un des avantages majeurs des cartes combinatoires. En effet, la plupart des opérations sur les cellules, en particulier dans le cadre de l'analyse d'images, n'ont pas d'incidence sur la géométrie. Par exemple l'opération de base, la fusion de régions, est une opération topologique ; la géométrie du contour de la région ne change pas. Cette séparation permet donc d'optimiser les opérations en s'affranchissant des problèmes liés à la géométrie.

C'est donc naturellement que nous avons proposé de nous appuyer sur ce modèle pour proposer des modélisations combinatoires des images segmentées. La section 3.2.2, qui suit, présente ces résultats.

3.2.2 TGF et Carte des bords

Outre le problème de la modélisation en elle-même de l'image, il était aussi important de proposer une méthode efficace pour calculer cette représentation à partir d'une image segmentée. En effet, nous ne sommes pas dans le cadre classique de la modélisation géométrique des images puisque ce n'est pas l'utilisateur qui crée l'image « à la main », mais dans le cadre d'un processus complet d'analyse d'images. Notre problématique est, après une segmentation d'images en régions, de construire une représentation permettant d'utiliser des algorithmes utilisant des connaissances *a priori* sur l'image. Il est donc important de proposer, non seulement une modélisation s'appuyant sur la topologie des images, mais aussi un algorithme efficace permettant de la calculer. L'originalité de nos travaux repose essentiellement sur cette double proposition d'un modèle associé à un algorithme d'extraction et de l'utilisation de *précodes* afin de permettre ce calcul en une seule passe sur l'image.

TGF

En fait, c'est après avoir travaillé sur un problème de segmentation d'images à base de graphe d'adjacence [2] que je me suis rendu compte de la nécessité de disposer d'une bonne représentation des images. J'ai donc choisi de m'appuyer sur les cartes combinatoires présentées à la section 3.2.1 pour proposer le *graphe topologique des frontières* (voir la définition 3.5). À l'époque, je ne voulais pas utiliser directement les cartes combinatoires, en effet j'avais besoin que la notion de région soit explicitement représentée, et donc codée par les sommets du graphe, ce qui signifie un passage au dual en terme de graphe. De plus, une image segmentée comporte des « trous », c'est-à-dire des régions entièrement incluses dans une autre, ce que ne permet pas de coder directement

les cartes. J'ai donc rajouté cette propriété à mon modèle. On obtient donc la définition suivante (plus de précisions peuvent être trouvées dans [81, 82]) et un exemple de T.G.F est donné à la figure 3.7.

Définition 3.5 (Graphe Topologique des Frontières). Un graphe topologique des frontières $G(V, D, \alpha, \sigma)$ où V désigne l'ensemble des sommets du graphe, D l'ensemble des demi-arêtes, σ une involution sur D et α une permutation sur D possède les caractéristiques suivantes :

1. Chaque région est représentée par un et un seul sommet.
2. Une demi-arête (brin) e est toujours incidente à un sommet R de V .
On dit que e est incidente à R , c'est-à-dire une région (voir représentation ci-contre). On utilisera la notation e^R si l'on veut préciser le sommet d'incidence de la demi-arête.
3. Une arête du graphe est un couple non ordonné $(e, \sigma(e))$ où $e \in D$. Soit E l'ensemble des arêtes, $(e_1^R, e_2^{R'}) \in E$ si et seulement si il existe une frontière entre R et R' . Une arête représente donc une frontière. On dira alors qu'une demi-arête symbolise la frontière "vue" du côté de la région dont elle est incidente.
4. α est une permutation sur l'ensemble D dont les cycles sont des permutations circulaires sur un ensemble de demi-arêtes incidentes à un même sommet. Chaque cycle correspond à un contour d'une région donnée et respecte l'ordre induit par la suite des frontières composant le contour.
5. $\infty \in D$ est un sommet particulier du graphe correspondant à l'extérieur de l'image. Celui-ci est donc vu comme une région englobant les régions de l'image.
6. À chaque sommet on associe la liste des cycles de α correspondant aux contours de la région correspondante à ce sommet. Le contour extérieur, s'il existe⁷, sera toujours par convention le premier de la liste.



Comme dit précédemment, une des originalités de cette approche tient à la proposition d'un algorithme d'extraction optimal de la structure basé sur une notion de précode : l'algorithme 1 consiste donc à parcourir l'image en construisant un graphe au fur et à mesure du balayage.

Puisque notre image a été segmentée à l'aide de nos algorithmes présentés à la section 3.1, nous avons à notre disposition une structure Union-Find (voir la section 3.1.1) nous permettant de faire le lien entre un pixel et sa région. Ainsi le balayage est effectué en déplaçant, de gauche à droite et de haut en bas, une fenêtre 2×2 sur l'image. La première fenêtre sera placée de telle sorte à ne couvrir que le premier pixel de l'image⁸. La

⁷ ∞ est l'unique sommet n'ayant pas de contour extérieur, mais un unique contour intérieur

⁸On considère que l'image est entourée d'une région extérieure qui sera représentée dans le graphe par un sommet particulier.

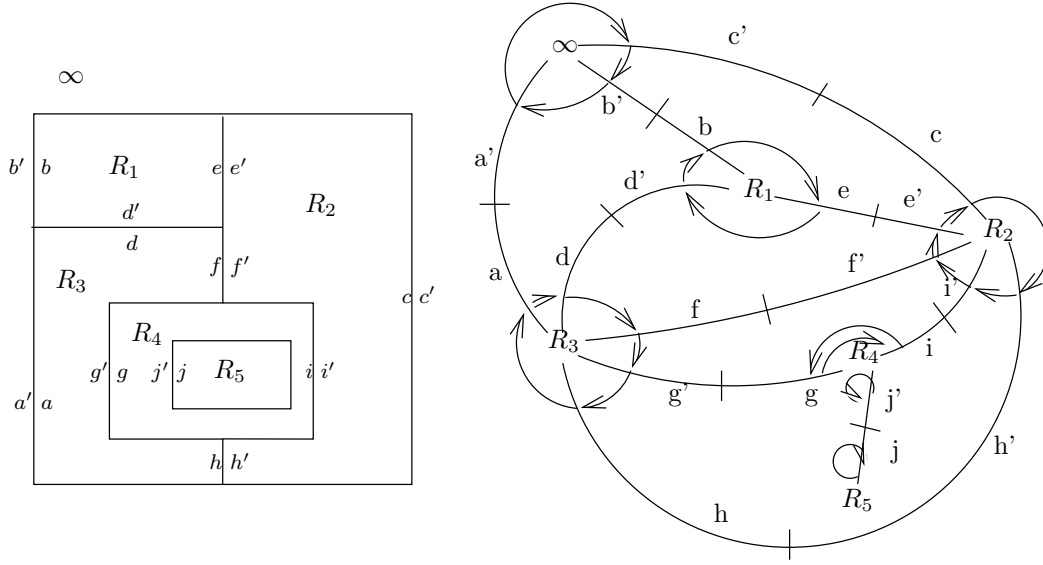


FIG. 3.7 – Un exemple de graphe topologique des frontières

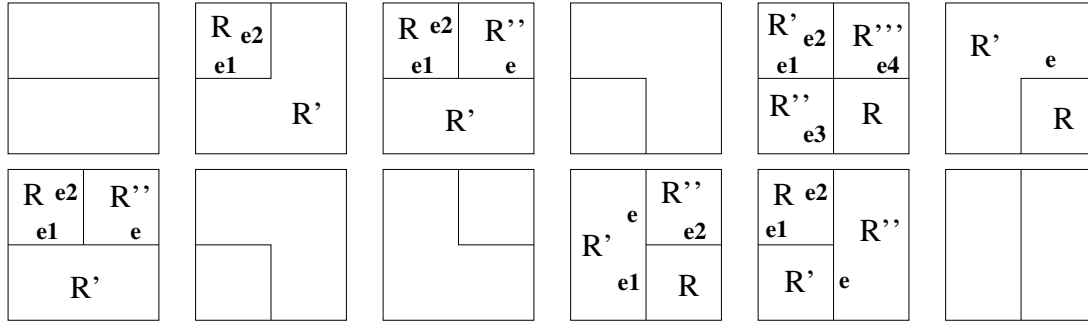


FIG. 3.8 – les 12 précodes possibles en dimension 2

configuration des frontières dans cette fenêtre, appelée *précode*, indique les traitements à effectuer pour construire le graphe. Il est facile de voir qu'il n'y a que 12 configurations possibles (voir figure 3.8).

Une liste L nous permet de retrouver les arêtes du graphe correspondant aux frontières présentes dans le précode. Ces mêmes arêtes nous donnent les sommets correspondant aux régions de l'image visibles dans le précode. En fonction du précode, nous créons des sommets et des arêtes, nous maintenons à jour les relations α et σ entre demi-arêtes, nous fusionnons des arêtes et des sommets. À la fin du balayage, le graphe obtenu est le graphe topologique des frontières (voir l'algorithme 1).

On pourra trouver dans [81, 82] une description complète et détaillée des précodes. Puisque cette modélisation devait servir à des processus d'analyse d'images, nous avons

Algorithm 1: Extraction du *Graphe topologique des Frontières*

Données: une image

Résultat: le graphe topologique des frontières

- 1 Initialiser le graphe avec un sommet isolé ∞ ;
 Initialiser D (ensemble des demi-arêtes) à vide;
 Initialiser la liste L à vide;
pour chaque précode de l'image faire
 └ Exécuter le code associé au précode courant;
-

également étudié dans [3] les différentes opérations, notamment celle de fusion de régions, nécessaire à un processus d'analyse d'images.

Carte des bords

Si le graphe topologique des frontières est bien adapté au cas 2d, son extension en dimension 3 semblait difficile. C'est pourquoi, dans la perspective de représenter des images 3d segmentées, notamment dans le cadre d'une collaboration en imagerie médicale, nous avons dû revenir à une représentation plus proche de celle des cartes combinatoires [24, 21, 22], tout en étant toujours capable de calculer efficacement cette représentation à partir d'une image segmentée.

Nous proposons avec G. Damiand plusieurs niveaux de cartes combinatoires, le dernier étant celui de la *carte topologique*, duale en 2d du graphe topologique des frontières, sans la notion d'inclusion, gérée ici par un arbre d'inclusion des composantes connexes. L'avant-dernier niveau, la *carte des bords*, est le niveau de carte qui est extensible en dimension n . En effet le dernier niveau, notamment à cause des problèmes d'inclusion qui sont beaucoup plus complexes en dimension 3, pose des difficultés à être étendu en dimension supérieure. On donnera ici les définitions en dimension 2 de ces différents niveaux. On pourra trouver à l'annexe C la version complète de l'article présentant ces résultats. On pourra alors noter que, si les définitions sont données de manière algorithmique, construisant un niveau à partir de l'autre, nous proposons des algorithmes à partir de précodes qui permettent d'extraire la carte directement au niveau choisi. Nous avons donc repris les méthodes développées pour le T.G.F.

Définition 3.6 (carte complète). La carte de niveau 0 correspondant à une image de $n \times m$ pixels, est la carte composée de $n \times m$ faces carrées β_2 -cousues entre elles, chaque face correspondant à un pixel, avec en plus une face entourant toutes les autres et correspondant à la région infinie.

Cette carte code tous les éléments interpixels de l'image. La figure 3.9(b) montre la carte de niveau 0 correspondant à l'image de la figure 3.9(a).

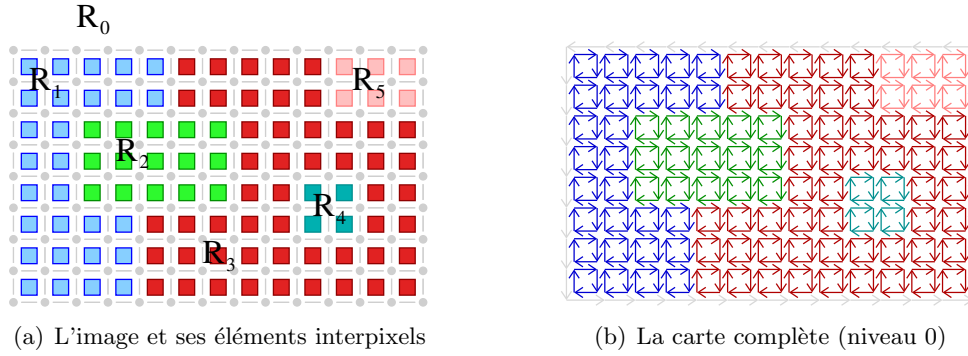


FIG. 3.9 – Carte complète de niveau 0 d'une image.

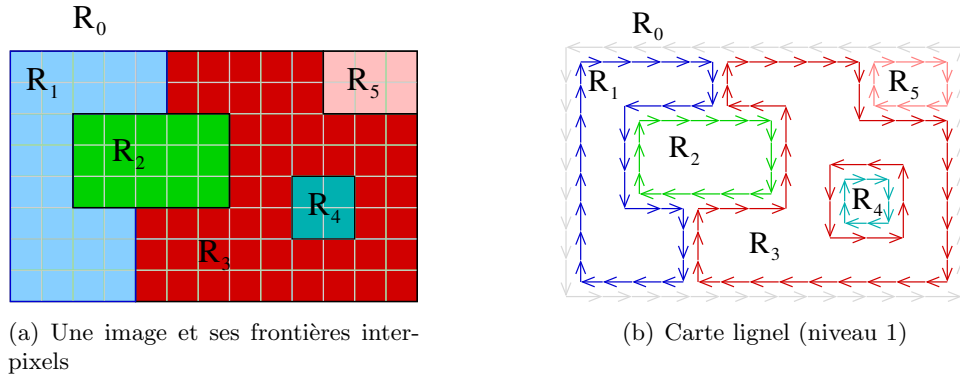


FIG. 3.10 – Carte lignal de niveau 1 d'une image.

La carte complète code tous les éléments interpixels de l'image. Pour obtenir une carte qui code les frontières interpixels, il faut enlever les arêtes de la carte de niveau 0 correspondant à des lighels d'une frontière. Cette opération se fait facilement grâce à un opérateur topologique. On obtient alors la *carte lignal* ou carte de niveau 1 :

Définition 3.7 (carte lignal). La carte de niveau 1 est la carte obtenue à partir de la carte de niveau 0 en fusionnant chaque couple de faces adjacentes appartenant à la même région.

La figure 3.10(a) montre l'image de la figure 3.9(a) représentée avec ses frontières interpixels et la figure 3.10(b) montre la carte correspondante.

Il est facile de vérifier que cette carte représente bien toutes les frontières de l'image et que chaque frontière interpixels est bien codée dans la carte.

Proposition 3.5. À chaque arête de la carte lignal correspond exactement un lighel d'une frontière de l'image, et chaque lighel d'une frontière de l'image est représenté par une arête dans la carte lignal.

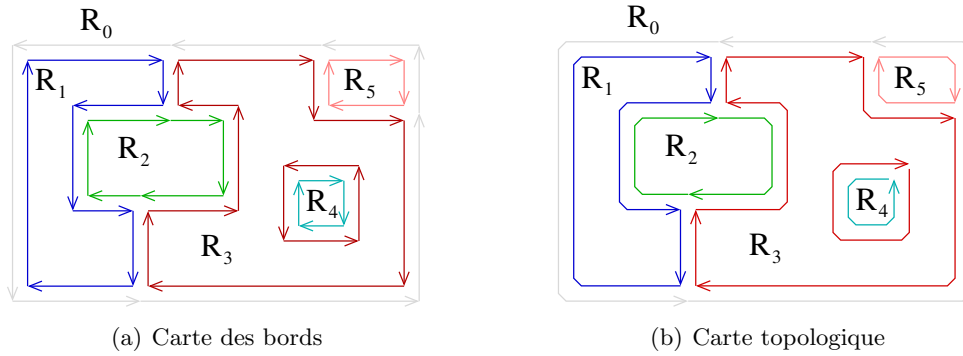


FIG. 3.11 – Carte des bords et topologique (niveau 2 et 3)

La *carte des bords* correspond à la carte présentée dans [24]. Dans cette carte, chaque arête représente un segment droit d'une frontière. La définition 3.8 est obtenue en construisant la carte de niveau 2 à partir de la carte de niveau 1.

Définition 3.8 (carte des bords). La carte de niveau 2 est la carte obtenue à partir de la carte de niveau 1 en fusionnant chaque couple d'arêtes alignées adjacentes à un sommet de degré 2.

La carte des bords de l'image de la figure 3.10(a) est présentée à la figure 3.11(a). La principale propriété de cette carte est que chaque arête représente un segment de droite d'une frontière interpixels de l'image.

Proposition 3.6. *Chaque arête d'une carte des bords correspond à un segment de droite maximal d'une frontière de l'image.*

Enfin, afin d'arriver à une véritable représentation topologique, nous devons fusionner les arêtes incidentes à des sommets de degré 2 afin de n'avoir qu'une seule arête pour représenter une frontière. On obtient alors une carte topologique :

Définition 3.9 (carte topologique). La carte de niveau 3 est la carte obtenue à partir de la carte de niveau 2 en fusionnant chaque couple d'arêtes incidentes à un sommet de degré 2.

La figure 3.11(b) présente un exemple de carte topologique. La principale propriété de la carte topologique est donc la suivante :

Proposition 3.7. *Chaque arête d'une carte topologique correspond exactement à une frontière de l'image représentée.*

Cette carte topologique est, dans le cas de la dimension 2, le dual de notre graphe topologique des frontières sans l'inclusion des régions. En revanche, cette définition, niveau par niveau, permet de l'étendre à n'importe quelle dimension.

Nous avons également proposé des algorithmes, à base de précodes, permettant de calculer chaque niveau de la carte directement (voir l'article [55] présenté à l'annexe C). Guillaume Damiand a ensuite continué à suivre cette voie. Il a étudié le cas 3d en détail (modèle, algorithmes d'extraction et opérateurs) et s'est également intéressé aux pyramide de cartes comme représentation multi-échelle des images (voir la section 3.2.3 pour plus de précisions).

Travaux concomitants

Dans le même temps, d'autres auteurs se sont ainsi intéressés à des représentations permettant de rendre compte de la topologie des images. V.A. Kovalevsky fut le premier en 89 [146, 144] à s'intéresser à la topologie des images dans le cadre de la segmentation. Comme indiqué à la section 2.4.3, ce fut le premier à voir les images comme des complexes cellulaires abstraits. Néanmoins, la structure de données qu'il propose alors, les *cell-list*, n'est pas satisfaisante puisque le dual est équivalent au graphe d'adjacence. Ce n'est qu'en 2001 qu'il propose une amélioration [147], les *block-cell*, qui permet d'atteindre ce but. En 2003 il étend son modèle pour pouvoir représenter des variétés combinatoires de dimensions n [145]. Entre temps nous avons proposé le graphe topologique des frontières. Néanmoins d'autres auteurs ont étudié la question, également du point de vue des cartes combinatoires. On citera en particulier, Achille Braquelaire, Luc Brun et Jean-Philippe Domenger [39, 26, 40] qui comme nous ont utilisé les cartes, d'abord comme un outil pour aider à la segmentation, puis qui les ont étudiées d'un point de vue modèle de représentation. Walter Kropatsch [151, 152, 150] quant à lui propose, dans le cadre des représentations pyramidales, une structure de graphes duaux permettant également de rendre compte de la topologie. Néanmoins, cette structure est plus difficile à manipuler puisqu'elle oblige à maintenir deux structures, le graphe d'adjacence et son dual. Jacques-Olivier Lachaud s'est également intéressé à la question de la représentation des complexes cellulaires et a proposé en 2003 un codage permettant d'implémenter efficacement des algorithmes [155]. Ce codage, bien que très efficace en terme d'espace ou de complexité, a l'inconvénient de ne pas être symbolique. De plus, contrairement aux cartes combinatoires qui peuvent être vues comme une représentation algébrique sur laquelle il est donc facile de concevoir et prouver des algorithmes, on a affaire ici à un codage qui nécessite donc d'écrire des algorithmes guidés par la structure de données et pour lesquels il n'est pas facile d'en donner une interprétation intuitive. En fait, comme le montre la section 3.2.3 qui suit, c'est essentiellement dans la lignée du T.G.F [81, 82, 3], de la carte des bords [24] et des cartes topologiques [21, 22, 55] que se situent les travaux qui ont suivi.

3.2.3 Quelques perspectives de l'utilisation des cartes en imagerie

Dans un premier temps, j'ai proposé à l'équipe de chimie-informatique d'utiliser les cartes combinatoires pour représenter des molécules. En effet, une de leurs préoccupations était de représenter l'ordre des liaisons atomiques autour d'un atome. Les cartes combinatoires de par la permutation β_1 semblaient donc prometteuses. Cette idée a permis de proposer une nouvelle modélisation [69, 70] de la structure des molécules chimiques. De même, en collaboration avec le CEMAGREF, nous avons étudié l'utilisation du T.G.F dans le cadre d'une segmentation d'images, sa structure permettant de naviguer facilement de région en région tout en suivant le contour. Cela nous a permis de proposer des algorithmes pour la reconnaissance de feuilles dans des scènes naturelles [64, 63].

Mais les travaux les plus significatifs, et ceux mettant le plus en valeur le potentiel de notre proposition initiale, sont ceux de Guillaume Damiand, qui après avoir étudié le cas de la dimension 3 [54], s'est attaché à les utiliser dans le cadre d'une segmentation d'images [57]. Il a d'autre part étendu ces résultats aux représentations pyramidales en étudiant en détails les opérateurs topologiques sur ces modélisations multi-échelles [56, 112, 111, 210].

3.3 Conclusion

Mon idée a toujours été de m'appuyer sur un modèle cohérent et une modélisation par des structures combinatoires afin de proposer des algorithmes d'analyse plus performants car tirant parti à la fois du modèle et s'appuyant sur les représentations proposées. C'est le sens des idées et travaux présentés dans ce chapitre, et tout en cherchant à améliorer les résultats existants, voir par exemple les travaux que nous avons entrepris avec A. Mas et que j'ai introduit à la section 3.1.2, je m'intéresse désormais à l'utilisation de ces modélisations pour améliorer une segmentation initiale. Ainsi, initiée par le co-encadrement en thèse avec Gilles Rabatel de Benoît de Mezzo [63], j'ai entrepris une collaboration avec le CEMAGREF sur des problématiques d'analyse d'images de scènes naturelles. Après avoir proposé d'utiliser la structure du graphe topologique d'adjacence pour faciliter la reconnaissance de feuilles [64] et proposé une méthode de segmentation par collaboration régions-contours [107], nous nous intéressons actuellement avec Nathalie Gorretta et Gilles Rabatel à l'analyse hyperspectrale d'images : l'idée est de combiner des critères spatiaux et topologiques avec des critères chimio-métriques issus de l'analyse hyperspectrale. Pour cela nous utilisons la structure Union-Find afin de faciliter la segmentation en régions en liaison avec les caractéristiques spectrales de chaque pixel. Après un premier résultat concernant une application dans le cas d'images multi-spectrales [108], nous avons proposé une nouvelle stratégie de segmentation d'images hyperspectrales utilisant la topologie des régions pour améliorer le résultat de l'analyse [110, 109].

On notera aussi les travaux de R. Nock qui propose une approche semi-supervisée pour une segmentation en régions d'images couleurs [187]. Je pense pour ma part comme beaucoup d'autres auteurs (voir par exemple [100, 208, 129, 179, 216]) que l'on devrait pouvoir améliorer nettement les résultats des segmentations par une approche collaborative régions-contours. Cela suppose des structures de données bien adaptées si l'on veut conserver une certaine efficacité. La structure d'Union-Find que nous utilisons dans nos algorithmes (voir la section 3.1.1) et les cartes combinatoires (voir la section 3.2) ont tout le potentiel pour cela. Je pense qu'en utilisant des méthodes algorithmiques travaillant sur la structure de graphes afin d'analyser plus globalement l'image, tel ce que propose par exemple J. Shi et J. Malik [209], et que nous utilisons dans [109], on devrait pouvoir proposer de nouvelles méthodes d'amélioration de segmentation. Un autre problème réside dans le fait que les critères d'homogénéité sont souvent manichéens, or il y a de nombreux cas où le critère d'homogénéité devrait nous placer dans une zone d'incertitude et ne pas pouvoir décider clairement. C'est pourquoi, notre proposition avec A. Mas est particulièrement intéressante. En effet, nous sommes à même de détecter que nous sommes justement dans ce cas où le choix de fusionner ou pas ne peut pas être décidé simplement. Nous proposons bien un moyen de résoudre ce dilemme et de trancher grâce à un autre critère. Néanmoins, dans une stratégie plus globale on peut imaginer se servir d'informations de régions dans un voisinage proche, ou même de remettre en question certaines décisions incertaines par une approche multi-résolution, telle que le permettent les pyramides de cartes proposées notamment dans les travaux de G. Damiand et ses co-auteurs [56, 112, 111, 210].

Notre approche combinatoire de l'imagerie, les propositions à base d'Union-Find, de cartes combinatoires, de critères de segmentation basés sur les inégalités de concentration ont ouvert la voie à un certain nombre de travaux par d'autres auteurs et me permettent de me rapprocher de mon but initial : proposer des outils d'analyse s'appuyant à la fois sur une modélisation mathématique formelle, des structures de données expressives, et des algorithmes performants, afin d'obtenir des résultats pertinents.

Chapitre 4

Modélisation géométrique discrète

Préambule

La géométrie discrète, plus qu'un sujet de recherche, est le thème général d'une grande partie de mes travaux. En fait, j'ai toujours mené de front deux aspects de l'imagerie, l'analyse d'images et la géométrie discrète. Le premier thème a été développé à la section 3.1. La géométrie discrète recouvre l'autre partie de mes recherches, celles bien sûr sur la topologie des espaces discrets [1, 81] présentées au chapitre 2.2, celles aussi sur la modélisation à base topologique [82, 69, 24, 3, 21, 22, 55] présentées à la section 3.2, mais aussi d'autres travaux comme ceux sur les polyominos [58, 59, 83], les plans [18, 19] ou les cercles [87, 91] et courbes discrètes [92]. C'est cet aspect, lié à la modélisation d'objets et structures discrètes que je vais aborder dans ce chapitre. Si la géométrie discrète a donc toujours été présente en tant que domaine de recherche, je n'ai pas continuellement mené des travaux en modélisation géométrique. En fait ma première approche de la géométrie discrète l'a été par la topologie. Cela m'a ensuite mené à l'étude des polyominos [58, 59] et de la tomographie discrète [83]. Je n'ai pas creusé cette piste de recherche préférant alors me concentrer sur la modélisation à base topologique. Néanmoins, dans la perspective de modéliser au mieux les images et notamment les images segmentées, je savais que la problématique de la modélisation des objets discrets et notamment des contours reviendrait au cœur de mes préoccupations. Ainsi dernièrement, avec D. Jamet, étudiant alors en thèse sous la direction de V. Berthé, nous nous sommes intéressés à la structure des plans discrets (voir la section 4.2 et l'annexe D). Plus récemment, avec J.-L. Toutant que j'ai encadré en thèse et qui est maintenant Maître de Conférence à Clermont-Ferrand, nous nous sommes intéressés à la problématique de la modélisation des objets discrets non linéaires et en particulier des cercles et courbes discrets (voir la section 4.3). Ce chapitre présente donc très brièvement cet aspect de mes recherches, pour plus de détails, le lecteur pourra se référer à nos dernières publications sur le sujet [18, 87, 91, 92, 19].

4.1 Introduction

Depuis l'apparition des premiers traceurs [29], puis ensuite de celle des écrans graphiques d'ordinateur, se pose le problème de représenter des objets géométriques dans un espace discret. Un moyen classique d'obtenir des objets géométriques à l'écran est d'utiliser leur forme analytique, définie dans le domaine continu, et de numériser les valeurs réelles obtenues. De nombreux travaux se sont intéressés à la problématique de la discrétisation des objets géométriques euclidiens. Cette approche majoritaire peut se résumer en : « *un objet discret est le résultat d'un processus d'approximation local appliqué à la représentation continue de l'objet euclidien* ». C'est ce processus que l'on appelle la *discrétisation*. Cette approche a l'avantage d'être simple et a été appliquée

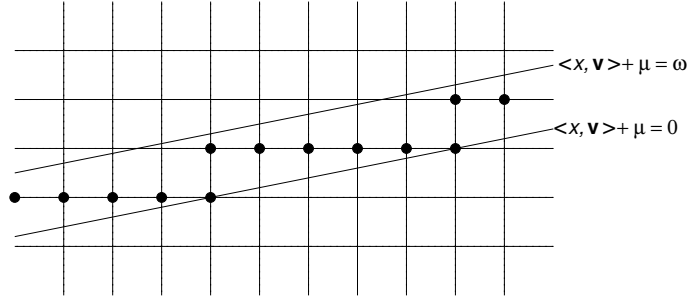


FIG. 4.1 – Exemple de droite arithmétique.

avec succès à différents types d'objets. Néanmoins elle présente aussi des désavantages, comme notamment de ne pas pouvoir déterminer facilement les propriétés intrinsèques des objets ainsi obtenus. Une autre approche a été proposée par J.-P. Réveilles [195]. Elle consiste à définir analytiquement les objets discrets comme solution de systèmes d'inéquations diophantiennes. Ainsi l'ensemble des points entiers obtenus forme bien un ensemble discret tout en pouvant être étudié analytiquement. C'est cette approche que j'ai adoptée. Elle correspond à mon approche des problèmes : un modèle mathématique définissant l'objet et des algorithmes combinatoires pour les manipuler.

Afin de présenter le contexte et les notations, je rappelle ici dans cette introduction, les notions de base de la géométrie discrète arithmétique en présentant les définitions et théorèmes liés à la droite discrète. Une droite arithmétique est un ensemble de points à coordonnées entières et satisfaisant une double inégalité diophantienne de degré 1.

Définition 4.1 (Droite arithmétique [195]). Soit $\mathbf{v} = v_1\mathbf{e}_1 + v_2\mathbf{e}_2 \in \mathbb{R}^2$, où $\{\mathbf{e}_1, \mathbf{e}_2\}$ désigne la base canonique du $\mathbb{R}$ -espace vectoriel $\mathbb{R}^2$. Soit $\mu \in \mathbb{R}$ et $w \in \mathbb{R}_+$. La *droite arithmétique* $\mathfrak{D}(\mathbf{v}, \mu, w)$ de *vecteur normal* $\mathbf{v}$, de *décalage* μ et d'*épaisseur arithmétique* w est le sous-ensemble de $\mathbb{Z}^2$ (voir figure 4.1) définie par :

$$\mathfrak{D}(\mathbf{v}, \mu, w) = \{\mathbf{x} \in \mathbb{Z}^2, 0 \leq \langle \mathbf{x}, \mathbf{v} \rangle + \mu < w\},$$

où $\langle \mathbf{x}, \mathbf{v} \rangle = x_1v_1 + x_2v_2$ désigne le produit scalaire usuel sur $\mathbb{R}^2$. Si $w = \|\mathbf{v}\|_\infty = \max\{|v_1|, |v_2|\}$ (resp. $w = \|\mathbf{v}\|_1 = |v_1| + |v_2|$), alors la droite arithmétique $\mathfrak{D}(\mathbf{v}, \mu, w)$ est dite *naïve* (resp. *standard*) (voir Fig. 4.2).

Géométriquement, la droite arithmétique de vecteur normal $\mathbf{v}$, de décalage μ et d'*épaisseur arithmétique* w , est l'ensemble des points de $\mathbb{Z}^2$ compris entre la droite d'équation $\langle \mathbf{x}, \mathbf{v} \rangle + \mu = 0$ et celle d'équation $\langle \mathbf{x}, \mathbf{v} \rangle + \mu = w$ (voir Figure 4.1).

D'un point de vue applicatif (représentation, reconnaissance ...), les droites arithmétiques à paramètres rationnels sont suffisantes dès lors que l'on travaille sur un espace fini, on supposera donc souvent que si $\mathfrak{D}(\mathbf{v}, \mu, w)$ est une droite arithmétique et si

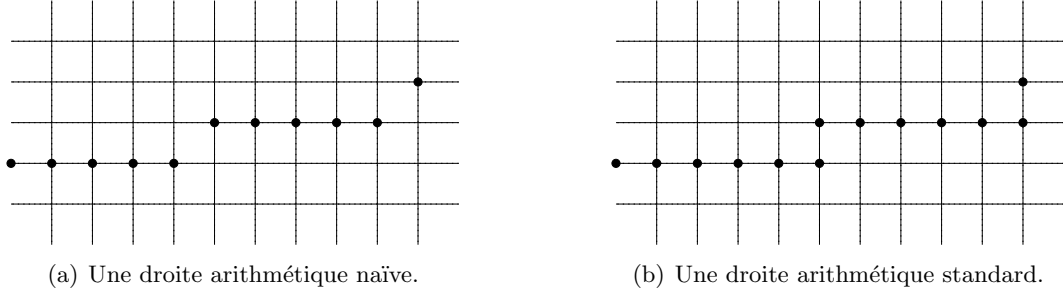


FIG. 4.2 – Une droite arithmétique naïve et une droite arithmétique standard.

$\dim_{\mathbb{Q}}\{v_1, v_2\} = 1$, alors $\mathbf{v} \in \mathbb{Z}^2$, $\mu \in \mathbb{Z}$, $w \in \mathbb{N}$ et $\gcd\{v_1, v_2\} = 1$.

Les droites arithmétiques naïves jouent un rôle très important dans la connexité des droites arithmétiques :

Théorème 4.1 ([195]). *Les droites naïves sont les droites les plus fines à être connexes. Autrement dit, pour tout $\mathbf{v} \in \mathbb{R}^2$, pour tout $\mu \in \mathbb{R}$ et tout $w \in \mathbb{R}_+$:*

$$\mathfrak{D}(\mathbf{v}, \mu, w) \text{ est connexe} \iff w \geq \|\mathbf{v}\|_{\infty}.$$

Pour prouver ce théorème, on peut facilement montrer qu'il existe une bijection entre les points d'une droite arithmétique et les points de $\mathbb{Z}$. Géométriquement, cette propriété signifie que pour tout $k \in \mathbb{Z}$, la droite arithmétique naïve $\mathfrak{D}(\mathbf{v}, \mu, w)$, où $0 \leq v_1 \leq v_2$, possède exactement un et un seul point d'abscisse k . On dit que les droites arithmétiques naïves sont *fonctionnelles*.

Le théorème 4.2 qui reprend les propriétés du théorème 4.1 a été prouvé initialement par Jean-Pierre Reveillès [195] et sert à caractériser la topologie des droites arithmétiques (voir exemple Figure 4.3).

Théorème 4.2 ([195]). *Soit $\mathfrak{D} = \mathfrak{D}(\mathbf{v}, \mu, \omega)$ une droite discrète.*

- i) *Si $\omega = \|\mathbf{v}\|_{\infty}$, alors la droite discrète $\mathfrak{D}(\mathbf{v}, \mu, \omega)$ est 0-connexe ; elle est dit naïve.*
- ii) *Si $\omega < \|\mathbf{v}\|_{\infty}$, alors la droite discrète $\mathfrak{D}(\mathbf{v}, \mu, \omega)$ est déconnectée.*
- iii) *Si $\omega = \|\mathbf{v}\|_1$, alors la droite discrète $\mathfrak{D}(\mathbf{v}, \mu, \omega)$ est 1-connexe ; elle est dit standard.*
- iv) *Si $\|\mathbf{v}\|_{\infty} < \omega < \|\mathbf{v}\|_1$, alors la droite discrète $\mathfrak{D}(\mathbf{v}, \mu, \omega)$ est $\star$ -connexe : elle possède des 1-connexités et des 0-connexités.*

La définition de droite arithmétique introduite par J.-P. Reveillès [195] (voir définition 4.1) s'étend naturellement aux dimensions supérieures $n \geq 2$ et à la section 4.2, nous nous intéressons donc aux plans arithmétiques et plus particulièrement à la généralisation de la propriété de fonctionnalité. D'autre part, ma conviction, que nous

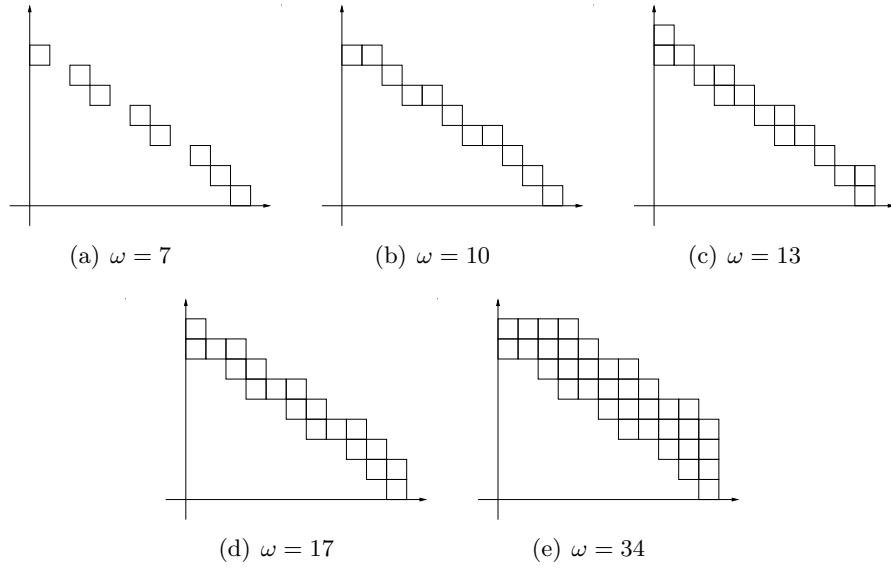


FIG. 4.3 – Les droites arithmétiques de vecteur normal $7\mathbf{e}_1 + 10\mathbf{e}_2$.

avons essayé de démontrer à travers plusieurs résultats [87, 91, 92], est que l'épaisseur constante proposée par J.-P. Réveilles pour les droites et plans discrets [195], et reprise par É. Andres pour les cercles discrets [6, 8], n'est pas adaptée à ces objets. Je suis convaincu que l'épaisseur arithmétique doit dépendre de la primitive représentée et peut donc être variable. C'est le cas en particulier des objets non linéaires que j'aborderai à la section 4.3.

4.2 Plans discrets : une structure fondamentalement en 2 dimensions

La notion de droite arithmétique s'étend naturellement aux dimensions supérieures. Soit $n \in \mathbb{N}$, un entier supérieur ou égal à 2. Par $\mathbb{R}^n$, nous désignons l'espace euclidien usuel, muni de sa base canonique $\{\mathbf{e}_1, \dots, \mathbf{e}_n\}$. Soit $\mathbf{x} \in \mathbb{R}^n$, nous notons x_i , pour $i \in \{1, \dots, n\}$ la i^{ieme} coordonnée de $\mathbf{x}$ dans la base $\{\mathbf{e}_1, \dots, \mathbf{e}_n\}$: $\mathbf{x} = \sum_{i=1}^n x_i \mathbf{e}_i$.

Il est usuel de représenter un point $\mathbf{x}$ de $\mathbb{Z}^n$ par le *voxel* $\mathbf{x}$, à savoir l'hypercube d'arête 1 centré en $\mathbf{x}$ (voir figure 2.2).

Soit $\langle \cdot, \cdot \rangle : \mathbb{R}^n \times \mathbb{R}^n \longrightarrow \mathbb{R}$, le produit scalaire usuel défini sur $\mathbb{R}^n$: $\forall \mathbf{x} \in \mathbb{R}^n, \forall \mathbf{y} \in \mathbb{R}^n$, $\langle \mathbf{x}, \mathbf{y} \rangle = \sum_{i=1}^n x_i y_i$, on définit un hyperplan discret arithmétique de la manière suivante :

Définition 4.2 (Hyperplan arithmétique). Soient $\mathbf{v} \in \mathbb{R}^n$, $\mu \in \mathbb{R}$ et $w \in \mathbb{R}$. L'*hyperplan arithmétique* $\mathfrak{P}(\mathbf{v}, \mu, w)$ de vecteur normal $\mathbf{v}$, de décalage μ et d'épaisseur w est le sous-

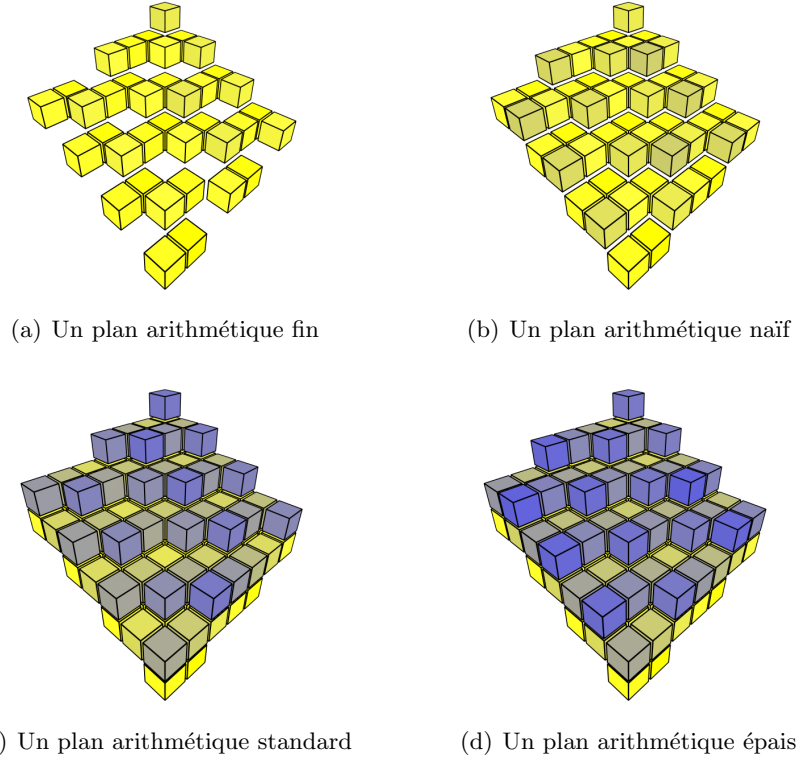


FIG. 4.4 – Différents morceaux finis de plans arithmétiques.

ensemble de $\mathbb{Z}^n$ défini par :

$$\mathfrak{P}(\mathbf{v}, \mu, w) = \{\mathbf{x} \in \mathbb{Z}^n, 0 \leq \langle \mathbf{x}, \mathbf{v} \rangle + \mu < w\}.$$

Si $w = \|\mathbf{v}\|_\infty = \max\{|v_1|, \dots, |v_n|\}$ (resp. $w = \|\mathbf{v}\|_1 = |v_1| + \dots + |v_n|$), alors l'hyperplan arithmétique $\mathfrak{P}(\mathbf{v}, \mu, w)$ est dit *naïf* (resp. *standard*).

Si $w < \|\mathbf{v}\|_\infty$ (resp. $w \geq \|\mathbf{v}\|_1$), on dit que l'hyperplan arithmétique $\mathfrak{P}(\mathbf{v}, \mu, w)$ est *fin* (resp. *épais*) (voir figure 4.4). Si $n = 3$, alors l'ensemble $\mathfrak{P}(\mathbf{v}, \mu, w)$ sera appelé le *plan arithmétique* de vecteur normal $\mathbf{v} \in \mathbb{R}^2$, de décalage μ et d'épaisseur w .

Définition 4.3 (Hyperplan arithmétique rationnel, irrationnel). Soit $\mathfrak{P}(\mathbf{v}, \mu, \omega)$ un hyperplan arithmétique où $\mathbf{v} \in \mathbb{R}^n$, $\mu \in \mathbb{R}$ et $\omega \in \mathbb{R}_+$. Si $\dim_{\mathbb{Q}}\{v_1, \dots, v_n\} = 1$ (resp. $\dim_{\mathbb{Q}}\{v_1, \dots, v_n\} \neq 1$), on dit que le vecteur $\mathbf{v}$ est *rationnel* (resp. *irrationnel*). Nous étendons cette terminologie à l'hyperplan arithmétique $\mathfrak{P}(\mathbf{v}, \mu, \omega)$ de la manière suivante : l'hyperplan arithmétique $\mathfrak{P}(\mathbf{v}, \mu, \omega)$ est dit *rationnel* (resp. *irrationnel*) si son vecteur normal $\mathbf{v}$ est rationnel (resp. irrationnel).

On peut facilement montrer que tout hyperplan arithmétique rationnel possède une représentation en *tout-entier*, c'est-à-dire de vecteur normal, de décalage et d'épaisseur entiers. Cette propriété est bien entendu intéressante dès lors qu'il s'agit d'implémenter des algorithmes traitant des hyperplans arithmétiques. Les hyperplans arithmétiques rationnels, naïfs ou standards, ont été largement étudiés ces dernières années (voir par exemple [195, 80, 66, 9, 227, 228, 103, 36, 37]). Plusieurs raisons semblent justifier ce choix. Tout d'abord, nombre de ces travaux proviennent d'une motivation pratique [66, 211, 212] et la nécessité d'implémenter les algorithmes ainsi obtenus conduit naturellement à considérer uniquement les hyperplans arithmétiques à paramètres entiers. Les hyperplans arithmétiques irrationnels permettent, quant à eux, une compréhension profonde de la structure topologique, arithmétique et dynamique des hyperplans arithmétiques quelconques. Leur étude autorise en effet l'utilisation de nombreux outils classiques, hérités de l'analyse réelle, de la théorie ergodique ou encore des systèmes dynamiques symboliques, possédant chacun un analogue discret aux propriétés structurales très proches. C'est pourquoi nous ne nous restreignons pas au cas rationnel bien qu'en pratique et dans la littérature ce soit généralement le cas.

Il est connu qu'à la manière des droites arithmétiques, les plans arithmétiques naïfs sont fonctionnels, c'est à dire en correspondance bijective avec les points à coordonnées entières d'un des plans projectifs. Mon intuition était qu'un plan avait fondamentalement une structure $2D$ et ce quelle que soit son épaisseur. Finalement un plan épais ne serait en quelque sorte qu'un plan fin replié sur lui-même, « froissé », afin de créer l'épaisseur. Le but de notre étude [19] était donc d'étendre la notion de fonctionnalité des plans arithmétiques discrets à tous les plans. Pour cela, au lieu de projeter orthogonalement sur un des plans projectifs, nous avons introduit une autre projection sur un plan le long d'une direction $\alpha = (\alpha_1, \alpha_2, \alpha_3)$, en quelque sorte duale du vecteur normal du plan discret $\mathfrak{P}(\mathbf{v}, \mu, \omega)$, c'est-à-dire $\alpha_1 v_1 + \alpha_2 v_2 + \alpha_3 v_3 = \omega$ telle que la projection de $\mathbb{Z}^3$ et des points du plan discret $\mathfrak{P}(\mathbf{v}, \mu, \omega)$ soient en bijection. Cette propriété que nous avons appelée la *fonctionnalité généralisée* est donnée par le théorème suivant :

Théorème 4.1 (Théorème 1 de [19] (fonctionnalité généralisée)). *Soit $\mathfrak{P}(\mathbf{v}, \mu, \omega)$ un plan arithmétique discret, et soit $\alpha \in \mathbb{Z}^3$ tel que $\gcd\{\alpha_1, \alpha_2, \alpha_3\} = 1$, notons $\Pi_\alpha^\perp : \mathbb{R}^3 \rightarrow \{\vec{x} \in \mathbb{R}^3, \langle \alpha, \vec{x} \rangle = 0\}$ la projection affine orthogonale sur le plan $\{\vec{x} \in \mathbb{R}^3, \langle \alpha, \vec{x} \rangle = 0\}$ le long de la direction α , alors l'application $\Pi_\alpha^\perp : \mathfrak{P}(\mathbf{v}, \mu, \omega) \rightarrow \Pi_\alpha^\perp(\mathbb{Z}^3)$ est une bijection si et seulement si $|\langle \alpha, \vec{v} \rangle| = \omega$.*

Au signe près, on appelle le vecteur α , le *vecteur fonctionnel* pour le plan. Dans [19] (voir l'annexe D) nous avons montré que tout plan arithmétique rationnel admet un vecteur fonctionnel.

Ce résultat est très important, car conformément à notre intuition à l'origine du théorème 4.1, cette propriété nous permet de ramener un problème de la dimension 3 à

la dimension 2 et donc de mieux comprendre la structure combinatoire et géométrique des plans arithmétiques. Cela nous permet également d'espérer pouvoir étendre la majorité des résultats sur les plans naïfs aux plans épais. Ainsi dans [19] (voir l'annexe D), sous l'hypothèse qu'un vecteur fonctionnel α possédant une coordonnée égale à 1 existe, nous donnons plusieurs applications de la fonctionnalité généralisée. Ainsi nous avons pu énumérer certaines configurations locales et généraliser les résultats sur les (m, n) -cubes [227, 102, 228, 125]. Nous avons également pu étendre à tous les plans rationnels $\mathfrak{P}(\mathbf{v}, \mu, \omega)$, avec $\omega - v_3 \in \gcd(v_1, v_2)\mathbb{Z}$, l'algorithme de I. Debled-Rennesson [65] de tracé de plan arithmétique naïf. Enfin nous avons montré que l'on pouvait exhiber de tout plan arithmétique discret $\mathfrak{P}(\mathbf{v}, \mu, \omega)$ une structure de variété combinatoire de dimension 2, sous l'hypothèse $\omega \in v_1\mathbb{Z} + v_2\mathbb{Z} + v_3$ et $\omega \leq \max\{|v_1|, |v_2|\}$.

On trouvera tous les détails de cette étude dans l'article [19] inclus à l'annexe D. Si ce résultat est essentiellement théorique, il est à la base d'une meilleure compréhension de la structure des plans discrets et a ainsi permis de généraliser des résultats autrefois limités uniquement aux plans naïfs. Mon but dans la compréhension des objets discrets est d'arriver à terme à pouvoir compacter le codage des contours des objets segmentés dans les images 2D ou 3D. C'est pourquoi je me suis intéressé également à la structure des objets non linéaires.

4.3 Objets non linéaires : objets à épaisseur variable

Ces travaux sont principalement ceux de J.-L. Toutant qu'il a effectués pendant sa thèse sous ma direction. Je n'en donnerai donc que les idées directrices et les principaux résultats. Le lecteur intéressé pourra se reporter aux différentes publications que nous avons eues sur le sujet [87, 91, 92] et à sa thèse [223].

Très rapidement l'idée nous est venue que pour les cercles discrets en particulier, et en général pour les objets non linéaires, une épaisseur constante n'était pas adaptée. En effet, si l'on regarde les différents travaux sur les cercles discrets, on peut noter que contrairement aux droites et plans discrets arithmétiques, aucune définition ne s'est vraiment imposée comme un standard. L'algorithme 2 de J. Bresenham [30] reste la première définition historique des cercles discrets. Un cercle $\mathcal{C}(r)$ peut être vu comme une courbe implicite, et, en particulier, comme l'ensemble des racines du polynôme $c_r(\mathbf{x}) = x_1^2 + x_2^2 - r^2$. Comme pour les droites discrètes, l'idée est de prendre le pixel dont le reste est minimum, intuitivement ce devrait être le pixel le plus proche du cercle euclidien. B. W. Jordan, W. J. Lennon et B. D. Holm [127] sont les premiers à avoir proposé un tel algorithme. Cette approche est aussi celle de J. Bresenham [30] qui reste un des algorithmes de référence. Des optimisations de ces algorithmes ont ensuite été proposés, notamment par M. D. McIlroy [171] et Y. P. Kuzmin [154]. L'algorithme 2 de

J. Bresenham est un algorithme de tracé incrémental : à chaque étape il s'agit, étant donné le pixel $\mathbf{p} = (i, j)$ appartenant déjà au cercle, de sélectionner $\mathbf{p}_1 = (i, j + 1)$ ou $\mathbf{p}_2 = (i - 1, j + 1)$ en fonction de leur reste $|c_{(r)}(\mathbf{p}_1)|$ et $|c_{(r)}(\mathbf{p}_2)|$, ce qui revient à vérifier la condition $c_{(r)}(i, j + 1) > i$.

Algorithm 2: Tracé de cercle de Bresenham

Input: $r \in \mathbb{N}$.
Output: $\mathbb{C}(r)$.
 $i \leftarrow r$;
 $j \leftarrow 0$;
 $\Delta \leftarrow 0$;
while $(i \geq j)$ **do**
 AjouterPixel(i, j);
 $\Delta \leftarrow \Delta + 2j + 1$;
 $j \leftarrow j + 1$;
 if $(\Delta > i)$ **then**
 $\Delta \leftarrow \Delta - 2i + 1$;
 $i \leftarrow i - 1$;
end while

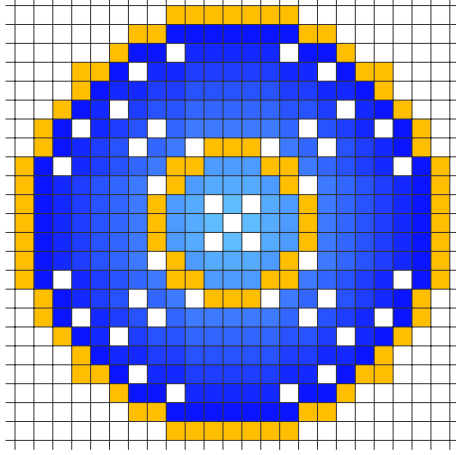
Z. Kulpa a montré dans [153] que ces cercles discrets présentaient deux défauts :

1. bien qu'étant des cercles fins, ils ne sont pas des courbes fermées simples ;
2. de tels cercles concentriques de rayons successifs ne pavent pas le plan.

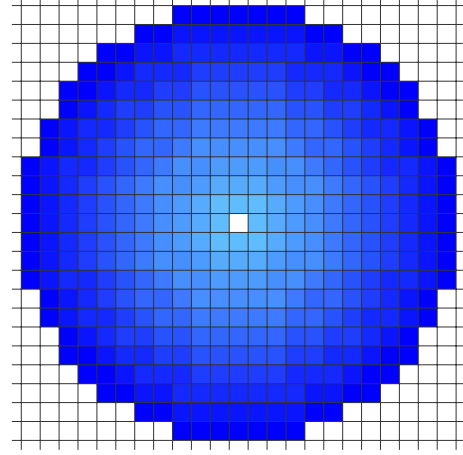
La figure 4.5(a) montre des cercles concentriques de rayons entiers successifs. On peut facilement vérifier qu'ils ne pavent pas le plan. On peut également vérifier, par exemple sur les cercles de rayons 4 et 11 (en jaune sur la figure 4.5(a)), qu'ils ne sont pas tous les plus fins possibles et qu'ils possèdent des points 0-simples sur leurs diagonales. Il a été montré que tous les cercles dont le rayon est solution de l'équation $r^2 = 2i^2 - i + 1$ présenteront cette erreur.

Si l'on veut que les cercles pavent le plan, ce qui est indispensable pour certaines applications, alors il faut considérer des anneaux discrets fins [153] plutôt que des approximations de cercles. On considère ici qu'un cercle discret est un anneau discret le plus fin possible, c'est-à-dire défini par deux cercles concentriques de rayons $r - \frac{1}{2}$ et $r + \frac{1}{2}$. On obtient alors des cercles au moins 0-connexes et au plus 1-connexe qui pavent le plan (voir figure 4.5(b)). L'algorithme 3 permet de tracer de tels cercles restreints au cas entier.

Ces premiers cercles sont en fait des algorithmes de tracé de cercles (pour une étude plus complète des différents algorithmes de tracé de cercles discrets, le lecteur intéressé pourra se reporter par exemple à la thèse de J.-L. Toutant [223]) et ne sont donc pas définis à partir de propriétés globales et vérifiables. En fait le seul moyen de vérifier certaines propriétés ou de calculer certaines opérations géométriques, comme l'intersection



(a) Cercles discrets dits de Bresenham (en clair - jaune - les cercles de rayons 4 et 11)



(b) Cercles discrets (anneaux) pavant le plan

de deux cercles, est de tracer tout le cercle afin de déterminer l'ensemble discret ainsi représenté. Éric Andres [6], à la suite des travaux de J.-P. Reveillès [195], a remédié à cela en proposant une définition de *cercle discret analytique* :

Définition 4.4 (cercle discret analytique [6]). Soient $\mathbf{o} \in \mathbb{R}^2$ et $r, \omega \in \mathbb{R}_+^*$. Alors, le *cercle discret analytique* de centre $\mathbf{o}$, de rayon r et d'épaisseur ω , est une couronne discrète dont l'ensemble des pixels $\mathbf{p}$ vérifient :

$$\left(r - \frac{\omega}{2}\right)^2 \leq (p_1 - o_1)^2 + (p_2 - o_1)^2 < \left(r + \frac{\omega}{2}\right)^2.$$

En tant que couronnes discrètes, ces cercles pavent le plan. De plus ils contiennent une courbe discrète fermée 0-connexe et sont inclus dans une courbe discrète fermée 1-connexe. Parmi eux, on distingue les *cercles discrets réguliers* qui sont les cercles discrets connexes les plus fins.

Définition 4.5 (cercles discrets réguliers [7]). Un cercle discret analytique est dit *régulier* si son épaisseur ω est égale à 1.

Non seulement on dispose d'une caractérisation analytique de cercles discrets, mais en plus cette définition s'étend sans difficulté aux dimensions supérieures contrairement aux définitions algorithmes données précédemment. Ainsi on peut définir une hypersphère discrète comme :

Définition 4.6 (hypersphère discrète analytique [10]). Soit d la dimension de l'espace considéré et $r \in \mathbb{R}_+^*$, $\mathbf{o} = (o_1, \dots, o_d) \in \mathbb{R}^d$ et $\omega \in \mathbb{R}_+^*$, l'*hypersphère discrète arithmétique*

Algorithm 3: Tracé d'un cercle discret pavant (anneau)

Input: $r \in \mathbb{N}$
Output: $\mathbb{C}(r)$
 $i \leftarrow r$;
 $j \leftarrow 0$;
 $\Delta \leftarrow 0$;
while $i \geq j$ **do**
 AjouterPixel(i, j);
 $\Delta \leftarrow \Delta + 2j + 1$;
 $j \leftarrow j + 1$;
 if $\Delta > r$ **then**
 $\Delta \leftarrow \Delta - 2i + 1$;
 $i \leftarrow i - 1$;
 else if $(\Delta > -r + 2i - 1)$ **et** $(i \geq j)$ **then**
 AjouterPixel(i, j);
 $\Delta \leftarrow \Delta - 2i + 1$;
 $i \leftarrow i - 1$;

$\mathbb{S}(\mathbf{o}, r, \omega)$ de centre $\mathbf{o}$, de rayon r et d'épaisseur arithmétique ω , est le sous-ensemble de $\mathbb{Z}^d$ défini par :

$$\mathbb{S}(\mathbf{o}, r, \omega) = \left\{ \mathbf{v} \in \mathbb{Z}^d \mid \left(r - \frac{\omega}{2} \right)^2 \leq \sum_{i=1}^d (v_i - o_i)^2 < \left(r + \frac{\omega}{2} \right)^2 \right\}. \quad (4.1)$$

Cependant, une telle définition ne permet pas de caractériser topologiquement ces cercles, en particulier on ne peut pas caractériser ceux qui sont séparants minimaux. C'est pourquoi nous avons proposé dans [87] une nouvelle définition de *cercle discret arithmétique*. L'originalité de notre approche réside dans le choix d'une épaisseur variable. En fait l'épaisseur devient une *fonction épaisseur* : $\omega : \mathbb{R}^2 \rightarrow \mathbb{R}_+^*$.

Définition 4.7 (cercle discret arithmétique [87]). Soit $\mathbf{o} = (o_1, o_2) \in \mathbb{R}^2$, $r \in \mathbb{R}_+$, soit $\omega : \mathbb{R}^2 \rightarrow \mathbb{R}$ une fonction, le *cercle discret arithmétique* $\mathbb{C}(\mathbf{o}, r, \omega)$ de centre $\mathbf{o}$, de rayon r et de fonction épaisseur ω est l'ensemble des points discrets défini par :

$$\mathbb{C}(\mathbf{o}, r, \omega_{\mathbf{o}}) = \left\{ \mathbf{v} \in \mathbb{Z}^2 \mid -\frac{\omega(\mathbf{v})}{2} \leq (v_1 - o_1)^2 + (v_2 - o_2)^2 - r^2 < \frac{\omega(\mathbf{v})}{2} \right\}. \quad (4.2)$$

Nous avons également mis en évidence qu'il est important de séparer l'expression analytique de la courbe (ici les cercles), de celle de l'épaisseur qui est en fait l'expression de la discrétisation que l'on choisit. Cette définition très générale nous permet, en fonction du choix de ω , de définir différentes classes de cercles et surtout nous permet de déterminer et prouver certaines propriétés topologiques comme la séparabilité.

Ainsi, nous avons d'abord étendu ce résultat en dimension quelconque afin d'obtenir la définition d'une *hypersphère arithmétique* :

Définition 4.8 (hypersphère discrète arithmétique [91]). Soit d la dimension de l'espace, soient $r \in \mathbb{R}_+^*$ et $\mathbf{o} = (o_1, \dots, o_d) \in \mathbb{R}^d$, soient $\omega_1 : \mathbb{R}^d \longrightarrow \mathbb{R}_-$ et $\omega_2 : \mathbb{R}^d \longrightarrow \mathbb{R}_+$ des fonctions, l'*hypersphère discrète arithmétique* $S(\mathbf{o}, r, (\omega_1, \omega_2))$ de *centre* $\mathbf{o}$, de *rayon* r et de fonctions épaisseurs ω_1 et ω_2 est l'ensemble des points discrets définis par :

$$S(\mathbf{o}, r, (\omega_1, \omega_2)) = \left\{ \mathbf{v} \in \mathbb{Z}^d \mid \omega_1(\mathbf{v}) \leq \sum_{i=1}^d (v_i - o_i)^2 - r^2 < \omega_2(\mathbf{v}) \right\}. \quad (4.3)$$

Il est intéressant de noter que cette définition englobe toutes les définitions existantes de cercle, il suffit pour cela de choisir correctement les fonctions ω_1 et ω_2 . Ainsi par exemple, l'*hypersphère discrète analytique* de É. Andres et M.A. Jacob [10] $S(\mathbf{o}, r, \omega)$ est l'*hypersphère discrète arithmétique* $S(\mathbf{o}, r, (\omega_1, \omega_2))$ telle que

$$\begin{array}{ll} \omega_1 : \mathbb{R}^d \longrightarrow \mathbb{R}_- & \omega_2 : \mathbb{R}^d \longrightarrow \mathbb{R}_+ \\ \mathbf{x} \longmapsto -r\omega + \frac{\omega^2}{4} & \mathbf{x} \longmapsto r\omega + \frac{\omega^2}{4} \end{array}$$

Le fait de séparer l'expression analytique de la courbe de celle de l'épaisseur est important car cela nous permet de lier l'épaisseur, non seulement à la forme de la courbe, mais aussi à la topologie de l'espace considéré. Ainsi nous proposons d'exprimer l'épaisseur en un point, comme la norme d'un vecteur exprimant la variation locale des points de la courbe. Pour cela, nous avons défini dans [91] une *norme de minimalité* (voir définition 4.9) et une fonction de *variation discrète locale* (voir la définition 4.10).

La norme de k -minimalité $] \cdot [_k$ (pour une preuve que $] \cdot [_k$ est bien une norme, voir la proposition 2 de [91]) est donnée par la définition 4.9 suivante :

Définition 4.9 (norme de k minimalité). Soit d la dimension de l'espace considéré et soit $k \in \mathbb{N}$ tel que $k < d$, on appelle *k -minimalité norme*, la norme $] \cdot [_k$ définie par :

$$\begin{array}{ll}] \cdot [_k : \mathbb{R}^d \longrightarrow \mathbb{R} \\ \mathbf{x} \longmapsto \sum_{i=d-k+1}^d |x_{\sigma(i)}|, \end{array}$$

avec σ une permutation de $\{1, \dots, d\}$ telle que $\forall i \in \{1, \dots, d-1\}, |x_{\sigma(i)}| \leq |x_{\sigma(i+1)}|$.

Les normes habituellement utilisées en géométrie discrète sont des normes de k -minimalité puisque $\|\mathbf{v}\|_\infty =]\mathbf{v}[_{(d-1)}$ et $\|\mathbf{v}\|_1 =]\mathbf{v}[_0$.

Nous définissons la fonction $\Delta_\lambda f$ de variation discrète locale comme :

Définition 4.10 (variation discrète locale [91]). Soit d la dimension de l'espace, soit $\lambda \in \mathbb{R}$ et $f : \mathbb{R}^d \longrightarrow \mathbb{R}$ une fonction, on note $\Delta_\lambda f$ d'épaisseur normale λ relativement à la fonction f la fonction :

$$\begin{aligned} \Delta_\lambda f : \mathbb{Z}^d &\longrightarrow \mathbb{R}^d \\ \mathbf{v} &\longmapsto \left(f\left(\mathbf{v} + \frac{\partial_1 f(\mathbf{v})}{|\partial_1 f(\mathbf{v})|} \lambda \mathbf{e}_1\right) - f(\mathbf{v}), \dots, f\left(\mathbf{v} + \frac{\partial_d f(\mathbf{v})}{|\partial_d f(\mathbf{v})|} \lambda \mathbf{e}_d\right) - f(\mathbf{v}) \right). \end{aligned} \quad (4.4)$$

et de définir une *hypersphère discrète arithmétique séparante* comme :

Définition 4.11 (hypersphère discrète arithmétique séparante [91]). Soit d la dimension de l'espace $k \in \mathbb{N}$ tel que $k \leq d$, soient $\mathbf{o} = (o_1, \dots, o_d)$ et $r \in \mathbb{R}_+^*$, soient $\lambda_1 \in \mathbb{R}_-$ et $\lambda_2 \in \mathbb{R}_+$, l'*hypersphère discrète arithmétique séparante* $S(\mathbf{o}, r, k, \lambda_1, \lambda_2)$ relativement à la norme de k -minimalité, de centre $\mathbf{o}$, de rayon r , d'*épaisseur normale* $\lambda = \lambda_2 - \lambda_1$ est l'ensemble des points discrets défini par :

$$S(\mathbf{o}, r, k, \lambda_1, \lambda_2) = \left\{ \mathbf{v} \in \mathbb{Z}^d \mid \omega_{(k, \lambda_1)}(\mathbf{v}) \leq s(\mathbf{v}) < \omega_{(k, \lambda_2)}(\mathbf{v}) \right\}. \quad (4.5)$$

avec $\omega_{(k, \lambda_1)}(\mathbf{v}) = -\lfloor \Delta_{\lambda_1} s(\mathbf{v}) \rfloor_k$ et $\omega_{(k, \lambda_2)}(\mathbf{v}) = \lfloor \Delta_{\lambda_2} s(\mathbf{v}) \rfloor_k$.

Dans [91], nous avons montré le théorème 4.2 qui prouve que les hypersphères discrètes arithmétiques sont bien séparantes et le théorème 4.3 prouvant que les hypersphères $S(\mathbf{o}, r, k, 0, 1)$ sont k -minimales. On peut donc à la manière des définitions pour les plans définir les hypersphères discrètes arithmétiques naïves et standards (voir définition 4.12).

Théorème 4.2 (Théorème 3 de [91]). Une hypersphère discrète arithmétique $S(\mathbf{o}, r, k, \lambda_1, \lambda_2)$ d'épaisseur normale $\lambda = \lambda_2 - \lambda_1$, tel que $\lambda \in [1, +\infty[$, est (λ, k) -séparante dans $\mathbb{Z}^d$.

Théorème 4.3 (Théorème 4 de [91]). L'hypersphère discrète arithmétique $S(\mathbf{o}, r, k, 0, 1)$ est k -minimale dans $\mathbb{Z}^d$ pour $k \in \{0, \dots, d-1\}$. Une hypersphère discrète arithmétique $S(\mathbf{o}, r, k, \epsilon-1, \epsilon)$ est aussi k -minimale pour $\epsilon \in [0, 1[$ et $k \in \{1, \dots, d-1\}$.

Définition 4.12 (hypersphères discrètes arithmétiques naïves et standards). Une hypersphère discrète arithmétique *naïve* (respectivement *standard*) est une hypersphère discrète arithmétique $(n-1)$ -minimale (respectivement 0-minimal).

En fait l'utilisation d'une épaisseur variable dépendante de la variabilité de la fonction au point considéré et surtout dépendante d'une norme est une approche générale et peut également s'appliquer aux objets linéaires. Ainsi on peut démontrer facilement

(voir [91]) qu'un plan discret naïf $\mathfrak{P}$ est un plan 0-minimal au sens de $\mathfrak{P}(\mathbf{n}, \mu, \omega_k) = \{\mathbf{v} \in \mathbb{Z}^d \mid 0 \leq p(\mathbf{v}) < \lceil \Delta_1 p(\mathbf{v}) \rceil_k\}$ où $\forall \mathbf{x} \in \mathbb{R}^d, p(\mathbf{x}) = \mathbf{n} \cdot \mathbf{x} + \mu$.

Dans [92], nous avons étendu ce résultat aux courbes polynomiales. Néanmoins des problèmes subsistent notamment dans le positionnement de l'épaisseur par rapport à la courbure. Notre modèle laisse toutes les possibilités mais il semblerait qu'il soit nécessaire de le préciser dans le cas de courbes polynomiales. D'autre part, l'extension aux hypersurfaces plus complexes, comme par exemple celles définies implicitement par un polynôme sans racine double, n'est pas évidente. En effet, si l'on peut appliquer sans difficulté notre définition générale, on n'arrive plus à garantir la minimalité. Or une bonne connaissance de ces objets est nécessaire, d'abord parce que ce sont des objets géométriques courants et ensuite parce que nous en avons besoin pour décrire les frontières des volumes dans les images en dimension 3.

Nous avons donc proposé une approche nouvelle de la notion d'épaisseur. Celle-ci nous semble prometteuse, vus les premiers résultats [87, 91, 92], néanmoins il faut poursuivre l'étude et notamment affiner la notion de courbure afin de mieux définir l'épaisseur. On notera également la notion de k -séparabilité introduite par Jean-Luc dans sa thèse [223] et qui nous est fondamentale dans notre définition de courbe.

4.4 Conclusion et perspectives

Si je me suis intéressé à la géométrie discrète, c'est d'abord par le biais de la topologie des espaces discrets. Néanmoins j'ai toujours suivi de prêt les travaux de modélisation géométrique discrète. En effet, il m'apparaissait évident que cet outil théorique et pratique permettrait de modéliser de manière compacte les contours des régions segmentées dans l'image. Ainsi on peut noter, par exemple, les travaux concernant les chemins en interpixels de A. Vialard et J.-P. Braquelaire [28] sont utilisés par L. Brun [39, 26] pour représenter les contours des régions segmentées, les différents travaux sur la reconnaissance des plans [65, 227, 226, 42, 104, 105, 33, 38, 79] et ceux cherchant à reconnaître des portions de plans discrets [229, 174, 211] afin de polyédriciser des surfaces.

Je pense que l'approche nouvelle que nous avons proposée concernant l'épaisseur variable pour les objets discrets apporte un regard nouveau sur cette épaisseur en lui donnant une justification topologique intéressante. Cette piste mérite d'être creusée, notamment par l'étude plus précise de la notion de courbure afin de pouvoir proposer une définition générale pour les hypersurfaces.

Chapitre 5

Conclusion et perspectives

J'ai essayé de présenter dans ce document un panorama de mes travaux de recherche, en mettant à chaque fois en perspective l'impact qu'ils ont eu sur le domaine, et en donnant mon point de vue sur ce qui me semble être la meilleure approche de la problématique. Je n'hésite pas à regarder les thèmes connexes qui pourraient enrichir la solution au problème courant. C'est pourquoi je ne suis pas resté concentré uniquement sur l'analyse d'images, et que je me suis également intéressé à la topologie des espaces discrets, aux modélisations à base topologique, à la géométrie discrète et enfin aux aspects statistiques du traitement d'images. Cette approche globale de l'imagerie, si elle semble dispersée, est en fait guidée par une ligne directrice unique, *proposer une approche combinatoire de l'analyse d'images*, guidée par une démarche scientifique qui consiste à *élaborer une théorie, définir un cadre formel, proposer un modèle mathématique, définir des algorithmes* s'appuyant sur ce modèle, et si possible les *mettre en œuvre* dans le cadre d'une application. Ainsi partant d'une problématique souvent proche des applications, j'ai souvent fait des allers retours entre la théorie et la pratique. Le titre de ce document résume donc bien ma thématique : « Modélisation, Analyse, Représentation des Images Numériques ».

Au chapitre 2, j'ai développé ce qui reste à la base de mes travaux en imagerie : l'*approche interpixels*. Cette modélisation des espaces images à base de complexes cellulaires est pour moi la plus intéressante. En effet, comme je l'ai démontré à la section 2.4, et en particulier à la section 2.5, cette approche, qui tend d'ailleurs à se généraliser, apporte une notion de dimension des éléments de l'espace permettant une approche plus intuitive et plus proche des notions continues de surfaces et de bord. De plus, il est assez facile de démontrer qu'un tel espace peut être muni de la topologie quotient de $\mathbb{R}^n$ (voir démonstration dans mes publications [1, 81]), ce qui renforce donc le lien entre la topologie de l'espace discret image et celui de l'espace euclidien. Je pense finalement que l'approche quotient associée à celle d'interpixels et de complexe cellulaire permet de définir correctement une topologie adaptée aux espaces des images numériques, tout en permettant de modéliser les différents problèmes liés à l'analyse d'images. Ma conviction est renforcée par le fait que beaucoup de propriétés définies dans différentes propositions se rapprochent ou sont équivalentes à celles que j'avais proposées pour la *topologie-étoile* (voir la section 2.5 pour différents exemples). Une synthèse, à partir d'une topologie quotient basée sur les complexes convexes, semble donc envisageable et permettrait d'unifier tous ces concepts. C'est une des pistes que je compte développer dans un futur proche.

Au chapitre 3, et en particulier à la section 3.1, j'ai présenté mon approche de l'analyse d'images. Celle-ci n'est pas à base de méthodes issues du domaine du traitement du signal (comme traditionnellement dans ce domaine) c'est une approche combinatoire et algorithmique. J'ai bien, dans ma thèse, proposé un algorithme de détection de contours en interpixels [81], mais l'essentiel de mes travaux dans ce domaine est com-

posé d'études algorithmiques avec la proposition d'algorithmes de segmentation basés sur l'*Union-Find*, *Scanline* et *MergeSquares* [84, 85, 86], des modèles statistiques pour la définition d'un *critère d'homogénéité* [88, 90, 89] (voir aussi la section 3.1.2 pour des travaux en cours), ou enfin des mises en œuvre de ces outils dans un domaine applicatif [64, 108, 110, 109]. Bien sûr ces algorithmes restent fidèles à l'approche interpixels et s'appuient sur la topologie-étoile afin de proposer une segmentation cohérente avec les données réelles représentées. À la section 3.2, j'ai rappelé les propositions que j'avais faites concernant le *graphe topologique des frontières* (T.G.F) [82, 3], puis les travaux menés avec G. Damiand sur une représentation à base de cartes combinatoires des images segmentées, la *carte topologique* et la carte des bords [24, 21, 22, 55]. L'idée de cette représentation est de permettre d'exploiter l'information topologique de la segmentation (caractéristiques des régions voisines, trous dans les régions, etc...) afin de proposer des algorithmes permettant, lors d'une seconde phase d'analyse, d'améliorer le résultat de la première segmentation obtenue avec les algorithmes de la section 3.1. J'ai montré également, grâce à des algorithmes basés sur une notion de précode, que nous étions capables de calculer efficacement ces structures. Je pense que mon approche combinatoire de l'imagerie, les propositions à base d'Union-Find, de cartes combinatoires, de critères de segmentation basés sur les inégalités de concentration ont ouvert la voie à un certain nombre de travaux et de perspectives qui permettent de se rapprocher de mon but initial : proposer des outils d'analyse s'appuyant à la fois sur une modélisation mathématique formelle, des structures de données expressives, et des algorithmes performants, afin d'obtenir des résultats pertinents efficacement. Il me semble clair que l'avenir se situe désormais dans l'utilisation de ces structures pour une analyse d'images plus efficace et de meilleure qualité. Quelques travaux ont ouvert la voie en ce sens, on notera par exemple une première proposition de G. Damiand et P. Resh pour un algorithme de segmentation de type division-fusion utilisant les cartes topologiques [57]. On remarquera également l'utilisation de cartes combinatoires pour de la segmentation basée sur « la ligne de partage des eaux » [41], ou encore la minimisation d'une mesure d'énergie basée sur des critères géométriques et colorimétriques dans une pyramide de cartes [62]. Bien sûr, dans le cadre de certaines applications, nous avons aussi essayé d'utiliser le graphe topologique des frontières afin de guider notre segmentation [64, 108]. Néanmoins, il me semble que l'avenir se situe plus dans l'utilisation de techniques d'algorithmes de graphes plus sophistiquées que dans ces propositions ad-hoc. Ainsi, on pourra par exemple essayer d'utiliser la notion de coupe normalisée dans les graphes, tel que le proposent J. Shi et J. Malik dans [209]. C'est ce que nous avons essayé de mettre en œuvre avec N. Gorretta et G. Rabatel [109]. D'autres résultats plus récents de clustering [67], de coupes normalisées basées sur des contraintes topologiques globales [237], ou d'apprentissage de paramètres de décision par une mesure d'incertitude sur les coupes dans les graphes [136], ouvrent

également de nouvelles perspectives qu'il me semble important d'explorer.

Enfin, au chapitre 4, j'ai proposé de nouvelles voies en géométrie discrète. D'abord, j'ai montré qu'en ne nous restreignant pas au seul cas des plans rationnels, nous pouvions utiliser de nombreux outils mathématiques classiques, hérités de l'analyse réelle, ou des systèmes dynamiques symboliques, possédant des analogues discrets aux propriétés structurelles très proches, et nous permettant de mieux comprendre la structure intrinsèque des objets discrets. C'est ainsi que nous avons pu démontrer que les hyperplans arithmétiques (et donc aussi les droites) de n'importe quelle épaisseur possédaient la propriété de *fonctionnalité généralisée*, et donc une structure 2D. Ensuite, dans l'idée de pouvoir coder de manière compacte les contours des régions segmentées, j'ai lancé une étude sur les objets discrets non linéaires et en particulier les cercles et hypersphères discrètes [87, 91], puis les courbes discrètes [92]. L'originalité de ces travaux vient du regard nouveau posé sur la notion d'épaisseur arithmétique des objets discrets. À l'origine ce paramètre était constant, nous proposons d'utiliser une *épaisseur variable*, basée sur une fonction de *variation discrète locale* et sur une *norme de k-minimalité*. Cette modélisation nous permet de lier l'épaisseur, non seulement à la forme de la courbe, mais aussi à la topologie de l'espace considéré. Cette approche, qui généralise la précédente, permet ainsi de vérifier certaines propriétés topologiques des objets discrets, comme la séparabilité, en fonction de leur épaisseur. Cette étude n'en est qu'à ses débuts, et beaucoup de travail doit être encore fait, notamment sur la compréhension de la notion de courbure, liée à celle de l'épaisseur, dans les objets discrets. C'est le sujet de la nouvelle thèse qui démarre et que nous co-encadrons avec Christian Mercat. Nous avons également un projet d'ANR sur les convolutions discrètes permettant une autre approche de la mesure de courbure.

Les résultats que j'ai présentés permettent une approche globale du problème de l'analyse d'images. La perspective de proposer des outils d'analyse d'images obtenant des résultats pertinents, s'appuyant à la fois sur une modélisation mathématique formelle, des structures de données expressives, et des algorithmes performants, est désormais proche. Cet objectif est le mien. Mais pour cela, il faut avancer dans les trois domaines explorés ici. En topologie numérique, il faut approfondir les liens entre les différentes propositions de la littérature et notamment préciser les notions de voisinage, région, adjacence, ensemble régulier, forte normalité, affichage, écran, dessin, afin de proposer une topologie les intégrant toutes, basée sur les notions d'espace quotient et de complexe cellulaire, et dont la relation d'ordre de spécialisation serait associée à la dimension. Ainsi on disposera d'un outil théorique permettant de faire le lien entre les représentations continues des objets discrets et les objets discrets eux-mêmes. Cette topologie doit désormais faire partie intégrante des algorithmes de segmentation. Actuellement, ceux que nous avons proposés, ainsi que ceux de la littérature, se contentent d'une seg-

mentation en $(n - 1)$ -connexité afin d'éviter les abbérations topologiques. Nous avons pourtant démontré que cette topologie n'était pas satisfaisante. Néanmoins, intégrer la topologie-étoile, ou une topologie équivalente, dans les algorithmes de segmentation en régions, demanderait, pour rester efficace, de s'appuyer, non seulement sur une structure Union-Find, mais aussi sur une structure modélisant l'adjacence entre régions. Il me semble donc qu'il est désormais nécessaire d'étudier la piste d'un algorithme de segmentation intégrant les cartes combinatoires comme modélisation de la topologie des régions, et de prendre en compte cette topologie dans les algorithmes de segmentation. Ceci devrait également nous amener à réfléchir à de nouveaux critères de segmentation, toujours basés sur les principes développés ici, mais intégrant cette fois-ci la possibilité de fusionner plusieurs régions en une seule fois. La dernière pierre à l'édifice d'un outil d'analyse global de l'image concerne la modélisation géométrique des objets segmentés. Si l'on sait depuis longtemps modéliser des droites et segmenter des contours en segments discrets, cela reste insuffisant pour une bonne modélisation géométrique. Il est important de s'attaquer aux objets non linéaires et de mieux les comprendre. C'est ce que nous avons commencé à faire, et je pense que l'approche que nous avons proposée, concernant l'épaisseur variable pour les objets discrets, apporte un regard nouveau sur cette épaisseur en lui donnant une justification topologique intéressante. Les perspectives ouvertes par cette voie sont à explorer et pourront sans doute, par une meilleure compréhension de la notion de courbure, nous permettre de proposer une définition générale pour les hypersurfaces et nous guider vers la reconnaissance et la segmentation de tels objets discrets.

Enfin, je terminerai par cette conviction qui est finalement celle qui guide mes recherches :

Afin de permettre une analyse complète de l'image, il est indispensable de s'appuyer sur un modèle topologique et statistique des images pour aborder de manière cohérente et globale le traitement des images numériques, c'est-à-dire la segmentation, la modélisation des frontières et le calcul de la géométrie des objets identifiés. Il est donc nécessaire d'étudier des modèles mathématiques dans ces différents domaines, puis de vérifier ces modèles en proposant des algorithmes de segmentation les utilisant, et de les mettre en œuvre dans le cadre d'applications afin de les valider.

Bibliographie

- [1] E. AHRONOVITZ, J.-P. AUBERT et C. FIORIO. : The star-topology : a topology for image analysis. *In 5th International Conference on Discrete Geometry for Computer Imagery (DGCI'05)*, p. 107–116. Groupe GDR PRC/AMI du CNRS, september 1995. citée pages [4](#), [10](#), [11](#), [20](#), [21](#), [30](#), [31](#), [33](#), [35](#), [39](#), [40](#), [46](#), [78](#), [92](#)
- [2] E. AHRONOVITZ, P. CHARNIER et C. FIORIO : A high level merging tool in image segmentation applications. *In Workshop on Machine Vision Applications (MVA'94)*, p. 218–221, Kawasaki, Japan, december 1994. IAPR. citée pages [3](#), [4](#), [5](#), [7](#), [67](#)
- [3] E. AHRONOVITZ, C. FIORIO et S. GLAIZE : Topological operator on the topological graph of frontiers. *In 8th International Conference on Discrete Geometry for Computer Imagery (DGCI'1999)*, num. 1568 de Lecture Notes in Computer Science, p. 207–217, Marne-la-Vallée, France, 1999. Disponible depuis <http://www.springerlink.com/content/hmgw3uv271btnc5k/>. citée pages [7](#), [64](#), [70](#), [73](#), [78](#), [93](#)
- [4] S. ALAYRANGUES : *Modèles et Invariants Topologiques en Imagerie Numérique*. Thèse de doctorat, Université Bordeaux I, july 2005. Disponible depuis <http://www.labri.fr/perso/alayrang/Memoire/SAthese-all.pdf>. citée pages [46](#)
- [5] S. ALAYRANGUES et J.-O. LACHAUD : Equivalence between order and cell complex representations. *In Proc. Computer Vision Winter Workshop 2002 (CVWW'02)*, p. 222–233, Bad Aussee, Austria, 2002. Disponible depuis <http://www.labri.fr/perso/alayrang/Recherche/preprintcvww.pdf>. citée pages [43](#), [46](#)
- [6] E. ANDRES : *Cercles Discrets et Rotations Discrètes*. Thèse de doctorat, Université Louis Pasteur, Strasbourg (France), 1992. citée pages [81](#), [86](#)
- [7] E. ANDRES : Discrete circles, rings and spheres. *Computers & Graphics*, 18(5):695–706, 1994. citée pages [86](#)
- [8] E. ANDRES : *Modélisation Analytique Discrète d'Objets Géométriques*. Habilitation à diriger des recherches, Université de Poitiers, Décembre 2000. citée pages [45](#), [81](#)

- [9] E. ANDRES, R. ACHARYA et C. SIBATA : Discrete analytical hyperplanes. *Graph. Models Image Process.*, 59(5):302–309, 1997. doi:[10.1006/gmip.1997.0427](https://doi.org/10.1006/gmip.1997.0427). citée pages [83](#)
- [10] E. ANDRES et M. JACOB : The discrete analytical hyperspheres. *Visualization and Computer Graphics (IEEE Transactions on)*, 3(1):75–86, 1997. citée pages [86](#), [88](#)
- [11] L. A. ANKENY et G. X. RITTER : Cellular topology and its applications in image processing. *International Journal of Parallel Programming*, 12(6):433–456, 1983. doi:[10.1007/BF00977969](https://doi.org/10.1007/BF00977969). citée pages [22](#), [23](#), [31](#)
- [12] S. ANSALDI, L. de FLORIANI et B. FALCIDIENO : An edge-face relational scheme for boundary representations. *Computer Graph Forum*, 4:319–332, 1985. citée pages [49](#)
- [13] R. AYALA., E. DOMINGUEZ., A. FRANCES et A. QUINTERO. : Determining the components of the complement of a digital $(n - 1)$ -manifold in z_n . In S. MIGUET, A. MONTANVERT et S. UBÉDA, éd. : *6th International Conference on Discrete Geometry for Computer Imagery (DGCI'96)*, vol. 1176 de *Lecture Notes in Computer Sciences*, p. 163–174, Lyon, France, november 1996. Springer-Verlag, Berlin, Germany. citée pages [31](#), [46](#)
- [14] R. AYALA, E. DOMINGUEZ, A. FRANCES et A. QUINTERO : Digital lighting functions. In *International Conference on Discrete Geometry for Computer Imagery (DGCI'97)*, vol. 1347 de *Lecture Notes in Computer Science*, p. 137–150. Springer-Verlag, 1997. doi:[10.1007/BFb0024836](https://doi.org/10.1007/BFb0024836). citée pages [31](#), [40](#), [42](#), [43](#), [46](#)
- [15] R. AYALA, E. DOMINGUEZ, A. R. FRANCES et A. QUINTERO : Weak lighting functions and strong 26-surfaces. *Theoretical Computer Science*, 283(1):29–66, 2002. doi:[10.1016/S0304-3975\(01\)00051-2](https://doi.org/10.1016/S0304-3975(01)00051-2). citée pages [40](#), [42](#), [43](#), [46](#)
- [16] R. AYALA, E. DOMINGUEZ, A. R. FRANCES et A. QUINTERO : Digital homotopy with obstacles. *Discrete Applied Mathematics*, 139(1-3):5–30, 2004. doi:[10.1016/S0304-3975\(01\)00051-2](https://doi.org/10.1016/S0304-3975(01)00051-2). citée pages [42](#), [43](#)
- [17] V. BELLON-MAUREL, N. GORRETTA, G. RABATEL et J.-M. ROGER : From multispectral to hyperspectral imaging for analysing agricultural product quality : A critical review and opportunities. In *12th International Conference on Near-Infrared Spectroscopy (NIR2005)*, avril 2005. citée pages [56](#)
- [18] V. BERTHE, C. FIORIO et D. JAMET : Generalized functionality for arithmetic discrete planes. In P. L. E. ANDRES, G. Damiand, éd. : *12th International Conference on Discrete Geometrey for Computer Imagery (DGCI'05)*, num. 3429 de LNCS, p. 276–286. Springer, avril 2005. doi:[10.1007/b135490](https://doi.org/10.1007/b135490). citée pages [7](#), [78](#)
- [19] V. BERTHE, C. FIORIO, D. JAMET et F. PHILIPPE : On some applications of generalized functionality for arithmetic discrete planes. *Image and Vision Com-*

- puting, 25(10):1671–1684, october 2007. doi:10.1016/j.imavis.2006.06.023. citée pages iv, 7, 18, 78, 83, 84, 207
- [20] G. BERTRAND et R. MALGOUYRES : Some topological properties of surfaces in $\mathbb{Z}^3$. *Journal of Mathematical Imaging and Vision*, 11(3):207–221, 1999. doi:10.1023/A:1008348318797. citée pages 18
- [21] Y. BERTRAND, G. DAMIAND et C. FIORIO : Topological encoding of 3d segmented images. In G. BORGEFORS, I. NYSTRÖM et G. S. di BAJA, édés : *9th International Conference on Discrete Geometry for Computer Imagery (DGCI'00)*, num. 1953 de Lecture Notes in Computer Science, p. 311–324, Uppsala (Sweden), december 2000. Disponible depuis <http://www.springerlink.com/content/r2fmm57qlnf17ehx>. citée pages 7, 56, 64, 70, 73, 78, 93
- [22] Y. BERTRAND, G. DAMIAND et C. FIORIO : Topological map : minimal encoding of 3d segmented images. In *Workshop on Graph-Based Representations in Pattern Recognition (GBR'01)*, p. 64–73, Ischia, Italy, may 2001. IAPR-TC15. citée pages 7, 40, 64, 70, 73, 78, 93
- [23] Y. BERTRAND et J.-F. DUFOURD : Algebraic specification of a 3d-modeler based on hypermaps. *CVGIP : Graphical Models and Image Processing*, 56(1):29–60, january 1994. citée pages 65
- [24] Y. BERTRAND, C. FIORIO et Y. PENNANEACH : Border map : a topological representation for nd image analysis. In *8th International Conference on Discrete Geometry for Computer Imagery (DGCI'1999)*, num. 1568 de LNCS, p. 242–257, Marne-la-Vallée, France, 1999. Disponible depuis <http://www.springerlink.com/content/xdu9g3q53cc2jpw/>. citée pages 7, 40, 64, 70, 72, 73, 78, 93
- [25] D. BLOSTEIN et N. AHUJA : A multiscale region detector. *Computer Vision, Graphics, and Image Processing.*, 45:22–41, 1989. citée pages 57
- [26] J.-P. BRAQUELAIRE et L. BRUN : Image segmentation with topological maps and inter-pixel representation,. *Journal of Visual Communication and Image Representation*, 9(1):62–79, 1998. doi:10.1006/jvci.1998.0374. citée pages 39, 73, 90
- [27] J.-P. BRAQUELAIRE et J.-P. DOMENGER : Representation of segmented images with discrete geometric maps. *Image and Vision Computing*, 17(10):715–735, 1999. doi:10.1016/S0262-8856(98)00152-8. citée pages 56
- [28] J.-P. BRAQUELAIRE et A. VIALARD : Euclidean paths : A new representation of boundary of discrete regions;. *Graphical Models and Image Processing*, 61(1):16–43, 1999. doi:10.1006/gmip.1999.0488. citée pages 90
- [29] J. BRESENHAM : Algorithm for computer control of a digital plotter. *IBM Systems Journal*, 4(1):25–30, 1965. citée pages 78

- [30] J. BRESENHAM : A linear algorithm for incremental digital display of circular arcs. *Communications of the ACM*, 20(2):100–106, 1977. citée pages 84
- [31] V. BRIMKOV, E. ANDRÈS et R. BARNEVA : Object discretization in higher dimensions. In *Discrete Geometry for Computer Imagery. 9th International Conference, DGCI 2000. Proceedings*, vol. 1953 de *Lecture Notes in Computer Science*, p. 210–221. Springer, 2000. Disponible depuis <http://www.springerlink.com/content/52qy4d5jb065tpe7>. citée pages 26
- [32] V. BRIMKOV, D. COEURJOLLY et R. KLETTE : Digital planarity—a review ;. *Discrete Applied Mathematics*, 155(4):468–495, 2007. doi:10.1016/j.dam.2006.08.004. citée pages 18
- [33] V. BRIMKOV et S. S. DANTCHEV : Complexity analysis for digital hyperplane recognition in arbitrary fixed dimension. In *12th International Conference on Discrete Geometry for Computer Imagery (DGCI'05)*, vol. 3429 de *Lecture Notes in Computer Science*, p. 287–298, Poitiers, France, avril 2005. Springer Berlin / Heidelberg. 10.1007/b135490. citée pages 90
- [34] V. BRIMKOV et R. KLETTE : Curves, hypersurfaces, and good pairs of adjacency relations. In *10th International Workshop on Combinatorial Image Analysis (IWCIA'04)*, vol. 3322 de *Lecture Notes in Computer Science*, p. 276–290, Auckland, New Zealand, november 2004. Springer Berlin / Heidelberg. doi:10.1007/b103936. citée pages 18
- [35] V. E. BRIMKOV, E. ANDRES et R. P. BARNEVA : Object discretizations in higher dimensions ;. *Pattern Recognition Letters*, 23(6):623–636, 2002. doi:10.1016/S0167-8655(01)00139-8. citée pages 26
- [36] V. E. BRIMKOV et R. P. BARNEVA : Graceful planes and lines. *Theoretical Computer Science*, 283(1):151–170, 2002. doi:10.1016/S0304-3975(01)00061-5. citée pages 83
- [37] V. E. BRIMKOV et R. P. BARNEVA : Connectivity of discrete planes. *Theoretical Computer Science*, 319(1-3):203–277, june 2004. doi:10.1016/j.tcs.2004.02.015. citée pages 18, 83
- [38] V. E. BRIMKOV et S. DANTCHEV : Digital hyperplane recognition in arbitrary fixed dimension within an algebraic computation model. *Image and Vision Computing*, 25(10):1631–1643, 2007. doi:10.1016/j.imavis.2006.06.013. citée pages 90
- [39] L. BRUN : *Segmentation d'images couleur à base topologique*. Thèse de doctorat, Université Bordeaux I, december 1996. citée pages 73, 90
- [40] L. BRUN, J.-P. DOMENGER et M. MOKHTARI : Incremental modifications of segmented image defined by discrete maps. *Journal of Visual Communication*

- and Image Representation*, 14(3):251–290, 2003. doi:10.1016/S1047-3203(03)00023-3. citée pages 56, 73
- [41] L. BRUN, M. MOKHTARI et F. MEYER : Hierarchical watersheds within the combinatorial pyramid framework. *Lecture Notes in Computer Science*, 3429:34–44, 2005. citée pages 56, 93
- [42] L. BUZER : A linear incremental algorithm for naive and standard digital lines and planes recognition. *Graph. Models*, 65(1-3):61–76, 2003. doi:10.1016/S1524-0703(03)00008-0. citée pages 90
- [43] F. CHANG, C.-J. CHEN et C.-J. LU : A linear-time component-labeling algorithm using contour tracing technique. *Computer Vision and Image Understanding*, 93(2):206–220, 2004. doi:10.1016/j.cviu.2003.09.002. citée pages 56
- [44] P. CHARNIER : *Outils algorithmiques pour le codage interpixel et ses applications*. Thèse de doctorat, Université Montpellier II, Janvier 1995. citée pages 5
- [45] J.-M. CHASSERY : Connectivity and consecutivity in digital pictures. *Computer Graphics and Image Process.*, 9:294–300, 1979. doi:10.1016/0146-664X(79)90043-1. citée pages 14
- [46] K.-L. CHUNG, H.-L. HUANG et H.-I. LU : Efficient region segmentation on compressed gray images using quadtree and shading representation;. *Pattern Recognition*, 37(8):1591–1605, 2004. doi:10.1016/j.patcog.2004.02.009. citée pages 56
- [47] D. COEURJOLLY, Y. GERARD, J. P. REVEILLES et L. TOUGNE : An elementary algorithm for digital arc segmentation; the 2001 international workshop on combinatorial image analysis. *Discrete Applied Mathematics*, 139(1-3):31–50, 2004. doi:10.1016/j.dam.2003.08.003. citée pages 18
- [48] R. CORI : Un code pour les graphes planaires et ses applications. In *Astérisque*, vol. 27. Soc. Math. de France, Paris, France, 1975. citée pages 65
- [49] T. CORMEN, C. LEISERSON et R. RIVEST : *Introduction à l’algorithmique : Cours et Exercices*. Dunod, 2002. citée pages 51
- [50] M. COUPRIE, G. BERTRAND et Y. KENMOCHI : Discretization in 2d and 3d orders; special issue : Discrete topology and geometry for image and object representation. *Graphical Models*, 65(1-3):77–91, 2003. doi:10.1016/S1524-0703(03)00003-1. citée pages 26, 45
- [51] J. COUSTY, M. COUPRIE, L. NAJMAN et G. BERTRAND : Weighted fusion graphs : Merging properties and watersheds. *Discrete Applied Mathematics*, In Press, Corrected Proof:–, 2008. doi:10.1016/j.dam.2008.01.005. citée pages 56
- [52] J.-P. D’ALES, J. FROMENT et J.-M. MOREL : Reconstruction visuelle et généricité. *Intellectica*, 1(28):11–35, 1999. citée pages 2

- [53] G. DAMIAND : *Définition et étude d'un modèle topologique minimal de représentation d'images 2d et 3d*. Thèse de doctorat, Université Montpellier II, december 2001. citée pages 56, 64
- [54] G. DAMIAND : Topological model for 3d image representation : Definition and incremental extraction algorithm. *Computer Vision and Image Understanding*, 109(3):260–289, 2008. doi:10.1016/j.cviu.2007.09.007. citée pages 74
- [55] G. DAMIAND, Y. BERTRAND et C. FIORIO : Topological model for two-dimensional image representation : definition and optimal extraction algorithm. *Computer Vision and Image Understanding*, 93(2):111–154, february 2004. doi:10.1016/j.cviu.2003.09.001. citée pages iv, 7, 64, 73, 78, 93, 161, 206
- [56] G. DAMIAND, M. DEXET-GUIARD, P. LIENHARDT et E. ANDRES : Removal and contraction operations to define combinatorial pyramids : application to the design of a spatial modeler. *Image and Vision Computing*, 23(2):259–269, 2005. doi:10.1016/j.imavis.2004.06.016. citée pages 74, 75
- [57] G. DAMIAND et P. RESCH : Split-and-merge algorithms defined on topological maps for 3d image segmentation ; special issue : Discrete topology and geometry for image and object representation. *Graphical Models*, 65(1-3):149–167, 2003. doi:10.1016/S1524-0703(03)00009-2. citée pages 39, 56, 74, 93
- [58] G. D'ANDRÉA et C. FIORIO : Maximal superpositions of grids and an application. In *7th International Conference on Discrete Geometry for Computer Imagery (DGCI'97)*, num. 1347 de Lecture Notes in Computer Science, p. 63–71, Montpellier, France, december 1997. Springer. doi:10.1007/BFb0024830. citée pages 5, 78
- [59] G. D'ANDRÉA et C. FIORIO : Maximal superpositions of horizontally convex polyominoes. *Theoretical Computer Science*, 218:273–283, 1999. doi:10.1016/S0304-3975(98)00326-0. citée pages 5, 78
- [60] X. DARAGON, M. COUPRIE et G. BERTRAND : Derived neighborhoods and frontier orders. *Discrete Applied Mathematics*, 147(2-3):227–243, 2005. doi:10.1016/j.dam.2004.09.013. citée pages 45
- [61] X. DARAGON, M. COUPRIE et G. BERTRAND : Discrete surfaces and frontier orders. *Journal of Mathematical Imaging and Vision*, 23(3):379–399, 2005. doi:10.1007/s10851-005-2029-4. citée pages 45
- [62] M. de CALIGNON, L. BRUN et J.-O. LACHAUD : Combinatorial pyramids and discrete geometry for energy-minimizing segmentation. *Advances in Visual Computing*, p. 306–315, 2006. doi:10.1007/11919629_32. citée pages 93
- [63] B. de MEZO : *Weed leaves Recognition in complex Natural Scenes by Model-Guided*

- Edge Pairing*. Thèse de doctorat, Université de Montpellier II Sciences et Techniques du Languedoc, novembre 2004. citée pages 55, 74
- [64] B. de MEZZO, C. FIORIO et G. RABATEL : Weed leaves recognition in complex natural scenes by model-guided edge pairing. In J. STAFFORD et A. WERNE, édés : *4th European Conference on Precision Agriculture*, p. 141–147, Berlin, juin 2003. Wageningen Academic Publishers. citée pages 8, 55, 74, 93
- [65] I. DEBLED-RENESSON : *Étude et reconnaissance des droites et plans discrets*. Thèse de doctorat, Université Louis-Pasteur de Strasbourg, Strasbourg, décembre 1995. citée pages 84, 90
- [66] I. DEBLED-RENESSON et J.-P. REVEILLES : New approach to digital planes. In R. A. MELTER et A. Y. WU, édés : *Vision Geometry III*, p. 12–21, Boston, MA, USA, janvier 1995. doi:10.1117/12.198607. citée pages 83
- [67] E. D. DEMAINE, D. EMANUEL, A. FIAT et N. IMMORLICA : Correlation clustering in general weighted graphs; approximation and online algorithms. *Theoretical Computer Science*, 361(2-3):172–187, 2006. doi:10.1016/j.tcs.2006.05.008. citée pages 93
- [68] P. DESBARATS et S. GUEORGUEVA : Cw complexes : Topological mainframe for numerical representations of objects. In *Computational Science and Its Applications (ICCSA 2003)*, vol. 2669 de *Lecture Notes in Computer Science*, p. 986–986. Springer, 2003. Disponible depuis <http://www.springerlink.com/content/c7gdp1x3k35h0drn>. citée pages 39
- [69] A. DIETZ, C. FIORIO, M. HABIB et C. LAURENÇO : Topology-based stereochemistry representation. *Compte Rendu de l'Académie des Sciences de Paris*, 1(2):95–100, février 1998. doi:10.1016/S1251-8069(97)86267-7. citée pages 8, 74, 78
- [70] A. DIETZ, C. FIORIO, M. HABIB et C. LAURENÇO : Representation of stereochemistry using combinatorial maps. In *Discrete Mathematical Chemistry*, vol. 51 de *Series in Discrete Mathematics and Theoretical Computer Science*, p. 117–128, 2000. Disponible depuis <http://www.ams.org/mathscinet-getitem?mr=1762935>. citée pages 8, 74
- [71] M. DILLEN COURT, H. SAMET et M. TAMMINEN : A general approach to connected-component labeling for arbitrary image representation. *J. of the Association for Computing Machinery*, 39(2):253–280, avril 1992. Corr. p. 985–986. doi:10.1145/128749.128750. citée pages 51
- [72] E. DOMINGUEZ et A. FRANCES : An axiomatic approach to digital topology. In *Digital and Image Geometry, Advanced Lectures [based on a winter school held at Dagstuhl Castle, Germany in December 2000]*, vol. 2243 de *Lecture Notes in*

- Computer Science*, p. 35–74. Springer Berlin, 2001. Disponible depuis <http://www.springerlink.com/content/51epbktvvhx2p/>. citée pages 31
- [73] E. DOMINGUEZ, A. FRANCES et A. MARQUEZ : A framework for digital topology. *Systems, Man and Cybernetics, 1993. IEEE Conference Proceedings., International Conference on*, 2:65–70, 17-20 Oct 1993. doi:10.1109/ICSMC.1993.384851. citée pages 40, 42, 46
- [74] R. O. DUDA, P. E. HART et J. H. MUNSON : Graphical data processing research study and experimental investigation. Technical Report ECOM-01901-26. 26, AI Center, SRI International, 333 Ravenswood Ave, Menlo Park, CA 94025, march 1967. SRI Project 5864. Prepared for US Army Electronics Command. Contract DA 28-043 AMC-01901(E). Disponible depuis http://www.ai.sri.com/pub_list/1343. citée pages 14, 17
- [75] U. ECKHARDT et L. J. LATECKI : Topologies for the digital spaces z^2 and z^3 . *Computer Vision and Image Understanding*, 90(3):295–312, 2003. doi:10.1016/S1077-3142(03)00062-6. citée pages 18
- [76] J. EDMONDS : A combinatorial representation for polyhedral surfaces. *Notices Amer. Math. Soc.*, 7:646, 1960. citée pages 65
- [77] A. V. EVAKO, R. KOPPERMAN et Y. V. MUKHIN : Dimensional properties of graphs and digital spaces. *Journal of Mathematical Imaging and Vision*, 6(2):109–119, 1996. doi:10.1007/BF00119834. citée pages 45
- [78] J. FEDER : Languages of encoded line patterns. *Information Control*, 13(3):230–244, 1968. doi:10.1016/S0019-9958(68)91105-4. citée pages 10
- [79] T. FERNIQUE : Generation and recognition of digital planes using multi-dimensional continued fractions. In D. COEURJOLLY, I. SIVIGNON, L. TOUGNE et F. DUPONT, eds : *Discrete Geometry for Computer Imagery, 14th International Conference, DGCI 2008*, vol. 4992 de *Lecture Notes in Computer Science*, p. 33–44, Lyon, France, 2008. doi:10.1007/978-3-540-79126-3_5. citée pages 90
- [80] O. FIGUEIREDO et J.-P. REVEILLÈS : A contribution to 3d digital lines. In *5th International Conference on Discrete Geometry for Computer Imagery (DGCI'95)*, p. 187–189, september 1995. citée pages 83
- [81] C. FIORIO : *Approche interpixel en analyse d'images : une topologie et des algorithmes de segmentation*. Thèse de doctorat, Université Montpellier II, november 1995. citée pages 4, 10, 11, 20, 21, 30, 31, 33, 35, 38, 40, 44, 46, 64, 68, 69, 73, 78, 92
- [82] C. FIORIO : A topologically consistent representation for image analysis : the frontiers topological graph. In S. MIGUET, A. MONTANVERT et U. S., eds : *6th International Conference on Discrete Geometry for Computer Imagery (DGCI'96)*,

- num. 1176 de Lecture Notes in Computer Sciences, p. 151–162, Lyon, France, november 1996. Disponible depuis <http://www.springerlink.com/content/hmgw3uv271btnc5k/>. citée pages 7, 10, 31, 40, 56, 68, 69, 73, 78, 93
- [83] C. FIORIO : Maximal polyominoes superposition and extension to polyocubes. International Workshop on Discrete Tomography and Related Problem, Thionville, France, march 1999. (invited Talk). citée pages 5, 78
- [84] C. FIORIO et J. GUSTEDT : Two linear time Union-Find strategies for image processing. *Theoretical Computer Science*, 154:165–181, 1996. doi:10.1016/0304-3975(94)00262-2. citée pages iv, 5, 6, 48, 52, 54, 55, 56, 57, 93, 127
- [85] C. FIORIO et J. GUSTEDT : Memory management for union-find algorithms. In R. REISCHUK, éd. : *14th Symposium on Theoretical Aspects of Computer Science (STACS '97)*, vol. 1200 de *Lecture Notes in Computer Science*, p. 67–79, Luebeck, Germany, february 1997. Springer-Verlag. doi:10.1007/BFb0023449. citée pages 6, 52, 93
- [86] C. FIORIO, J. GUSTEDT et T. LANGE : Union-find volume segmentation. In *7th International Workshop on Combinatorial Image Analysis (IWCIA'00)*, p. 181–197, Caen, France, 2000. citée pages 6, 52, 93
- [87] C. FIORIO, D. JAMET et J.-L. TOUTANT : Discrete circles : an arithmetical approach with non-constant thickness. In A. Y. W. LONGIN JEAN LATECKI, David M. Mount, éd. : *Vision Geometry XIV, Electronic Imaging, SPIE*, vol. 6066, p. 60660C, San Jose (CA), USA, january 2006. SPIE. doi:10.1117/12.642976. citée pages 7, 8, 78, 81, 84, 87, 90, 94
- [88] C. FIORIO et R. NOCK : Image segmentation using a generic, fast and non-parametric approach. In *Tools with Artificial Intelligence, 10th IEEE International Conference on*, p. 450–458, Taipei, Taiwan, R.O.C., november 1998. IEEE Computer Society. doi:10.1109/TAI.1998.744885. citée pages 6, 55, 56, 61, 93
- [89] C. FIORIO et R. NOCK : A concentration-based adaptive approach to region merging of optimal time and space complexities. In *British Machine Vision Conference*, vol. 02, p. 775–784, Bristol, England, 2000. Disponible depuis <http://www.bmva.ac.uk/bmvc/2000/papers/p78.pdf>. citée pages iv, 6, 56, 57, 59, 60, 61, 93, 145, 160
- [90] C. FIORIO et R. NOCK : Sorted region merging to maximize test reliability. In *International Conference on Image Processing*, vol. 01, p. 808–811, Vancouver, Canada, 2000. IEEE. doi:10.1109/ICIP.2000.901082. citée pages iv, 6, 55, 56, 57, 60, 61, 93, 145, 160
- [91] C. FIORIO et J.-L. TOUTANT : Arithmetic discrete hyperspheres and separateness. In *13th International Conference on Discrete Geometry for Computer Im-*

- gery (DGCI'06), vol. 4245 de *Lecture Notes in Computer Science*, p. 425–436, Szeged, Hungary, october 2006. Springer. doi:10.1007/11907350_36. citée pages 7, 8, 18, 78, 81, 84, 88, 89, 90, 94
- [92] C. FIORIO et J.-L. TOUTANT : Arithmetic characterization of polynomial based discrete curves. In *Vision Geometry XV, IS&T/SPIE 19th Annual Symposium on Electronic Imaging*, vol. 6499, p. 64990F1–64990F8, San Jose, California, USA, January 2007. SPIE. doi:10.1117/12.714475. citée pages 7, 8, 78, 81, 84, 90, 94
- [93] S. FOUREY, T. Y. KONG et G. T. HERMAN : Generic axiomatized digital surface-structures. *Discrete Applied Mathematics*, 139(1-3):65–93, 2004. doi:10.1016/j.dam.2003.03.001. citée pages 18, 20
- [94] J. FRANÇON : On recent trends in discrete geometry in computer science. In S. U. S. MIGUET, A. Montanvert, éd. : *6th International Conference on Discrete Geometry for Computer Imagery (DGCI'96)*, num. 1176 de *Lecture Notes in Computer Sciences*, p. 1–16, Lyon, France, november 1996. doi:10.1007/3-540-62005-2_1. citée pages 2, 11
- [95] H. FREEMAN : On the encoding of arbitrary geometric configurations. *Electronic Computers, IEEE Transactions on*, EC-10:260–268, june 1961. citée pages 10
- [96] H. FREEMAN : Computer processing of line-drawing images. *ACM Computing Surveys*, 6(1):57–97, march 1974. doi:10.1145/356625.356627. citée pages 67
- [97] H. N. GABOW et R. E. TARJAN : A linear-time algorithm for a special case of disjoint set union. *Journal of Computer and System Sciences*, 30(2):209–221, 1984. citée pages 51
- [98] Z. GALIL et G. F. ITALIANO : Data structures and algorithms for disjoint set union problems. *ACM Computing Surveys*, 23(3):319–344, september 1991. citée pages 51
- [99] A. GALTON : A generalized topological view of motion in discrete space. *Theoretical Computer Science*, 305(1-3):111–134, 2003. doi:10.1016/S0304-3975(02)00701-6. citée pages 27, 39
- [100] J.-P. GAMBOTTO : A new approach to combining region growing and edge detection. *Pattern Recognition Letters*, 14(11):869–875, 1993. doi:10.1016/0167-8655(93)90150-C. citée pages 75
- [101] M. GANGNET, J.-C. HERVÉ, T. PUDET et J.-M. THONG : Incremental computation of planar maps. report 1, Digital PRL, 5 1989. citée pages 65
- [102] Y. GERARD : Local configurations of digital hyperplanes. In *8th International Conference on Discrete Geometry for Computer Imagery (DGCI'99)*, vol. 1568

- de *Lecture Notes in Computer Science*, p. 65–75, Marne-la-Vallée, France, march 1999. Springer Berlin / Heidelberg. [10.1007/3-540-49126-0](#). citée pages [84](#)
- [103] Y. GERARD : Periodic graphs and connectivity of the rational digital hyperplanes;. *Theoretical Computer Science*, 283(1):171–182, 2002. [doi:10.1016/S0304-3975\(01\)00063-9](#). citée pages [83](#)
- [104] Y. GERARD : A fast and elementary algorithm for digital plane recognition; 9th international workshop on combinatorial image analysis. *Electronic Notes in Discrete Mathematics*, 12:142–153, 2003. [doi:10.1016/S1571-0653\(04\)00481-0](#). citée pages [90](#)
- [105] Y. GERARD, I. DEBLED-RENNESON et P. ZIMMERMANN : An elementary digital plane recognition algorithm; iwcia 2003-ninth international workshop on combinatorial image analysis. *Discrete Applied Mathematics*, 151(1-3):169–183, 2005. [doi:10.1016/j.dam.2005.02.026](#). citée pages [90](#)
- [106] G. GOEL et J. GUSTEDT : Bounded arboricity to determine the local structure of sparse graphs. *Lecture Notes in Computer Science*, 4271:159–167, 2006. citée pages [56](#)
- [107] N. GORRETTA : Segmentation d’images naturelles par collaboration de méthodes de segmentation régions–contours. Mémoire de dea, Université de Montpellier II Sciences et Techniques du Languedoc, june 2003. citée pages [55](#), [74](#)
- [108] N. GORRETTA, C. FIORIO, G. RABATEL et J. MARCHANT : Cabbage/weed discrimination with a region/contour based segmentation approach for multispectral images. In *FRUTIC 05, Information and technology for sustainable fruit and vegetable production*, Montpellier, France, september 2005. citée pages [8](#), [56](#), [74](#), [93](#)
- [109] N. GORRETTA, G. RABATEL, C. FIORIO, J.-M. ROGER, C. LELONG et V. BELLON-MAUREL : Proposition d’une stratégie de segmentation d’images hyperspectrales. accepted for publication in *Revue Traitement du signal*, 2008, july. citée pages [8](#), [74](#), [75](#), [93](#)
- [110] N. GORRETTA, G. RABATEL, J.-M. ROGER, C. FIORIO, C. LELONG et V. BELLON-MAUREL : Hyperspectral imaging system calibration using image translations and fourier transform. *Journal of Near Infrared Spectroscopy*, 16(4):371–380, 2008. [doi:10.1255/jnirs.809](#). citée pages [8](#), [56](#), [74](#), [93](#)
- [111] C. GRASSET-SIMON et G. DAMIAND : Generalized map pyramid for multi-level 3d image segmentation. *Lecture Notes in Computer Science*, 4245:530–541, 2006. citée pages [56](#), [74](#), [75](#)
- [112] C. GRASSET-SIMON, G. DAMIAND et P. LIENHARDT : nd generalized map pyramids : Definition, representations and basic operations; graph-based representa-

- tions. *Pattern Recognition*, 39(4):527–538, 2006. doi:[10.1016/j.patcog.2005.10.004](https://doi.org/10.1016/j.patcog.2005.10.004). citée pages [74](#), [75](#)
- [113] GRUNBAÜM : *Convex polytopes*, vol. 16 de *Pure and Applied Mathematics*. John Wiley, interscience édn, 1967. citée pages [33](#)
- [114] J. GUSTEDT : Efficient union-find for planar graphs and other sparse graph classes. *Theoretical Computer Science*, 203(1):123–141, 1998. doi:[10.1016/S0304-3975\(97\)00291-0](https://doi.org/10.1016/S0304-3975(97)00291-0). citée pages [56](#)
- [115] R. M. HARALICK et L. G. SHAPIRO : Survey : Image segmentation techniques. *Computer Vision, Graphics, and Image Processing.*, 29:100–132, 1985. citée pages [50](#), [54](#)
- [116] R. M. HARALICK et L. G. SHAPIRO : *Computer and Robot Vision*, vol. 1-2. Addison-Wesley Publishing Company, Inc., june 1992. citée pages [48](#)
- [117] L. HE, Y. CHAO et K. SUZUKI : A linear-time two-scan labeling algorithm. In *IEEE International Conference on Image Processing (ICIP 2007)*, vol. 5, p. V–241 – V–244, San Antonio (USA), september 2007. IEEE Computer Society. doi:[10.1109/ICIP.2007.4379810](https://doi.org/10.1109/ICIP.2007.4379810). citée pages [56](#)
- [118] G. HERMAN : On topology as applied to image analysis. *Computer Vision, Graphics, and Image Processing.*, 52:409–415, 1990. doi:[10.1016/0734-189X\(90\)90084-9](https://doi.org/10.1016/0734-189X(90)90084-9). citée pages [22](#), [29](#)
- [119] G. HERMAN et D. WEBSTER : A topological proof of a surface tracking algorithm. *Computer Vision, Graphics, and Image Processing.*, 23:162–177, 1983. citée pages [22](#)
- [120] R. HORAUD : La vision artificielle. *Traitement du Signal*, 4(3):1–5, 1987. Éditorial. citée pages [3](#)
- [121] R. HORAUD et O. MONGA : *Vision par Ordinateur, outils fondamentaux*. Trait[é] des nouvelles technologie, s[é]rie informatique. Hermes, Paris, 1993. citée pages [3](#)
- [122] S. L. HOROWITZ et T. PAVLIDIS : Picture segmentation by a directed split-and-merge procedure. In *International Conference on Pattern Recognition*, p. 424–433, 1974. citée pages [49](#), [50](#)
- [123] S. HOUELLE et G. GIRAUDON : Model based region segmentation using cooccurrence matrices. *International Conference on Vision and Pattern Recognition (CVPR'92)*, 3:636–639, june 1992. citée pages [57](#)
- [124] Q. HU, S. LUO, Y. QIAO et G. QIAN : Supervised grayscale thresholding based on transition regions. In Press, Accepted Manuscript, 2008. doi:[10.1016/j.imavis.2008.05.003](https://doi.org/10.1016/j.imavis.2008.05.003). citée pages [55](#)

- [125] M.-A. JACOB-DA COL : About local configurations in arithmetic planes ;. *Theoretical Computer Science*, 283(1):183–201, 2002. doi:10.1016/S0304-3975(01)00064-0. citée pages 84
- [126] A. JACQUES : Constellations et graphes topologiques. In *Combinatorial Theory and Applications*, p. 657–673, Budapest, 1970. citée pages 65
- [127] J. JORDAN, B. W., W. J. LENNON et B. D. HOLM : An improved algorithm for the generation of nonparametric curves. *Computers, IEEE Transactions on*, C-22(12):1052–1060, 1973. Disponible depuis http://ieeexplore.ieee.org/xpls/abs_all.jsp?isnumber=35072&arnumber=1672243&count=17&index=2. citée pages 84
- [128] C. JUNG : Unsupervised multiscale segmentation of color images ;. *Pattern Recognition Letters*, 28(4):523–533, 2007. doi:10.1016/j.patrec.2006.10.001. citée pages 61
- [129] C. KERMAD et K. CHEHDI : Automatic image segmentation system through iterative edge-region co-operation ;. *Image and Vision Computing*, 20(8):541–555, 2002. doi:10.1016/S0262-8856(02)00043-4. citée pages 75
- [130] E. KHALIMSKY : Topological structures in computer science. *Journal of Applied Mathematics and Simulation*, 1(1):25–40, 1987. doi:10.1155/S1048953388000036. citée pages 11, 20, 21, 22, 23, 28, 39
- [131] E. KHALIMSKY, R. KOPPERMAN et P. R. MEYER : Boundaries in digital planes. *J. of Applied Mathematics and Stochastic Analysis*, 3(1):27–55, 1990. doi:10.1155/S1048953390000041. citée pages 23, 24, 25, 26, 30, 36, 46
- [132] E. KHALIMSKY, R. KOPPERMAN et P. R. MEYER : Computer graphics and connected topologies on finite ordered sets. *Topology and its Applications*, 36(1):1–17, 1990. doi:10.1016/0166-8641(90)90031-V. citée pages 20, 21, 22, 23, 24, 40
- [133] R. KLETTE : Digital topology for image analysis - part 1. Rap. tech. CITR-TR-101, University of Auckland, october 2001. Disponible depuis <http://www.citr.auckland.ac.nz/techreports/show.php?id=101>. citée pages 11
- [134] R. KLETTE : Topologies on the planar orthogonal grid. In *16th International Conference on Pattern Recognition (ICPR'02)*, vol. 02, p. 20354, Los Alamitos, CA, USA, 2002. IEEE Computer Society. doi:10.1109/ICPR.2002.1048312. citée pages 11, 28, 29
- [135] K. KOFFKA : *Principles of Gestalt Psychology*. Harcourt Brace, new-york édn, 1935. citée pages 2
- [136] P. KOHLI et P. TORR : Measuring uncertainty in graph cut solutions. *Computer Vision and Image Understanding*, 112(1):30–38, 2008. doi:10.1016/j.cviu.2008.07.002. citée pages 93

- [137] A. KOLESNIKOV et P. FRANTI : Polygonal approximation of closed discrete curves. *Pattern Recognition*, 40(4):1282–1293, 2007. doi:[10.1016/j.patcog.2006.09.002](https://doi.org/10.1016/j.patcog.2006.09.002). citée pages [18](#)
- [138] T. Y. KONG : A digital fundamental group. *Computers & Graphics*, 13(2):159–166, 1989. doi:[10.1016/0097-8493\(89\)90058-7](https://doi.org/10.1016/0097-8493(89)90058-7). citée pages [19](#)
- [139] T. Y. KONG : Topological adjacency relations on $\mathbb{Z}^n$. *Theoretical Computer Science*, 283(1):3–28, 2002. doi:[10.1016/S0304-3975\(01\)00050-0](https://doi.org/10.1016/S0304-3975(01)00050-0). citée pages [18](#), [20](#)
- [140] T. Y. KONG : The khalimsky topologies are precisely those simply connected topologies on whose connected sets include all $2n$ -connected sets but no $(3n-1)$ -disconnected sets. *Theoretical Computer Science*, 305(1-3):221–235, 2003. doi:[10.1016/S0304-3975\(02\)00710-7](https://doi.org/10.1016/S0304-3975(02)00710-7). citée pages [20](#), [44](#)
- [141] T. Y. KONG, R. KOPPERMAN et P. R. MEYER : A topological approach to digital topology. *The American Mathematical Monthly*, 98(10):901–917, december 1991. doi:[10.2307/2324147](https://doi.org/10.2307/2324147). citée pages [26](#), [31](#)
- [142] T. Y. KONG, A. W. ROSCOE et A. ROSENFELD : Concepts of digital topology. *Topology and its Applications*, 46(3):219–262, 1992. doi:[10.1016/0166-8641\(92\)90016-S](https://doi.org/10.1016/0166-8641(92)90016-S). citée pages [19](#), [20](#)
- [143] T. Y. KONG et A. ROSENFELD : Digital topology : Introduction and survey. *Computer Vision, Graphics, and Image Processing.*, 48(3):357–393, december 1989. doi:[10.1016/0734-189X\(89\)90147-3](https://doi.org/10.1016/0734-189X(89)90147-3). citée pages [11](#), [14](#), [21](#)
- [144] V. KOVALEVSKY : Finite topology and image analysis. *Advances in Electronics and Electron Physics*, 84:197–259, 1992. citée pages [28](#), [31](#), [46](#), [73](#)
- [145] V. KOVALEVSKY : Multidimensional cell lists for investigating 3-manifolds;. *Discrete Applied Mathematics*, 125(1):25–43, 2003. doi:[10.1016/S0166-218X\(02\)00222-6](https://doi.org/10.1016/S0166-218X(02)00222-6). citée pages [40](#), [46](#), [73](#)
- [146] V. A. KOVALEVSKY : Finite topology as applied to image analysis. *Computer Vision, Graphics, and Image Processing.*, 46(2):141–161, may 1989. doi:[10.1016/0734-189X\(89\)90165-5](https://doi.org/10.1016/0734-189X(89)90165-5). citée pages [11](#), [20](#), [21](#), [22](#), [26](#), [27](#), [28](#), [30](#), [32](#), [39](#), [40](#), [46](#), [49](#), [64](#), [73](#)
- [147] V. A. KOVALEVSKY : Algorithms and data structures for computer topology. In *Digital and Image Geometry, Advanced Lectures [based on a winter school held at Dagstuhl Castle, Germany in December 2000]*, vol. 2243 de *Lecture Notes in Computer Science*, p. 38–58. Springer-Verlag, 2001. Disponible depuis <http://www.springerlink.com/content/51epbktvbx2p/>. citée pages [40](#), [46](#), [73](#)
- [148] V. A. KOVALEVSKY : Axiomatic digital topology. *Journal of Mathematical Imaging and Vision*, 26(1):41–58, 2006. doi:[10.1007/s10851-006-7453-6](https://doi.org/10.1007/s10851-006-7453-6). citée pages [45](#)

- [149] E. H. KRONHEIMER : The topology of digital images. *Topology and its Applications*, 46(3):279–303, 1992. doi:10.1016/0166-8641(92)90019-V. citée pages 11, 45, 46
- [150] W. KROPATSCH et S. BEN YACOB : A revision of pyramid segmentation. In *13th International Conference on Pattern Recognition (ICPR'96)*, vol. 02, p. 477, Los Alamitos, CA, USA, 1996. IEEE Computer Society. doi:10.1109/ICPR.1996.546871. citée pages 73
- [151] W. G. KROPATSCH : Building irregular pyramids by dual graph contraction. Rap. tech. PRIP-TR-35, Dept. for Pattern Recognition and Image Processing, Institute for Automation, Technical University of Vienna, july 1994. citée pages 73
- [152] W. G. KROPATSCH et H. MACHO : Finding the structure of connected components using dual irregular pyramids. In *5th International Conference on Discrete Geometry for Computer Imagery (DGCI'05)*, p. 147–158, september 1995. invited lecture. citée pages 73
- [153] Z. KULPA : On the properties of discrete circles, rings, and disks. *Computer Graphics and Image Process.*, 10:348–365, 1979. citée pages 85
- [154] Y. P. KUZMIN : An efficient circle-drawing algorithm. *Computer Graphics Forum*, 9(4):333–336, 1990. citée pages 84
- [155] J.-O. LACHAUD : Coding cells of digital spaces : a framework to write generic digital topology algorithms. *Electronic Notes in Discrete Mathematics, 9th International Workshop on Combinatorial Image Analysis*, 12:337–348, 2003. doi:10.1016/S1571-0653(04)00497-4. citée pages 40, 73
- [156] J. LAMY : Integrating digital topology in image-processing libraries. *Computer Methods and Programs in Biomedicine*, 85(1):51–58, 2007. doi:10.1016/j.cmpb.2006.08.006. citée pages 40
- [157] L. LATECKI : Topological connectedness and 8-connectedness in digital pictures. *CVGIP : Image Understanding*, 57(2):261–262, march 1993. doi:10.1006/ciun.1993.1017. citée pages 14, 30
- [158] L. LATECKI : Multicolor well-composed pictures. *Pattern Recognition Letters*, 16(4):425–431, 1995. doi:10.1016/0167-8655(94)00104-B. citée pages 21, 30
- [159] L. LATECKI, U. ECKHARDT et A. ROSENFELD : Well-composed sets. *Computer Vision and Image Understanding*, 61(1):70–83, 1995. doi:10.1006/cviu.1995.1006. citée pages 21, 30
- [160] M. LEDOUX et M. TALAGRAND : *Probability in Banach spaces, Isoperimetry and processes*. Springer-Verlag, Berlin, 1991. citée pages 62
- [161] S. LEFSCHETZ : *Topology*, vol. 12. American Mathematical Society Colloquium Publications, New York, 1930. citée pages 32, 33

- [162] E. L. LEHMANN : *Testing statistical hypotheses*. Wiley Series in Probability and Mathematical Statistics : Probability and Mathematical Statistics. John Wiley and Sons, New York, second edition. édn, 1986. citée pages 61
- [163] P. LIENHARDT : Topological models for boundary representation : A survey. research report R 90-02, Université Louis Pasteur, Computer Science dpt, 7, rue René Descartes, 67084 Strasbourg Cedex, february 1990. citée pages 65
- [164] P. LIENHARDT : Topological models for boundary representations : a comparison with n-dimensional generalized maps. *Computer Aided Design*, 23(1):59–82, 1991. doi:10.1016/0010-4485(91)90082-8. citée pages 65
- [165] P. LIENHARDT : *Extensions de la notion de carte et modélisation géométrique ‘à base topologique*. Habilitation à diriger des recherches, Université Louis Pasteur, Strasbourg (France), janvier 1992. citée pages 65
- [166] P. LIENHARDT : Aspects in topology-based geometric modeling. In E. AHRONOVITZ et C. FIORIO, eds : *Discrete Geometry for Computer Imagery*, vol. 1347 de *Lecture Notes in Computer Science*, p. 33–48. Springer, december 1997. (invited speaker). citée pages 65
- [167] R. MALGOUYRES : There is no local characterization of separating and thin objects in z^3 ; . *Theoretical Computer Science*, 163(1-2):303–308, 1996. doi:10.1016/0304-3975(96)00021-7. citée pages 18
- [168] J. MARCHADIER, D. ARQUÉS et S. MICHELIN : Thinning grayscale well-composed images. *Pattern Recogn. Lett.*, 25(5):581–590, 2004. doi:10.1016/j.patrec.2003.12.005. citée pages 40
- [169] D. MARR : *Vision*, p. 31. Freeman, New York, 1982. citée pages 48
- [170] C. MC DIARMID : Concentration. In HABIB, M. DIARMID, RAMIREZ-ALFONSIN et REED, eds : *Probabilistic methods for algorithmic discrete mathematics*, p. 195–248, New-York, 1998. Springer. citée pages 59, 61
- [171] M. D. MCILROY : Best approximate circles on integer grids. *ACM Transactions on Graphics*, 2(4):237–263, october 1983. citée pages 84
- [172] K. MEHLHORN : *Data Structures and Algorithms : Sorting and Searching*. Springer, 1984. citée pages 51
- [173] A. MEIJSTER et M. WILKINSON : A comparison of algorithms for connected set openings and closings. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(4):484–494, april 2002. doi:10.1109/34.993556. citée pages 56
- [174] M. M. MESMOUDI : A simplified recognition algorithm of digital planes pieces. In *10th International Conference on Discrete Geometry for Computer Imagery : (DGCI’02)*, vol. 2301 de *Lecture Notes in Computer Science*, p. 404–416, Bordeaux, France, April 2002. Springer Berlin / Heidelberg. citée pages 90

- [175] O. MONGA : An optimal region growing algorithm for image segmentation. *International Journal of Pattern Recognition and Artificial Intelligence*, 1:351–375, 1987. citée pages 51
- [176] E. F. MOORE : Machine models of self-reproduction. In *Symposia in Applied Mathematics, Proceedings of*, vol. 14, p. 17–33. The American Mathematical Society, 1962. citée pages 22
- [177] D. MORGENTHALER et A. ROSENFELD : Surfaces in three-dimensional digital images. *Information and Control*, 51(3):227–247, december 1981. citée pages 18
- [178] J. MUERLE et D. ALLEN : Experimental evaluation of techniques for automatic segmentation of objects in a complex scene. In G. C. CHENG *et al.*, édés : *Pictorial Pattern Recognition*, p. 3–13. Thompson, Washington, 1968. citée pages 50
- [179] X. MUÑOZ, J. FREIXENET, X. CUFÍ et J. MARTÍ : Strategies for image segmentation combining region and boundary information. *Pattern Recognition Letters*, 24(1-3):375–392, 2003. doi:10.1016/S0167-8655(02)00262-3. citée pages 50, 55, 75
- [180] J. P. MYLOPOULOS et T. PAVLIDIS : On the topological properties of quantized spaces, i. the notion of dimension. *J. of the Association for Computing Machinery*, 18(2):239–246, 1971. doi:10.1145/321637.321644. citée pages 10
- [181] J. P. MYLOPOULOS et T. PAVLIDIS : On the topological properties of quantized spaces, ii. connectivity and order of connectivity. *J. of the Association for Computing Machinery*, 18(2):247–254, 1971. doi:10.1145/321637.321645. citée pages 10
- [182] L. NAJMAN et M. COUPRIE : Building the component tree in quasi-linear time. *Image Processing, IEEE Transactions on*, 15(11):3531–3539, november 2006. doi:10.1109/TIP.2006.877518. citée pages 56
- [183] J. V. NEUMANN : *Theory of Self-Reproducing Automata*. University of Illinois Press, Champaign, IL, USA, 1966. citée pages 22
- [184] F. NIELSEN et R. NOCK : On region merging : the statistical soundness of fast sorting, with applications. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2003)*., 2:II–19–26 vol.2, june 2003. doi:10.1109/CVPR.2003.1211447. citée pages 56, 61
- [185] R. NOCK : Fast and reliable color region merging inspired by decision tree pruning. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001)*, 1:I–271–I–276 vol.1, 2001. doi:10.1109/CVPR.2001.990486. citée pages 56, 61

- [186] R. NOCK et F. NIELSEN : Statistical region merging. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(11):1452–1458, 2004. doi:10.1109/TPAMI.2004.110. citée pages 61
- [187] R. NOCK et F. NIELSEN : Semi-supervised statistical region refinement for color image segmentation. *Pattern Recognition : Image Understanding for Photographs*, 38(6):835–846, 2005. doi:10.1016/j.patcog.2004.11.009. citée pages 56, 61, 75
- [188] N. R. PAL et S. K. PAL : A review on image segmentation techniques. *Pattern Recognition*, 26(9):1277–1294, 1993. doi:10.1016/0031-3203(93)90135-J. citée pages 50, 54
- [189] T. PAVLIDIS : Segmentation of pictures and maps through functional approximation. *Computer Graphics and Image Process.*, 1:360–372, 1972. citée pages 49, 50
- [190] T. PAVLIDIS : *Structural Pattern Recognition*. Springer-Verlag, New York, 1977. citée pages 64
- [191] M. A. PETERSON : Organization, segregation and object recognition. *Intellectica*, 1(28):37–51, 1999. Disponible depuis http://www.intellectica.org/archives/n28/n28_table.htm. citée pages 2
- [192] J. C. PICHEL, D. E. SINGH et F. F. RIVERA : Image segmentation based on merging of sub-optimal segmentations ;. *Pattern Recognition Letters*, 27(10):1105–1116, 2006. doi:10.1016/j.patrec.2005.12.012. citée pages 55
- [193] M. POPOVIC, F. CHANTEMARGUE, R. CANALS et P. BONTON : Several approaches to implement the merging step of the split and merge region segmentation. In F. POST et W. BARTH, eds : *EUROGRAPHICS '91*, p. 399–412. Elsevier Science publishers B.V., 1991. citée pages 57
- [194] T. QUIGER, P. MICHÉ et R. DEBRIE : Segmentation by auto-adaptative thresholding. In *6th International Conference on Image Analysis and Processing (ICIAP'91)*, p. 34–42, Villa Olmo, Como, Italy, 4–6 September 1991. Progress in Image Analysis and Processing II. citée pages 57
- [195] J.-P. REVEILLÈS : *Géométrie discrète, Calcul en nombres entiers et algorithmique*. Thèse de doctorat, Université Louis Pasteur, Strasbourg, France, 1991. citée pages 7, 79, 80, 81, 83, 86
- [196] J. RODRIGUES, W. PUECH et C. FIORIO : Lossless crypto-data hiding in medical images without increasing the original image size. In *2nd International Conference on Advances in Medical Image and Signal Processing (MEDSIP'04)*, p. 358–365, Malta, september 2004. citée pages 8

- [197] A. ROSENFELD : Connectivity in digital pictures. *J. of the Association for Computing Machinery*, 17(1):146–156, 1970. doi:10.1145/321556.321570. citée pages 10, 14, 17, 44
- [198] A. ROSENFELD : Adjacency in digital pictures. *Inform. and Control*, 26(1):24–33, 1974. doi:10.1016/S0019-9958(74)90696-2. citée pages 14, 64
- [199] A. ROSENFELD : Digital topology. *The American Mathematical Monthly*, 86(8):621–630, 1979. doi:10.2307/2321290. citée pages 11, 13
- [200] A. ROSENFELD : Three-dimensional digital topology. *Information and Control*, 50(2):119–127, 1981. doi:10.1016/S0019-9958(81)90177-7. citée pages 15
- [201] A. ROSENFELD, R. HUMMEL et S. ZUCKER : Scene labelling by relaxation operations. *Systems, Man, and Cybernetics, IEEE Transactions on*, 6:420–453, 1976. citée pages 49
- [202] A. ROSENFELD et J. L. PFALTZ : Sequential operations in digital picture processing. *J. of the Association for Computing Machinery*, 13(4):471–494, 1966. doi:10.1145/321356.321357. citée pages 14, 17
- [203] A. ROSENFELD et J. STRONG : A grammar for maps. In J. TOU, éd. : *Software Engineering*, vol. 2. Academic Press, New York, 1971. citée pages 49
- [204] V. ROSENTHAL et Y.-M. VISETTI : Sens et temps de la gestalt. *Intellectica*, 1(28):147–227, 1999. Disponible depuis http://www.intellectica.org/archives/n28/n28_table.htm. citée pages 2
- [205] P. K. SAHA, D. D. MAJUMDER et A. ROSENFELD : Local topological parameters in a tetrahedral representation. *Graphical Models and Image Processing*, 60(6):423–436, 1998. doi:10.1006/gmip.1998.0481. citée pages 29
- [206] P. K. SAHA et A. ROSENFELD : Strongly normal sets of convex polygons or polyhedra. *Pattern Recognition Letters*, 19(12):1119–1124, 1998. doi:10.1016/S0167-8655(98)00088-9. citée pages 29, 30, 46
- [207] P. K. SAHA et A. ROSENFELD : The digital topology of sets of convex voxels;. *Graphical Models*, 62(5):343–352, 2000. doi:10.1006/gmod.2000.0527. citée pages 29, 30
- [208] M. SALOTTI et C. GARBAY : Cooperation between edge detection and region growing : the problem of control. In *Image Processing : Theory and Applications*, p. 95–98. Elsevier Science Publishers, 1993. citée pages 75
- [209] J. SHI et J. MALIK : Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):888–905, august 2000. doi:10.1109/34.868688. citée pages 75, 93

- [210] C. SIMON, G. DAMIAND et P. LIENHARDT : nd generalized map pyramids : definition, representations and basic operations. *Pattern Recognition*, 39(4):527–538, April 2006. doi:10.1016/j.patcog.2005.10.004. citée pages 30, 74, 75
- [211] I. SIVIGNON, F. DUPONT et J.-M. CHASSERY : Decomposition of a three-dimensional discrete object surface into discrete plane pieces. *Algorithmica*, 38(1):25–43, 2003. doi:10.1007/s00453-003-1041-6. citée pages 83, 90
- [212] I. SIVIGNON, F. DUPONT et J.-M. CHASSERY : Digital intersections : minimal carrier, connectivity, and periodicity properties;. *Graphical Models*, 66(4):226–244, 2004. doi:10.1016/j.gmod.2004.05.002. citée pages 18, 83
- [213] S. P. SMITH : Threshold validity for mutual neighborhood clustering. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 15(1):89–92, january 1993. doi:10.1109/34.184777. citée pages 57
- [214] M. B. SMYTH : Semi-metrics, closure spaces and digital topology. *Theoretical Computer Science, Selected Papers of the Workshop on Topology and Completion in Semantics*, 151(1):257–276, 1995. doi:10.1016/0304-3975(95)00053-Y. citée pages 27, 39
- [215] M. B. SMYTH : Region-based discrete geometry. *Journal of Universal Computer Science*, 6(4):447–459, 2000. Disponible depuis http://www.jucs.org/jucs_6_4/region_based_discrete_geometry. citée pages 27
- [216] J. S. STAHL et S. WANG : Globally optimal grouping for symmetric closed boundaries by combining boundary and region information. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(3):395–411, 2008. doi:10.1109/TPAMI.2007.1186. citée pages 55, 75
- [217] K. SUZUKI, I. HORIBA et N. SUGIE : Linear-time connected-component labeling based on sequential local operations. *Computer Vision and Image Understanding*, 89(1):1–23, 2003. doi:10.1016/S1077-3142(02)00030-9. citée pages 56
- [218] R. E. TARJAN : Efficiency of a good but not linear set union algorithm. *J. of the Association for Computing Machinery*, 22:215–225, 1975. citée pages 51
- [219] R. E. TARJAN : A class of algorithms which require nonlinear time to maintain disjoint sets. *Journal of Computer and System Sciences*, 18(2):110–127, 1979. citée pages 51, 52
- [220] R. E. TARJAN et J. van LEEUWEN : Worst-case analysis of set union algorithms. *J. of the Association for Computing Machinery*, 31(2):245–281, 1984. citée pages 51
- [221] A. TORSELLO et E. R. HANCOCK : Graph embedding using tree edit-union. *Pattern Recognition*, 40(5):1393–1405, 2007. doi:10.1016/j.patcog.2006.09.006. citée pages 51

- [222] J. TOUTANT, W. PUECH et C. FIORIO : Minimizing data-hiding noise in color jpeg images by adapting the quantization. In *European Conference on Colour in Graphics, Imaging, and Vision*, Leeds, United Kingdom, june 2006. Disponible depuis <http://www.imaging.org/store/epub.cfm?abstrid=33973>. citée pages 8
- [223] J.-L. TOUTANT : *Modèles d'objets discrets, connexes et séparants, en dimension quelconque*. Thèse de doctorat, Université de Montpellier II Sciences et Techniques du Languedoc, Montpellier, France, december 2007. citée pages 8, 84, 85, 90
- [224] J.-L. TOUTANT, W. PUECH et C. FIORIO : Amélioration de l'invisibilité par adaptation de la quantification aux données à insérer. In *GRETSI'05 : 20eme Colloque sur le Traitement du Signal et des Images*, p. 1193–1196, Louvain-la-Neuve, Belgium, september 2005. citée pages 8
- [225] J.-L. TOUTANT, W. PUECH et C. FIORIO : Asynchronous dct-based data-hiding robust to cropping. In *5th International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS'05)*, april 2005. citée pages 8
- [226] J. VITTONÉ : *Caractérisation et reconnaissance de droites et plans en géométrie discrète*. Thèse de doctorat, Université Joseph Fourier, Grenoble, december 1999. citée pages 90
- [227] J. VITTONÉ et J.-M. CHASSERY : Coexistence of tricubes in digital naive plane. In *International Conference on Discrete Geometry for Computer Imagery (DGCI'97)*, p. 99–110. Springer-Verlag, 1997. citée pages 83, 84, 90
- [228] J. VITTONÉ et J.-M. CHASSERY : (n,m)-cubes and farey nets for naive planes understanding. In *8th International Conference on Discrete Geometry for Computer Imagery (DGCI'99)*, vol. 1568 de *Lecture Notes in Computer Science*, p. 76–87, Marne-la-Vallée, France, march 1999. Springer Berlin / Heidelberg. citée pages 83, 84
- [229] J. VITTONÉ et J.-M. CHASSERY : Recognition of digital naive planes and polyhedrization. In *9th International Conference on Discrete Geometry for Computer Imagery (DGCI'00)*, vol. 1953 de *Lecture Notes in Computer Science*, p. 296–307, Uppsala, Sweden, december 2000. Springer Berlin / Heidelberg. citée pages 90
- [230] Y. WANG et P. BHATTACHARYA : A theory of parameter-dependent connected components of gray images and segmentation. In *International Conference on Image Processing, Proceedings.*, vol. 3, p. 69–72 vol.3. IEEE, 1995. doi:10.1109/ICIP.1995.537582. citée pages 21, 30, 57
- [231] Y. WANG et P. BHATTACHARYA : On parameter-dependent connected components of gray images. *Pattern Recognition*, 29(8):1359–1368, 1996. doi:10.1016/0031-3203(95)00159-X. citée pages 21

- [232] Y. WANG et P. BHATTACHARYA : Digital connectivity and extended well-composed sets for gray images. *Computer Vision and Image Understanding*, 68(3):330–345, 1997. doi:10.1006/cviu.1997.0551. citée pages 21, 30, 31, 36
- [233] R. WATZEL, K. BRAUN, A. HESS, W. ZUSCHRATTER et H. SCHEICH : Restoration of dendrites and spines with the objective of topologically correct segmentation. In *Proceedings of the 13th International Conference on Pattern Recognition (ICPR '96)*, p. 472, Washington, DC, USA, 1996. IEEE Computer Society. doi:10.1109/ICPR.1996.546870. citée pages 57
- [234] M. WILKINSON et J. ROERDINK : Fast morphological attribute operations using tarjan's union-find algorithm. In *Mathematical Morphology and its Applications to Image and Signal Processing*, p. 311–320. Kluwer, 2000. Disponible depuis <http://citeseer.ist.psu.edu/wilkinson00fast.html>. citée pages 56
- [235] G. WINDREICH, N. KIRYATI et G. LOHMANN : Voxel-based surface area estimation : from theory to practice;. *Pattern Recognition*, 36(11):2531–2541, 2003. doi:10.1016/S0031-3203(03)00173-0. citée pages 18
- [236] R. YAPA et K. HARADA : A connected component labelling algorithm for greyscale mammography image processing as a pre-processing tool. *Machine Graphics and Vision*, 16(3-4):305–327, 2007. citée pages 56
- [237] Y. ZENG, D. SAMARAS, W. CHEN et Q. PENG : Topology cuts : A novel min-cut/max-flow algorithm for topology preserving segmentation in n-d images. *Computer Vision and Image Understanding*, 112(1):81–90, 2008. doi:10.1016/j.cviu.2008.07.008. citée pages 93
- [238] H. ZHANG, J. E. FRITTS et S. A. GOLDMAN : Image segmentation evaluation : A survey of unsupervised methods. *Computer Vision and Image Understanding*, 110(2):260–280, 2008. doi:10.1016/j.cviu.2007.08.003. citée pages 55, 61
- [239] S. C. ZHU et A. YUILLE : Region competition : unifying snakes, region growing, and bayes/mdl for multiband image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 18(9):884–900, 1996. doi:10.1109/34.537343. citée pages 57
- [240] S. W. ZUCKER : Survey : Region growing : Childhood and adolescence. *Computer Vision, Graphics, and Image Processing.*, 5:382–399, 1976. citée pages 49, 50, 54
- [241] J. ZUNIC : On the number of digital discs. *Journal of Mathematical Imaging and Vision*, 21(3):199–204, 2004. 10.1023/B:JMIV.0000043736.15525.ed. citée pages 18

Index

- κ -adjacence, 37
- κ -chemin, 16, 37
- κ -connexité, 17, 37
- κ -région, 37, 46
- κ -voisinage, 15
- κ -adjacents, 15
- 18-voisinage, 16
- 26-voisinage, 16
- 4-voisinage, 16
- 6-voisinage, 16
- 8-voisinage, 16

- adhérence, 13
- adjacence, 10
- adjacence entre régions, 38
- affichage (display), 25, 46
- analyse d'images, 48
- anneaux discrets fins, 85
- approche interpixels, 3, 92
- arêtes, 65
- augmentation de la variance intra-région, 59
- axiome de connexité du complémentaire, 43
- axiome de faible localité, 43
- axiome de faible monotonie, 43
- axiome de forte localité, 44
- axiome de l'objet, 43
- axiome du support, 43
- axiomes de séparation, 12

- bien composée, 31
- Block Cells, 64

- block-cell, 73
- bord, 10, 28, 38
- brin, 65
- brins, 65

- carte complète, 70
- carte des bords, 70, 72
- carte lignel, 71
- carte topologique, 70, 93
- cartes combinatoires, 65
 - définition, 65
- cartes généralisées, 65
- cartoon, 25
 - associated cartoon, 25
- Cell (application) , 34
- cell, 34
- cell-list, 73
- cellules, 65
- cercle de Bresenham, 85
- cercle discret analytique, 86
- cercle discret arithmétique, 87
- cercles discrets réguliers, 86
- complexe cellulaire, 22, 23, 27, 45
 - complexe cellulaire abstrait, 27, 31, 32, 46
 - complexe convexe, 32, 33
 - complexe engendré, 33
 - complexe polyédral, 30, 32
 - polyèdre d'un complexe, 33
 - sous-complexe, 33
- composante connexe, 17
- conditionnement, 49

- connexe, 35
- connexité, 10, 17
- COTS. Connected Ordered Topological Space, 22
- COTS : Connected Ordered Topological Space, 22, 23, 27, 28
- courbe, 10
- critère d'homogénéité, 93
- critère de segmentation, 57
- diamond distance, 15
- digital fundamental group, 19
- Digital Picture Space, 19
- digital topology, 11, 13
- discrétisation, 78
- distance, 14
- distance de l'échiquier, 15
- distance de Manhattan, 15
- distribution, 58
- Droite arithmétique, 79
- écran (screen), 25, 46
- ensemble régulier, 25, 46
- espace discret, 14
- espace séparé, 12
- espace topologique, 11
- étoile de α dans O , 42
- étoile étendue de α dans O , 42
- étoile ouverte, 28, 34
- extérieur, 38
- faces, 65
- fermeture, 10, 13
- fermé, 12
- fin, 82
- Find, 52
- fonction d'éclairage fortement locale, 44
- fonction d'Ackerman, 51
- fonction d'éclairage, 40, 42, 43, 46
- fonction épaisseur, 87
- fonctionnalité, 83
- fonctionnalité généralisée, 7, 83, 94
- fonctionnelles, 80
- forte normalité, 29, 30, 46
- fortement locale, 44
- frontière, 12, 13, 38
- Fréchet, 12
- graphe d'adjacence des régions, 49
 - RAG, 49
- graphe topologique des frontières, 67, 93
 - définition, 68
- grid distance, 15
- groupe fondamental, 10, 19
- groupement, 49
- Hausdorff, 12
- homotopie, 10
- Hyperplan arithmétique, 81
- hypersphère arithmétique, 88
- hypersphère discrète arithmétique, 86
- hypersphère discrète arithmétique séparante, 89
- hypersphères discrètes arithmétiques naïve, 89
- hypersphères discrètes arithmétiques standard, 89
- image bien composée, 21, 30
- incidence, 33, 65
- interpixels, 27
- intérieur, 10, 12, 13
- Inverse du paradoxe de Jordan, 18
- inégalité de concentration, 60
- inégalité de concentration de Mc Diarmid, 59
- KKM, 22, 44
- Kolmogorov, 12
- lattice-point distance, 15

- Marr, 48
MergeSquare, 54, 55
Mergesquare, 5
MergeSquares, 93

naïf, 82
naïve, 79
normal, 29
normalité, 29
norme de minimalité, 88
 norme de k minimalité, 88, 94

orbite, 66
ouvert, 11

paradoxe de Jordan, 17, 18
pixel, 14, 15, 53
plan arithmétique, 82
plongement, 67
point frontière, 13
polyominos, 78
proto-arêtes, 20
précode, 67, 69
prédicat d'homogénéité, 50, 54, 61
prédicat d'homogénéité, 6
psychologie Gestalt, 2

RAG
 graphe d'adjacence des régions, 64
reconnaissance, 48
reconstruction, 48
regroupement perceptif, 2
regular DPS, 19
regular set, 25
relation de bornage, 33
région, 37
région homogène, 58

Scanline, 5, 54, 55, 93
screen, 25
 open screen, 25
 pure screen, 25
scène, 25
 scène robuste, 25, 26, 46
scène idéale, 57
segmentation, 48
segmentation en régions, 50
segmentation Union-Find, 54
simplement connexe, 30
sommets, 65
spels, 41
standard, 79, 82
star-topologie, 33
stricte κ -connexité, 17
stricte adjacence, 39, 46
strongly normal DPS, 19
subdivision de l'espace, 65
supercouverture, 26, 45
support, 43, 46
surface, 10

T0, 12
T0-séparé, 24
T1, 12
T2, 12
TGF
 Graphe Topologique des Frontières, 64
Théorème de Jordan, 17, 25
topologie de Moore, 23
topologie de von Neumann, 22
topologie des droites arithmétiques, 80
topologie numérique, 11
topologie-étoile, 4, 27, 34, 46, 65, 92

Union, 52
Union par rang, 52
Union-Find, 5, 50, 51, 68, 93

variation discrète locale, 88, 89, 94
vecteur fonctionnel, 83

voisinage, [12](#)

voisinage $2d$, [16](#)

voisinage $3d$, [16](#)

voisinage dérivé, [45](#)

volumes, [65](#)

voxel, [14](#), [15](#)

well-behaved digital spaces, [20](#)

élément représentatif de l'ensemble, [52](#)

épais, [82](#)

épaisseur, [79](#)

épaisseur variable, [94](#)

étiquetage, [49](#)

Notations

$M = (D, \beta_1, \dots, \beta_n)$	carte combinatoire M , D ensembles de brins, β_1 permutation, $\beta_2 \dots \beta_n$ involutions, 65
$C(\mathbf{o}, r, \omega)$	cercle discret arithmétique de centre $\mathbf{o}$, de rayon r , d'épaisseur arithmétique ω , 87
$\mathcal{C}(\mathcal{K})$	complexe engendré par un ensemble $\mathcal{K}$ de cellules de $\mathcal{C}$., 33
$\Delta_i(R, R')$	augmentation de la variance intra-région de la région résultat de la fusion de R et R' , 58
$\mathbf{DPS}(V, \beta, \omega)$	Digital Picture Space : V ensemble des points de la grille, β et ω relations d'adjacence des points de V ., 19
$\dim_{\mathbb{Q}}\{v_1, v_2\}$	dimension du $\mathbb{Q}$ -espace vectoriel engendré par $\{v_1, v_2\}$, si $\dim_{\mathbb{Q}}\{v_1, v_2\} = 1$ alors il existe un rationnel q tel que v_1 soit combinaison linéaire de V_2 et q ., 79
$\ \mathbf{p} - \mathbf{q}\ _1$	distance de Manhattan entre $\mathbf{p}$ et $\mathbf{q}$, 14
$\ \mathbf{p} - \mathbf{q}\ _{\infty}$	distance de l'échiquier entre $\mathbf{p}$ et $\mathbf{q}$, 14
$\mathfrak{D}(\mathbf{v}, \mu, w)$	droite arithmétique $\mathfrak{D}$, de vecteur normal $\mathbf{v}$, de décalage μ , d'épaisseur de la droite arithmétique w , 79
$(E, \mathcal{O})$	espace topologique E à base d'ouverts $\mathcal{O}$, 11
$E(\cdot)$	Espérance mathématiques, 58
$G(V, D, \alpha, \sigma)$	Graphe topologique des frontières, V ensemble des sommets du graphe, D ensemble des demi-arêtes, σ une involution sur D et α une permutation sur D , 68

$\mathbb{S}(\mathbf{o}, r, \omega)$	hypersphère discrète analytique de centre $\mathbf{o}$, de rayon r , d'épaisseur arithmétique ω , 86
$S(\mathbf{o}, r, (\omega_1, \omega_2))$	hypersphère discrète arithmétique de centre $\mathbf{o}$, de rayon r , de fonctions d'épaisseur ω_1 et ω_2 , 88
$S(\mathbf{o}, r, k, \lambda_1, \lambda_2)$	hypersphère discrète arithmétique séparante relativement à la norme de k -minimalité, de centre $\mathbf{o}$, de rayon r , d'épaisseur normale $\lambda = \lambda_2 - \lambda_1$, 89
I^*	scène idéale ou image idéale, 57
μ	décalage de la droite arithmétique, 79
$\cdot \cdot [k]$	norme de k -minimalité, 88
$\mathfrak{P}(\mathbf{v}, \mu, w)$	plan arithmétique $\mathfrak{P}$, de vecteur normal $\mathbf{v}$, de décalage μ , d'épaisseur de la droite arithmétique w , 81
$\mathbf{p}$	point d'une image, point de $\mathbb{Z}^n$, 14
$\Pr[\Delta_i(R, R') \geq \tau]$	probabilité que l'augmentation de la variance intra-région dépasse le seuil τ , 60
$\langle \mathbf{x}, \mathbf{v} \rangle$	produit scalaire de $\mathbf{x}$ et $\mathbf{v}$, 79
$\mathcal{P}(\mathcal{R}, \mathcal{R}')$	prédicat d'homogénéité, 60
R^*	région idéale de l'image idéale I^* , 57
$St(e, C)$	étoile ouverte de e dans C , cad le plus petit voisinage de e , 28
$st_n(\alpha, O)$	étoile de α dans O , 42
$st_n^*(\alpha, O)$	étoile étendue de α dans O , 42
$supp(O)$	support de O , 42
$\mathcal{V}(\mathbf{e})$	voisinage de $\mathbf{e}$, 12
$\mathcal{V}^*(e)$	voisinage de $\mathbf{e}$ privé de $\mathbf{e}$: $\mathcal{V}^*(e) = \mathcal{V}(e) \setminus \{\mathbf{e}\}$, 12
$\Delta_\lambda f$	variation discrète locale, 88

$\mathbf{v}$	vecteur normal à la droite arithmétique, 79
$\mathcal{V}_4(\mathbf{p})$	4-voisinage de $\mathbf{p}$, 16
$\mathcal{V}_8(\mathbf{p})$	8-voisinage de $\mathbf{p}$, 16
$\mathcal{V}_6(\mathbf{p})$	6-voisinage de $\mathbf{p}$, 16
$\mathcal{V}_{18}(\mathbf{p})$	18-voisinage de $\mathbf{p}$, 16
$\mathcal{V}_{26}(\mathbf{p})$	26-voisinage de $\mathbf{p}$, 16
$\mathcal{V}_{n,\kappa}(\mathbf{p})$	κ -voisinage de $\mathbf{p}$ en dimension n , 15
$\mathcal{V}_{n,\kappa}^*(\mathbf{p})$	κ -voisinage de $\mathbf{p}$ en dimension n privé de $\{\mathbf{p}\}$, 15
$\#\mathcal{V}_{n,\kappa}^*(\mathbf{p})$	nombre de κ -voisins de $\mathbf{p}$ en dimension n , 15
w.l.f.	<i>weak lightning function</i> : fonction d'éclairage faible, 43
w	épaisseur de la droite arithmétique, 79
∂X	frontière de X , 13
X^c	complémentaire de X dans E , 13
$\overline{X}$	fermeture de X , 13
$\overset{\circ}{X}$	intérieur de X , 12

Annexe A

Two linear time Union-Find
stratégies for image
processing [[84](#)]

Two linear time Union–Find strategies for image processing

Christophe Fiorio^{a,*}, Jens Gustedt^{b,1}

^a LIRMM, UMR 9928 Université Montpellier II/CNRS, 161 rue Ada, 34392 Montpellier Cedex 5, France

^b Technische Universität Berlin, Sekr. MA 6-1, Strasse des 17. Juni 136, D-10623 Berlin, Germany

Received June 1994; revised October 1994

Communicated by M. Nivat

Abstract

We consider Union–Find as an appropriate data structure to obtain two linear time algorithms for the segmentation of images. The linearity is obtained by restricting the order in which Union's are performed. For one algorithm the complexity bound is proven by amortizing the Find operations. For the other we use periodic updates to keep the relevant part of our Union–Find-tree of constant height. Both algorithms are generalized and lead to new linear strategies for Union–Find that are neither covered by the algorithm of Gabow and Tarjan (1984) nor by the one of Dillencourt et al. (1992).

1. Introduction and overview

An important problem in image processing is to capture the essential features of a scene. One way to do that is to extract (hopefully) significant regions from the image. The technique for extraction used in this paper is *region growing* first described in [13]. It consists of starting with the smallest regions (i.e. pixels or points of the image) and merging them until they are considered to be optimal. The merging criterion is some oracle that should guarantee the significance of the newly created region. The specification of such oracles is not the subject of this paper – for practical purposes we have chosen some classical threshold function.

As has already been observed by Dillencourt et al. [3], region growing as defined above leads naturally to the *disjoint set union problem*, Union–Find for short. Union–Find in general is not known to have a linear time solution. The best complexity known has been first obtained by an algorithm of Tarjan, see [15], that has been shown to perform in $O(\alpha(n, m)m)$ where α is a very slowly growing function

* Corresponding author. Email: fiorio@lirmm.fr.

¹ Supported by a postdoctoral grant of the Graduiertenkolleg Algorithmische Diskrete Mathematik. Part of this work was done during a visit to the LIRMM of the second author that was funded by the French government. Email: gustedt@math.tu-berlin.de.

and $n < m$ are the amounts of calls to a Union and Find operation, respectively. In [16,17,2] it has been proven that this bound is sharp for some classes of pointer machines and recently this has been generalized to general pointer machines by La Poutré in [6]. Whether or not an algorithm with better complexity on a random access machine might exist is not known until now. If the sequence of Union and Find operations is restricted there are algorithms due to Gabow and Tarjan [4] and Dillencourt et al. [3] that perform in linear time.

Both types of algorithms are not well suited for our purposes: the first – apart from being nonlinear – has tremendous constants of proportionality in the known bound on the complexity; the algorithms of the second type are either too restrictive or do not leave room for generalizations.

In this paper we consider two different variants of Union–Find that solve region growing and then give generalizations of them. For both we give an algorithm that *performs in linear time*. They use classical scanning strategies as used for example in [12] for a preprocessing step. The first algorithm scans the image line by line. For each line, we examine each pixel and we see if we can merge it with the two regions to the left and above. After we have processed a particular line we rescan it in a post-process to maintain our data structures accordingly. The second algorithm, in its recursive variant, assumes that the image is an $(\sqrt{n} \times \sqrt{n})$ -square and proceeds by dividing it into 4 subsquares of size $\sqrt{n}/2 \times \sqrt{n}/2$. After coming up from recursion the regions in the 4 subsquares are merged together along the common boundary; i.e. for every pair of neighboring pixels that belong to different subsquares we perform a Union on the corresponding regions if our decision oracle tells us so. This algorithm leads easily to a parallelization.

The linear time complexity of the first algorithm is due to the fact that we are able to keep the tree of our data structure that is constructed for each region flat. The linearity of the second is proven by amortizing the Find operation. From both it is possible to deduce a generic scheme of algorithms that solve restricted Union–Find's in linear time. The first generalizes to a so-called IntervalUnionFind where the sets that are allowed for Union and Find operations form antichains of an interval order. This scheme is applied to solve a similar problem on planar graphs in linear time, too. The second generalizes to EquilibratedUnionFind where certain restrictions on the size of the sets obtained are required. It leads to linear algorithms for data of higher dimensionality, e.g. spatial bitmaps.

Both algorithms have been implemented for two-dimensional bitmaps. The theoretical efficiency translates very well into *short running times*; in fact we achieve practical *real time interpretation* of the image on today's workstation, and as shown in Fig. 2 the results are well suitable even with the simple oracle chosen.

2. Basics of Union–Find

The general Union–Find problem, or more precisely the disjoint set union problem, can be formulated as follows. Given is a set S , the groundset, of elements, pixels in our

application, that form one-element subsets at the beginning. The goal is to perform arbitrary sequences of Union and Find operations in the best time complexity possible. Here a Union works on two disjoint subsets fusing them into one; a Find identifies the subset a certain element belongs to. For an introduction and overview to Union–Find see e.g. [10]; for recent results see [9].

In the following we will only assume a straightforward implementation of Union–Find that could easily be implemented on an arbitrary pointer machine. In fact there exist versions of Union–Find that are much more sophisticated, see e.g. [15] or [7], that perform in time $O(\alpha(n, m)m)$ and are thus optimal on pointer machines for the general case where no restrictions to the Union’s or Find’s apply. There is also a version that performs in linear time on a special case, first shown to work well when implemented on a random access machine, see [4], and then generalized to pointer machines in [9].

For these algorithms it is necessary to determine a tree of the elements in advance such that all subsets form connected subtrees of that tree at any time of the algorithm. For the application considered here this is not adequate because e.g. the number of pairs of elements that may form two-element subsets would only be $n - 1$ where it should be about $4n$ when considering all neighboring pairs of the matrix.

For our purposes it is sufficient that every set is represented by a rooted tree of the members, the root being the unique representative of the set. This can e.g. be done by giving each element a pointer to another element, the parent in the tree. Find identifies the root of the set by an iterative pointer search. The Union of two sets is done by linking the root of one set to the root of the other one. The choice of which element to link and of which to remain a root will be specified differently for each algorithm.

The cost of both operations, Union and Find, is dominated by the number of pointer jumps of a Find operation. We say that an element has direct access to its region if it is linked directly to the root of the tree.

2.1. Flattening the Union–Find-tree

We give a simple refinement of the Find operation that will be helpful for some special cases; for an example see Fig. 1.

Algorithm 1. FindCompress(p)

- (1) **if** $isTop(p)$ **then return** p
- (2) **else return** $p.parent := FindCompress(p.parent)$

We have

(2.1) After a call FindCompress(p) all elements on the path from p to the root have direct access to the root.

(2.2) FindCompress(p) performs with at most l pointer jumps where l is the length of the path from p to the root.

Suppose now that we have an arbitrary subset S_0 of the groundset such that

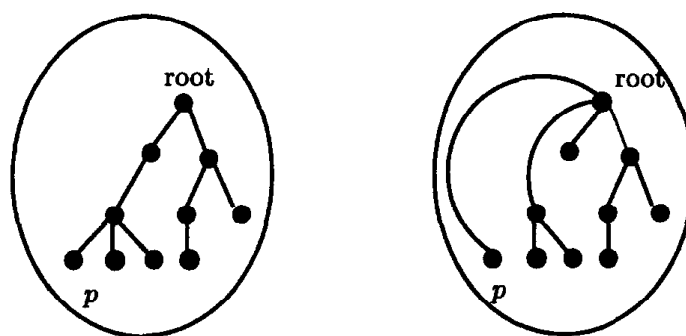


Fig. 1. Updating the representation of a set.

(2.3) every $p \in S_0$ has direct access to the root of its corresponding tree.

Suppose in addition that we perform some arbitrary Find and Union operations exclusively on the set S_0 . Clearly after several Union operations (2.3) might be violated. But then we may perform FindCompress(p) for all $p \in S_0$ which we denote by Flatten(S_0) and we get:

(2.4) After Flatten(S_0) all elements of S_0 have direct access to their region.

(2.5) Flatten(S_0) performs with at most $2|S_0|$ pointer jumps.

3. Image segmentation by merging regions

A major problem in image processing and particularly in scene analysis is to describe information compactly and to capture the essential feature of a scene. An approach is segmentation, e.g. dividing the image into regions (see Fig. 2). Several techniques for image segmentation have been described, see e.g. [5] for an overview. In this paper we are concerned with region growing. This approach searches for areas of the image presenting some homogeneous features.

In Fig. 2 we show two examples of segmented images each produced with the algorithms presented here. The images on the left-hand side show the original images, the middle ones show the borders of the regions obtained and the right ones show the images that result if we replace the original grey tone of each individual pixel with the average value of its particular region.

There are two dual approaches to region segmentation: the split and the merge, see e.g. [14]. In this paper we are working with the merge technique. It consists of starting with the smallest regions (i.e. pixels or points of the image) and merging them until they are optimal. This scheme is also called Region Growing.

The result and the complexity of this grouping depends much on the order in which the merging operations are done. Some criteria we want all grouping strategies to fulfill are the following:

(3.1) Every pair of neighboring pixels should only be considered at least once but at most a bounded number of times.



Fig. 2. Examples of segmentation.

This is to guarantee a linear number of questions to our grouping criteria and to ensure that no artificial borders between objects remain.

(3.2) The size of the regions should be equilibrated during the algorithm.

This is to avoid that certain regions dominate artificially before others had a chance to constitute themselves. This requirement excludes some simple graph searching techniques as e.g. depth-first search.

A commonly used representation is the region adjacency graph (RAG), proposed by Zucker [19]. This representation associates a vertex to each region and links two vertices with an edge if the two corresponding regions are adjacent. So region growing is the process of joining neighboring vertices into one, subject to some conditions as the predicate Oracle. For practical purposes this has the disadvantage that a relatively complicated data structure for the RAG must be maintained. This in general leads to algorithms with nonlinear complexity as in [12]. Our approach is a little different. We consider a region as a set of pixels and instead of grouping two vertices into one, we group them into a set of vertices. Thus we are led to the Union-Find problem.

3.1. Incorporating the oracles

For the overall complexity of a segmentation algorithm that uses Union-Find it will not only be important to perform Union's and Find's efficiently but also to guarantee that the oracle used will increase the complexity only by some factor. For

the simple oracles that we used this is easily achieved; they are threshold functions on

- (1) the absolute difference between the average colors,²
 - (2) the difference between the minimum and maximum color of a potentially created region,
 - (3) the variance of the color values of a potentially created region,
- and any combinations of these. Such oracles can be calculated in constant time per call if at every Union operation the minima, maxima, sums of the color values and sums of the squares of the color values are maintained properly.

4. A line by line strategy

In the following we will describe an algorithm that we denote ScanLine. A similar algorithm for a related problem, namely finding the connected components in a black and white image, also running in linear time has been developed by Dillencourt et al. [3]. Besides that it uses a quiet involved data structure for Union–Find, it does not lead to the same generalization as ours, namely planar graphs, as will be given below.

4.1. Scanning a raster image

ScanLine scans the image line by line and applies Union–Find on the encountered regions. For the following let us assume that the image is a $w \times l$ rectangle.

Algorithm 2. ScanLine

Input: A bitmap bm of size $w \times l$

- (1) *special treatment of the first line*
- (2) **for** $i := 2$ **to** l **do begin**
- (3) *special treatment of the first pixel of line i*
- (4) **for** $j := 2$ **to** w **do begin**
- (5) $left := \text{FindCompress}(bm[i, j - 1]);$
- (6) $up := \text{FindCompress}(bm[i - 1, j]);$
- (7) $this := \text{FindCompress}(bm[i, j]);$
- (8) **if** $\text{Oracle}(left, this)$ **then** $this := \text{Union}(left, this);$
- (9) **if** $\text{Oracle}(up, this)$ **then** $\text{Union}(up, this);$
- (10) **end**
- (11) $\text{Flatten}(\text{line } i)$
- (12) **end**

² All definitions given here are formulated for grey-scaled images. It is easy to see that they can be generalized to real color images when considering e.g. each color plane separately.

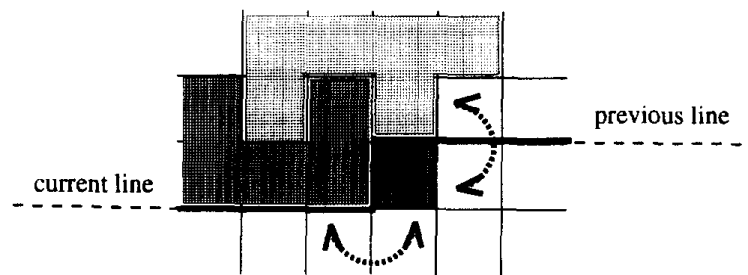


Fig. 3.

At the beginning all the regions consist of exactly one pixel. Initially we process on the first line, then line by line regarding the previous one (see Fig. 3). For each line, we examine each pixel and see if we can merge it with the two regions corresponding to the pixels to the left and above. Clearly in the first line we only deal with the pixel to the left. After we have processed a particular line we rescan it and make a call to Flatten for the set of pixels of this line.

To guarantee the overall complexity, Union links the region that occurred first on the line to the later one. This can easily be realized by a counter that is incremented for each new region. Since Union is done by linking one region to the root of the other we may assume that Union is performed in constant time. As a result we have Theorem 4.1.

Theorem 4.1. *Algorithm ScanLine touches every pair of neighboring pixels and performs in linear time.*

It is easy to see that every pair of pixels is touched. To prove the complexity we need Proposition 4.2 and Lemma 4.3.

Proposition 4.2. *At the beginning of the process on a line, each pixel of the previous line has direct access to its region.*

Proof. This is guaranteed by invariant (2.4) of Flatten. $\square$

With Proposition 4.2 we are able to prove the next lemma:

Lemma 4.3. *For each FindCompress realized when processing a line we have to do at most 4 pointer jumps.*

Proof. At the beginning all the pixels of the line and the previous one have direct access to their regions. When one of the regions consists simply of one pixel and a Union is necessary, we only have to add the pixel to the region. Things get more complicated when we need to realize the Union of two regions each including more than one pixel since the depth of the tree increases.

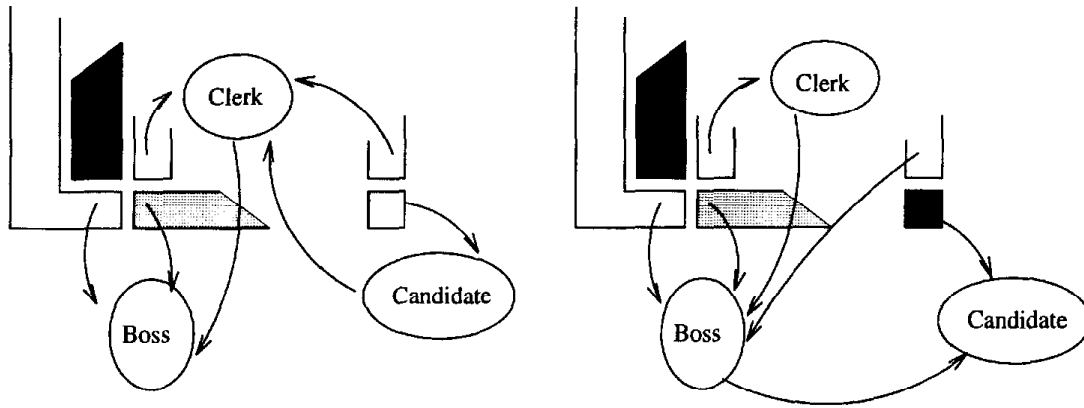


Fig. 4. The two possibilities to have 3 links.

Now we show that we will never meet a pixel which needs more than one extra pointer jump to find the root of its region. Therefore let us suppose that we just merged two regions, denote them *Boss* and *Clerk*. If we have to do a Union between this newly formed region and another one, *Candidate* say, we will risk creating a chain of length 3 in the tree. As we see in Fig. 4, there are only two possibilities³ to do that: either linking *Candidate* directly to *Clerk* or linking the *Boss* directly to *Candidate*.

The first case is impossible. Indeed, before doing a Union we always do a FindCompress and perform a Union operation only on the root of the regions. Since these are *Boss* and *Candidate*, *Clerk* will never be involved directly in such a Union. In the second case all pixels which are linked directly to *Clerk* need 3 pointer jumps to retrieve the root of their region, i.e. *Candidate*. But we will never meet such pixels when continuing on this particular line: since *Boss* was linked to *Candidate*, the later occurred first on the line. Furthermore *Candidate* is connected, so it is surrounding *Boss* and *Clerk*, see Fig. 5. Therefore we will never meet any pixel linked directly to the other two regions. So with invariant (2.2) we have at most 4 pointer jumps for each FindCompress. $\square$

Proof of Theorem 4.1. We will assume that the cost of the algorithm is dominated by the number of pointer jumps. First we scan the line and perform 2 FindCompress's for each pixel: one for the pixel above and one for the pixel to the left. Moreover we perform at most 2 Unions, but these are realized in constant time and do not use pointer jumps.

We will now compute the total number of pointer jumps. Flatten is repeated on each line, so with invariant (2.5) we can compute its total cost: $2 \cdot w \cdot l = 2n$. For each pixel we make at most 2 pointer jumps for each of the two FindCompress's, so in total $4 \cdot w \cdot l = 4n$. Overall the number of pointer jumps is $6n$. $\square$

³ Note that you can reverse *Boss* and *Clerk* in Fig. 4 without changing the argumentation.

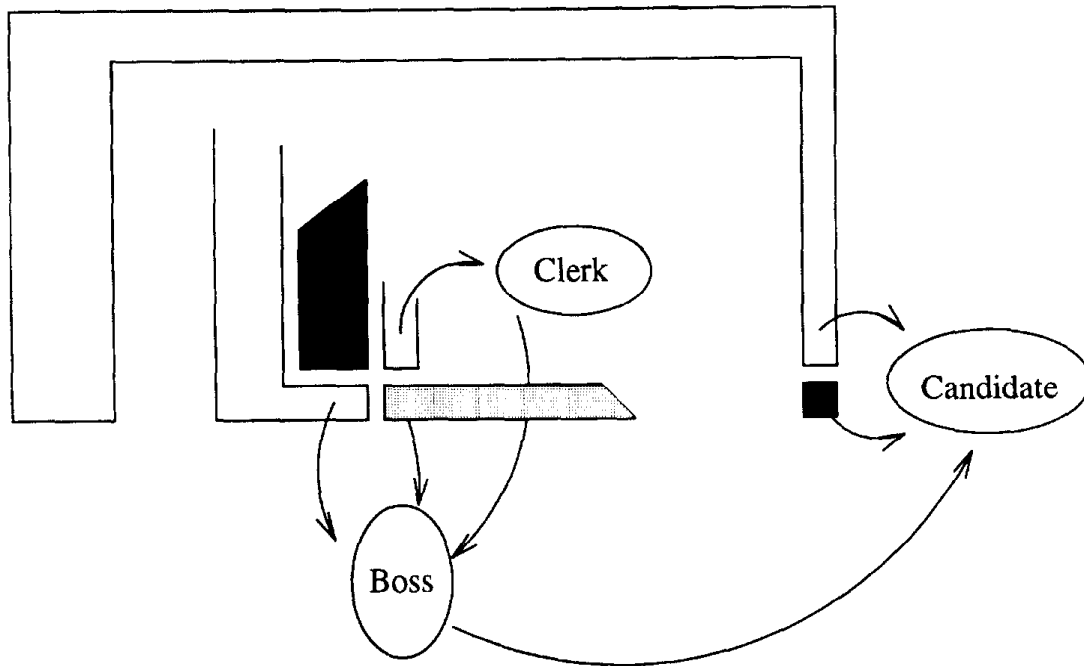


Fig. 5.

4.2. A generalization to planar graphs

In our algorithm it was important that we did not increase the distance of elements still to come on a line from their regions, and that we were able to update the whole line in a second run via a call to Flatten. A generic algorithm IntervalUnionFind that captures these features is

(4.1) **for** $i := 1$ **to** l **do** Generate E_i ; Flatten $X(E_i)$; Process E_i .

Here every E_i is a set of pairs of elements subject to a possible Union and $X(E_i)$ denotes the set of elements involved in these pairs. We require

(4.2) $X(E_i) \cap X(E_j) \subseteq X(E_k)$ for all $i < k < j$ and

(4.3) Process E_i performs Union and Find operations exclusively on the Union-Find sets of the elements in $X(E_i)$.

Observe that (4.2) means that the sets $X(E_i)$ a particular element belongs to appear consecutively. Thus we may associate an interval to each element that represents the period in our algorithm during which we have the right to access it. Invariants (2.4) and (2.5) then translate into:

(4.4) At the beginning of Process E_i in IntervalUnionFind each element of $X(E_i)$ has direct access to the root of its region.

(4.5) The running time for all calls to Flatten in IntervalUnionFind is $O(\sum_{i=1}^l |X(E_i)|)$.

Now suppose we have a planar graph $G = (V, E)$ that is equipped with some data on the vertices and where we want to perform a similar task as segmentation, i.e. where we want to cluster vertices into connected regions according to some

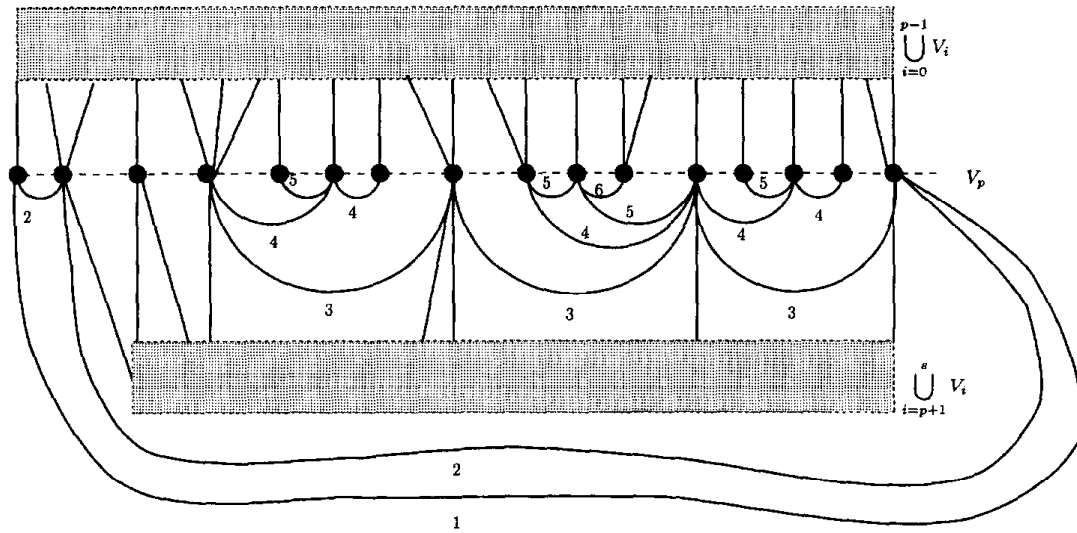


Fig. 6.

homogeneity criterion on the data. We assume that the graph is given together with a combinatorial embedding and has a designated outer face. We may then find a shelling $V_0, \dots, V_s$ of the graph as follows. Let V_0 be some consecutive part of the outer face and $V_p = N_G(V_{p-1}) \setminus \bigcup_{j=1}^{p-1} V_j$, i.e. the sets of equal distance to V_0 . We may assume that the V_p are given as lists in the natural order prescribed by the embedding.

Algorithm 3. PlanarShelling

Input: Planar Graph $G = (V, E)$, shelling $V_0, \dots, V_s$

- (1) **for** $p := 1$ **to begin**
- (2) **for all** $v \in V_p$ **do** Process all edges joining v and $\bigcup_{j=0}^{p-1} V_j$
- (3) Process internal edges of V_p
- (4) **end**

Here phase (2) connects the vertices of V_p to $\bigcup_{j=0}^{p-1} V_j$ in the natural order. This part is a direct generalization of ScanLine to this situation and does not need further explanation. Now in phase (3) we have to process all internal edges of V_p , i.e. that have both endpoints in V_p . Let E_v denote the set of such internal edges with one endpoint being vertex $v \in V_p$, see Fig. 6. We may assume that all vertices of V_p lie on a line, all vertices yet processed lie above that line, and all vertices still to come lie below. Observe also that there are no internal edges crossing that line since everybody is connected to the part above the line, so V_p is outerplanar.

Now let E_1 be the lower cover of the set of internal edges, i.e. that are visible from below. Remove E_1 and obtain a new lower cover E_2 . Repeat this procedure until no internal edges remain and collect the edges in sets $E_1, \dots, E_t$ and let $V(E_i)$ be the set of vertices being endpoint of an edge in E_i .

In the figure the numbers at the edges indicate the set E_i they belong to. The E_i have the following properties:

(4.6) Every E_i is a collection of paths.

(4.7) For every $v \in V_p$ there is an interval $[l_v, r_v]$ s.t. $E_v \cap E_i \neq \emptyset \Leftrightarrow l_v \leq i \leq r_v$.

Clearly the interval in (4.7) may also be empty. Now we may process the internal edges by starting with E_l and proceeding with E_{l-1} and so on.

Algorithm 4. ProcessInternalEdges

Input: Sets $E_1, \dots, E_l$ of edges that fulfill (4.6) and (4.7).

- (1) **for** $i := l$ **downto** 1 **do begin**
- (2) Flatten($V(E_i)$)
- (3) scan E_i from left to right
- (4) **End**

Theorem 4.4. *PlanarShelling runs in linear time.*

Proof. With what is said above it is clear that all phases (2) of PlanarShelling together run in linear time.

Each particular ProcessInternalEdges fulfills the requirements for IntervalUnion-Find and, moreover, the same topological argument as above ensures that we do not have to follow long chains of references to find a root of a particular region. So provided we are able to generate the sets E_i of edges in linear time each such phase also runs in linear time. But this is easy to achieve, since the internal edges of V_p may be seen as a system of parentheses and the levels edges belong to can be found by a scan from left to right. $\square$

5. A divide and conquer strategy

Now we are going to present an algorithm that will also perform in linear time, but has the additional feature that it allows a straightforward parallelization. To get a good upper bound of its complexity it will be necessary to amortize the Find operation over the complete run of the algorithm; the number of pointer jumps for a particular Find might well be logarithmic and not constant any more. To achieve logarithmic time for every Find we use a variant of the Union operation, the so-called weighted union rule, that always links the smaller region to the larger one. Because of that choice we have the following invariant, see e.g. [10], that we will need later:

(5.1) Every Find operation can be done with $\log s$ pointer jumps where s is the cardinality of the set in question.

5.1. The recursive algorithm

For the following algorithm we assume that the image is an $(\sqrt{n} \times \sqrt{n})$ -square, $\sqrt{n}$ a power of 2, and proceed recursively by dividing it into 4 subsquares of size $\sqrt{n}/2 \times \sqrt{n}/2$. After coming up from recursion the regions in the 4 subsquares are

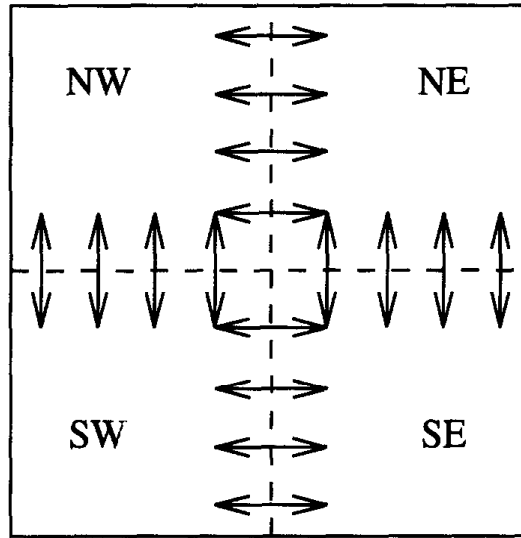


Fig. 7.

merged together along the common boundary; i.e. for every pair of neighboring pixels that belong to different subsquares we perform a Union on the corresponding regions if our oracle tells us so.

For the following formulation of the algorithm we assume that we have easy access to the four subsquares of our bitmap (see Fig. 7). If the bitmap is called bm we denote by $bm[NW]$ the northwestern submatrix, by $bm[NE]$ the northeastern, etc.

Algorithm 5. MergeSquares

Input: An integer k and a bitmap bm of size $2^k \times 2^k$

- (1) **if** $k = 0$ **then return;**
- (2) $h^- := 2^{k-1} - 1$; $h^+ := 2^{k-1}$;
- (3) **for** $DIR := NW$ **to** SE **do** MergeSquares($bm[DIR]$, $k - 1$);
- (4) **for** $i := 0$ **to** 2^k **do begin**
- (5) $left := Find(bm[i, h^-])$; $right := Find(bm[i, h^+])$;
- (6) **if** Oracle($left$, $right$) **then** Union($left$, $right$);
- (7) **end**
- (8) **for** $i := 0$ **to** 2^k **do begin**
- (9) $up := Find(bm[h^-, i])$; $down := Find(bm[h^+, i])$;
- (10) **if** Oracle(up , $down$) **then** Union(up , $down$);
- (11) **end**

Theorem 5.1. *MergeSquares touches all neighboring pairs of pixels of the bitmap and performs in total in linear time.*

Proof. It is easy to see that MergeSquares visits all neighboring pairs exactly once. For the complexity let us analyze a call to MergeSquares for the size of the square being $2^k \times 2^k$. We have 4 recursive calls and $2 \times 2^k = 2^{k+1}$ possible merging operations.

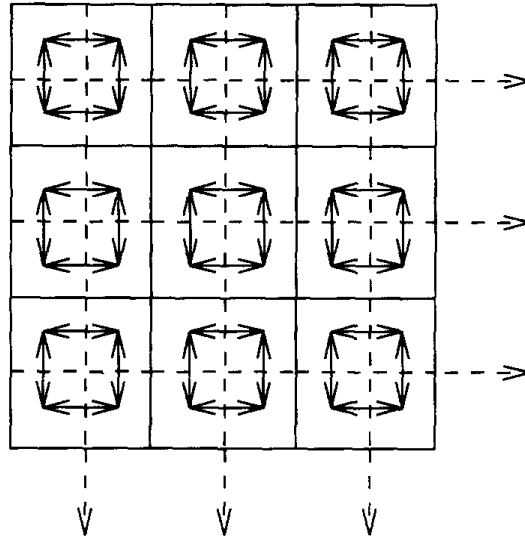


Fig. 8.

We will assume that the cost of such a merging operation is dominated by the length of the pointer jumps to perform when looking for the roots of the corresponding sets via two Find's.

The regions that might be merged together have a size bounded by $2^k \times 2^k$. So by (5.1) we know that each such merging operation needs at most $2 \log 2^{2^k} = 4k$ pointer jumps. So in total we perform with at most $8k2^k$ pointer jumps.

If we fix k for a moment, this is done $n/(2^{2^k})$ times for subsquares of size $2^k \times 2^k$. So for all such subsquares we need at most

$$(5.2) \quad 8k2^k n / (2^{2^k}) = 8(k/2^k)n$$

pointer jumps. In total all pointer jumps are now bounded by

$$(5.3) \quad \sum_{k=1}^{\log \sqrt{n}} 8(k/2^k)n < 8n \sum_{k=1}^{\infty} k \frac{1}{2^k}.$$

For $-1 < x < 1$ we have the well-known identity

$$(5.4) \quad \frac{x}{(1-x)^2} = \sum_{k=1}^{\infty} kx^k.$$

This can e.g. easily be seen when expanding the function $x/(1-x)^2$ in a Taylor series at 0. Thus the right-hand side of (5.3) evaluates to $16n$ which is linear in the size n of our bitmap. $\square$

5.2. An iterative formulation

If we look at all merging operations that are done on a specific recursion level l we see a characteristic pattern (see Fig. 8). That is if we cover the whole bitmap with all $2^l \times 2^l$ subsquares we see that the pairs of pixels that are possibly subject to a Union

operation on level l are found in only some specific rows and columns; it is easy to see that all Unions to do on level l are for pairs of the form $((i, j), (i + 1, j))$ where $i = r \cdot 2^l + 2^{l-1}$ for $r = 0, \dots, \sqrt{n}/2^l$ and $j = 1, \dots, n$ and of the form $((i, j), (i, j + 1))$ where $i = 1, \dots, n$ and $j = r \cdot 2^l + 2^{l-1}$ for $r = 0, \dots, \sqrt{n}/2^l$. So we may perform all these Unions in an arbitrary order. In fact by starting with level $l = 1$ we may proceed iteratively and obtain an algorithm that works on arbitrary rectangular bitmaps.

5.3. An adequate memory mapping for parallelization

For an efficient processing of the image it might be important to parallelize the algorithm MergeSquares. From the definition of the algorithm we get easily

Proposition 5.2. *MergeSquares can be implemented efficiently on a PRAM with $p < 2^k$ processors such that it runs on a bitmap of size $2^k \times 2^k$ in $O(2^{2k}/p)$ time.*

To implement MergeSquares efficiently it is important that

- (1) we do not copy the data needed for a recursive call,
- (2) a routine on a lower level of recursion should not have to take care of the global size of the data, and
- (3) we map the matrix linearly into memory such that every call has its data in a consecutive part.

Then all recursive calls can be performed completely independent of each other.

This can be realized by a recursive “quad tree” definition of the matrix: first mapping submatrix NE , then NW , etc., see above. What we get in fact by such a recursive definition is that pixel (i, j) is mapped in position $i_{k-1}j_{k-1} \dots i_0j_0$ if the binary representations of i and j are $i_{k-1} \dots i_0$ and $j_{k-1} \dots j_0$, respectively, i.e. the index of a particular matrix element in the linear memory is given by alternating the bits of i and j . For an example of a (4×4) -matrix A mapping into a vector v of length 16 consider Fig. 9. Here we map e.g. the element $(2, 2) = (10b, 10b)$ of matrix A (indices starting with 0) onto the element $1100b = 12$ of vector v . Now some juggling with bit-operations makes it easy to perform the merging phase of the algorithm sequentially.

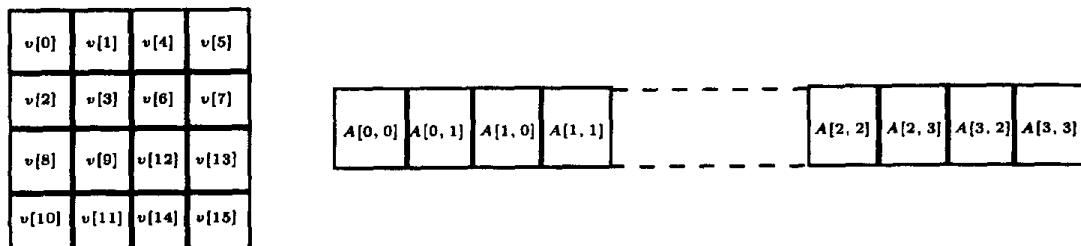


Fig. 9. Mapping a matrix to a vector.

5.4. A generalization to higher dimensions

For MergeSquares it was important that we had the guarantee that the regions would not grow too fast, and that the number of Find's to perform was small compared to the possible size of the regions. This is best generalized by introducing a logarithmic number of phases. If each phase i guarantees

(1) no subset is larger than $O(2^i)$ and

(2) no phase makes more than $O(2^{\log n - i})$ Find operations,

then the complexity is bounded by

$$(5.5) \quad O\left(\sum_{i=1}^{\log n} \log 2^i \cdot 2^{\log n - i}\right) = O\left(\sum_{i=1}^{\log n} i \cdot 2^{-i} \cdot n\right) = O(n).$$

We call such a strategy EquilibratedUnionFind. One application of Equilibrated-UnionFind could be the case where the elements are considered to be vertices of a planar graph G of bounded maximum degree. Then balanced separator techniques could be used to obtain again a linear time Union-Find strategy. We do not go into further details since this problem is already covered by PlanarShelling. Another more important application is to bitmaps of higher dimensions, e.g. spatial data. If we denote the natural generalization of MergeSquares to dimension d by MergeOctants _{d} we easily get

Theorem 5.3. *Let d be some fixed dimension, then MergeOctants _{d} runs in linear time.*

6. Notes on implementation

Both algorithms have been implemented straightforwardly in C^{++} . They show suprisingly good results both in the quality of the segmentation as well as in time performance. Compiled with $g++$, the C^{++} compiler of the GNU project, we achieve a running time of about 12 μ s per pixel. For example, for a small image with 256×256 pixels like the boat in Fig. 2 we had a processing time of 0.8 s. Certainly these times will improve when the implementation becomes more sophisticated or if the algorithms are realized on an appropriate hardware.

Even more surprising for the authors than the running time has been the quality of the segmentation. In order to reduce the data to be considered both algorithms originally were thought to form a preprocessing step to some other treatment. But seeing that the output is already competitive we believe now that they can be immediately followed by an interpretation step that tries e.g. to group regions into objects. One indication that our approach reaches the limits of what can be achieved with segmentation by itself is that iterating the algorithms does not change the picture very much.

For example, if we apply the divide and conquer algorithm on the boat in Fig. 2 several times we obtain the regions shown in Fig. 10. On the left we see what is given



Fig. 10. Three iterations of the segmentation algorithm.

by the first iteration (3223 regions), in the middle the second (2622 regions) and on the right the fourth (2574 regions). Then in any further iteration no additional improvement is made and the situation stabilizes with that number of regions. So the most important reduction of the complexity from 65 536 regions (=number of pixels) to 3223 is already done in the first iteration.

References

- [1] A. Aggarwal et al., eds., *Proc. 1st Ann. ACM–SIAM Symp. on Discrete Algorithms* (SIAM, Philadelphia, PA, 1990).
- [2] L. Banachowski, A complement to Tarjan's result about the lower bound on the complexity of the set union problem, *Inform. Process. Lett.* **11** (1980) 59–65.
- [3] M.B. Dillencourt, H. Samet and M. Tamminen, A general approach to connected-component labeling for arbitrary image representations, *J. ACM* **39** (1992) 253–280, Corr. pp. 985–986.
- [4] H.N. Gabow and R.E. Tarjan, A linear-time algorithm for a special case of disjoint set union, *J. Comput. System Sci.* **30** (1984) 209–221.
- [5] R.M. Haralick and L.G. Shapiro, Survey: image segmentation techniques, *Comput. Vision Graphics Image Process.* **29** (1985) 100–132.
- [6] J.A. La Poutré, Lower bounds for the union–find and the split–find problem on pointer machines, RUU-CS-89-21, 1989.
- [7] J.A. La Poutré, New techniques for the union–find problem, RUU-CS-89-19, 1989.
- [8] J.A. La Poutré, New techniques for the union–find problem, [1] (1990) 54–63, extended abstract of [7].
- [9] J.A. La Poutré, Dynamic graph algorithms and data structures, Ph.D. Thesis, Rijksuniversiteit te Utrecht, 1991.
- [10] K. Mehlhorn, *Data Structures and Algorithms 1: Sorting and Searching* (Springer, Berlin, 1984).
- [11] K. Mehlhorn and A. Tsakalidis, Data structures, in [18], Ch. 6, pp. 301–314.
- [12] O. Monga and B. Wrobel-Dautcourt, Segmentation d'images: vers une méthodologie, *Traitement du Signal* **4** (1987) 169–193.
- [13] J.L. Muerle and D.C. Allen, Experimental evaluation of techniques for automatic segmentation of objects in a complex scene, in: G.C. Cheng et al., eds., *Pictorial Pattern Recognition* (Thompson, Washington, 1968) 3–13.
- [14] M. Popovic, F. Chantemargue, R. Canals and P. Bonton, Several approaches to implement the merging step of the split and merge region segmentation, in: F.H. Post and W. Barth, eds., *EURO-GRAPHICS'91* (Elsevier, Amsterdam, 1991) 399–412.
- [15] R.E. Tarjan, Efficiency of a good but not linear set union algorithm *J. ACM* **22** (1975) 215–225.

- [16] R.E. Tarjan, A class of algorithms which require non-linear time to maintain disjoint sets, *J. Comput. System Sci.* **18** (1979) 110–127.
- [17] R.E. Tarjan and J. van Leeuwen, Worst-case analysis of set union algorithms, *J. ACM* **31** (1984) 245–281.
- [18] J. van Leeuwen, ed., *Handbook of Theoretical Computer Science, Vol. A: Algorithms and Complexity* (Elsevier, Amsterdam, 1990).
- [19] S.W. Zucker, Survey: region growing: childhood and adolescence, *Comput. Graphics Image Process.* **5** (1976) 382–399.

Annexe B

Segmentation en régions avec critère statistique basée sur les inégalités de concentration [[89](#), [90](#)]

Cette annexe présente deux papiers publiés en conférence, le premier [[89](#)] présentant le critère statistique d'homogénéité, le second [[90](#)] présentant l'algorithme de segmentation.

A Concentration-Based Adaptive Approach to Region Merging of Optimal Time and Space Complexities

Christophe Fiorio
LIRMM
161, rue Ada
34392 Montpellier, France
fiorio@lirmm.fr

Richard Nock
Université des Antilles-Guyane
UFR SEN-Dept of Maths and CS
97159 Pointe-A-Pitre Cedex, France
rnock@univ-ag.fr

Abstract

In this paper, we investigate image segmentation as a statistical and computational problem. The observed image is sampled from a theoretical, unknown image, in which pixels are represented by distributions. Our objective is to approximate as best as possible the region segmentation in the ideal image, where each region has pixels with identical expectations, but adjacent regions have different pixel's expectations. From that model, a concentration-based statistical test for deciding region merging is built, limiting the risk of wrong merges. The analysis is carried out without any assumption on the distributions: we avoid in particular the classics of variance analysis, normality and homocedasticity. A practical approximation of the test is given, of constant time and space computation, which leads in turn to a segmentation algorithm of optimal complexity, easy to implement. Some experiments on various types of images shed light on the quality of the segmentations obtained.

1 Introduction

In the early stage of a vision process it is necessary to isolate “objects” in the scene before recognising them, a step called *image segmentation*. In this paper, we investigate image segmentation by region merging, which consists in building up regions by combining smaller regions, pixels being taken as primary regions. Our ultimate goal is to obtain a good partition of an image I into regions $\{R_1, R_2, \dots, R_n\} = P(I)$, each region being 4-connected: between any of its pixels, a path exists, lying in the region, with horizontal or vertical connections.

In the field of image segmentation, many algorithms use statistical or mathematical techniques, models, relying on hypotheses made *a priori* on the image [7, 1, 6]. Many of these hypotheses constrain the models by introducing distributional assumptions on the image. Also, the computational costs can be huge. Other approaches focus rather on topologic and algorithmic properties of the image [4, 3], striving to find fast segmentation algorithms. However, statistical confidence in the results is generally not ensured. In that paper, we are interested in a bridging technique between the two approaches, based on a weakly constrained statistical model of image generation, which ultimately leads to a

reliable segmentation algorithm of optimal complexity.

Our model supports the claim according to which an observed image I is principally a snapshot of an ideal object “scene” I^* , taken under particular conditions. Formally, in I^* , pixels are represented by distributions, used to generate the observed image. Ideal regions satisfy the homogeneity criterion that pixel’s expectations are equal, expectations that are in turn different between adjacent regions. This model gives an intuitive notion of what is the *optimal segmentation*. This could be obtained by infinite sampling of the distributions, thus under infinitely many “conditions of observations”. Averaging over the observed images would produce for each region its theoretical grey-level expectation, and we would observe the exact frontiers between regions. Optimising segmentation to recognise regions of I^* may be of great practical interest, in particular when processing numerous images of similar scenes, such as for real-time segmentation. In that case, we better look for fast results of overall quality, rather than single image optimisation at the possible expense of time complexity.

In the following section, we present our model of image generation. Then, we propose a statistical merging test to recognise if two observed, adjacent regions, belong actually to the same object in I^* . It has the major computational feature of being constant time and space approximable. It has the major statistical feature of being completely adaptive, and relies on a concentration bound whose proof sketch is given. Finally, we provide a segmentation algorithm of optimal complexity implementing this test, along with experiments that were conducted.

2 A Model for Image Generation in I^*

Let I denote the image observed, containing $|I|$ pixels ($|\cdot|$ denotes the cardinality). I is described over a maximum, theoretical number, of g grey levels (generally, $g = 256$), but effectively contains g_I observed grey-levels. The model easily endorses RGB coding, by considering three separate models in the same way. I is an observation of a perfect scene I^* we do not know of, in which pixels are perfectly represented by a family of *distributions*, from which the observed grey-level is sampled. In all that follows, the “*” superscript denotes objects taken from I^* , such as R^* for a region of I^* , to which corresponds a region R of I . The j^{th} pixel in the i^{th} region R_i of the image I , $p_{i,j}$, is obtained by the outcome of some random variable (r.v.) $p_{i,j}^* \in I^*$, itself sum of Q independent r.v. $p_{i,j,1}^*, p_{i,j,2}^*, \dots, p_{i,j,Q}^*$ of I^* (each of these take values in the set $[0, g/Q]$, see Figure 1). Also, pixels are supposed sampled independently from each other. The role of Q is mainly to ensure the practical tractability of our merging test. Actually, as we shall see, the bigger the image I and the regions tested, the smaller Q can be chosen. As outlined in the introduction, each region in I^* satisfies an homogeneity property, formalised as follows:

- First, $\forall i \in \{1, 2, \dots, |I|\}, \forall j \in \{1, 2, \dots, |R_i^*|\}, \mathbf{E}(p_{i,j}^*) = \mathbf{E}(R_i^*)$ (expectations of all pixels in R_i^* are equal).
- Second, any couple of adjacent regions (R_i^*, R_j^*) satisfies $\mathbf{E}(R_i^*) \neq \mathbf{E}(R_j^*)$.

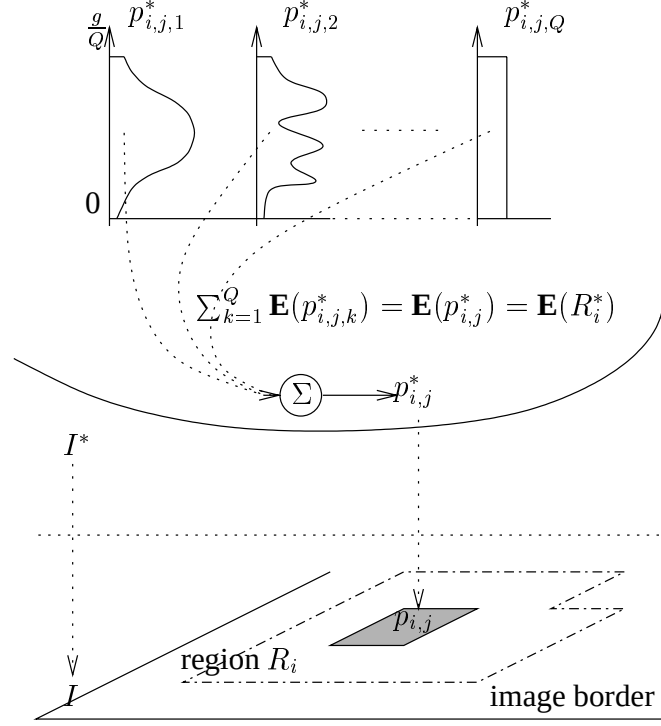


Figure 1: One pixel in I^* and I ($\mathbf{E}(\cdot)$ denotes expectation).

3 A statistical merging test in I

The test relies on some definitions related to ordinary variance analysis. In contrast however, our study does not rely on assumptions such as normality or homocedasticity, *i.e.* equal standard deviation between pixels in I^* . Though we do not observe it directly in I , we denote as $p_{i,j,k}$ the outcome of $p_{i,j,k}^*$, *i.e.* $p_{i,j} = \sum_{k=1}^Q p_{i,j,k}$. In all that follows, the bar notation “ $\bar{R}$ ” denotes the *observed grey-level average of some region*, that is, $(\sum_{j,k} p_{i,j,k})/|R_i|$. Similarly, the average grey-level of the observed image I is denoted $\bar{I}$.

Definition 1 Let $\{R_1, R_2, \dots, R_n\} = P(I)$ be a partition of I .

- The Intra-Pixel Variability (IPV) in I , is

$$V_P(I, P(I)) = \sum_{i,j,k} \left(p_{i,j,k} - \frac{p_{i,j}}{Q} \right)^2$$

- The Intra-Region Variability (IRV) in I , is

$$V_i(I, P(I)) = \frac{1}{Q} \sum_{i,j} (p_{i,j} - \bar{R}_i)^2$$

- The Extra-Region Variability (ERV) in I , is

$$V_e(I, P(I)) = \frac{1}{Q} \sum_i |R_i| (\bar{R}_i - \bar{I})^2$$

- The Overall Image Variability of I is

$$V(I) = \sum_{i,j,k} \left(p_{i,j,k} - \frac{\bar{I}}{Q} \right)^2$$

Some of the formulae presented before might appear misleading with respect to classical notations of variance analysis, but we have given them in the most compact form. For example, in order to follow the usual conventions, the IRV should be stated

$$V_i(I, P(I)) = \sum_{i,j} Q \left(\frac{\sum_k p_{i,j,k}}{Q} - \frac{\sum_{j,k} p_{i,j,k}}{Q|R_i|} \right)^2$$

This strictly represents the same fomula as in definition 1. Also, classical result of variance analysis is preserved since we have

$$V(I) = V_p(I, P(I)) + V_i(I, P(I)) + V_e(I, P(I)) \quad (1)$$

(proof straightforward). The membership to some region $R^* \subseteq I^*$ of two adjacent regions R and R' of I is checked by a measure of their “proximity”, the increase of IRV when merging them, $\Delta_i(R, R')$ (In RGB coding, it is the sum of three IRV increases in each of the R, G and B coordinates, and each pixel results from the outcomes of $3Q$ distributions). Though we do not have access to any $p_{i,j,k}$, the fact that pixels are themselves “regions” of fixed size Q makes that $V_p(I, P(I))$ is constant, and thus does not influence $\Delta_i(R, R')$. The following lemma shows that the computation of $\Delta_i(R, R')$ is indeed easy (proof straightforward).

Lemma 1 *The increase of IRV when merging R and R' satisfies*

$$\Delta_i(R, R') = \frac{|R| \times |R'|}{Q(|R| + |R'|)} (\bar{R} - \bar{R}')^2 \quad (2)$$

Therefore, as long as we merge regions in the image, $V_i(I, P(I))$ increases by *local* amounts, which can be computed in constant time, as well as for the parameter’s update (means and sizes). The deviations of $\Delta_i(R, R')$, when R and R' come from the same region in I^* , can be quantified as follows ($\#R$ is the number of distinct regions of size $|R|$, having grey levels included into those of R):

Theorem 1 *Let R, R' be two adjacent regions in I , belonging to the same region R^* in I^* . Fix*

$$m = \frac{6\sqrt{2}g^2 \sqrt{\max\{|R|, |R'|\}(|R|(|R| + 2|R'|)^2 + |R'|^3)}}{Q^3|R||R'|(|R| + |R'|)}$$

Suppose that $g, Q, |R|, |R'|$ are such that $m < 1/2$. Fix some $\tau > 0$. Then,

$$\Pr[\Delta_i(R, R') \geq \tau] < \frac{8\#R\#R'}{1 - 2m} \times e^{-\frac{2Q^2(|R|+|R'|)^2\tau}{g^2|R||R'|}} \quad (3)$$

Proof sketch: Though concentration bounds are not so hard to obtain provided unrealistic hypotheses are assumed, the necessity for us to stay as close as possible of a practical model of image generation led us to build a rather long proof for our bound. Moreover, ordinary concentration bounds, in that case, generally lead to too large bounds for practical use: in practice indeed, and we have observed it, this can lead to an over merged image with too few regions. That is why we have developped a bound tailor-made for our model, whose proof steps are the following ones. First, lemma 1 is used to shift from variability analyses to the deviation analysis of $|\overline{R} - \overline{R}'|$ beyond some τ' , for some *fixed* regions R and R' . This is of better easiness to upperbound. Second, we remark that

$$(\overline{R \cup R'} - \overline{R}^*)^2 = (|\overline{R} - \overline{R}'|)^2 + Z \quad (4)$$

with R^* being the region of I^* to which R and R' belong, and Z defining the outcome of a r.v. . In other words,

$$\Pr[|\overline{R} - \overline{R}'| \geq \sqrt{\tau'}] < \Pr[(\overline{R \cup R'} - \overline{R}^*)^2 \geq \tau'] / \Pr[Z \geq 0] \quad (5)$$

$$= \Pr[\mathbf{B}] / \Pr[\mathbf{C}] \quad (6)$$

Then, a lowerbound $\Pr[\mathbf{C}] \geq \frac{1}{2} - m$ is obtained, where m is the median of Z (Z has 0 expectation). A concentration upperbound for the median m is obtained by combining the use of the Independent Bounded Difference Inequality (IBDI, [5]) and a derived version of Talagrand's theorem [5]. Finally, using the IBDI on $\Pr[\mathbf{B}]$ gives the final result of theorem 1. Since the result holds for two *fixed* regions R and R' , the multiplication of the bound by the number of possible choices for R and R' gives the final, desired result. $\square$

Theorem 1 can be used to build a merging threshold τ for $\Delta_i(R, R')$, under which we can expect R and R' to come from the same region in I^* . This is done by solving the right inequality of theorem 1 for some particular probability. The merging test for two adjacent regions R, R' is then straightforward: “if $\Delta_i(R, R')$ does not exceed τ , then merge R and R' ”.

4 Practical approximations and experimental results

Theorem 1 gives a way to practically quantify the risk we make of rejecting the merging of two regions R and R' that would actually be elements of the same region R^* in I^* . Of course, when executing n' rejections $(R_1, R'_1), (R_2, R'_2), \dots, (R_{n'}, R'_{n'})$, the probability that the event of theorem 1 occurs can be upperbounded by n' times the upperbound given. Solving for τ the equality between the right member of Ineq. 3 and δ/n' gives the theoretical testing threshold, which is conveniently approximated for practical computability as follows. First, the segmentation algorithm we use is optimal but precludes the knowledge of n' in advance. We replace therefore n' by an upperbound for our segmentation algorithm, $2|I|$. n' appearing in a log factor for τ , this gives no visual difference in the results. Let

$$\#R = g_I^{|R|} / U_R \quad (7)$$

where U_R is a correction factor:

- $U_R = 0.5 * \frac{|R|!}{(|R|/2)!}$ if $|R| < g_I$,
- $U_R = 0.5 * \frac{|R|!}{(|R|-(g_I/2))!}$ otherwise (proof omitted due to the lack of space)

Factorials are approximated using Stirling's formula: $x! \approx \sqrt{2\pi x}(x/e)^x$. All tests are processed with reasonable values of Q, m, δ :

$$Q = \sqrt{g} \quad (8)$$

$$m = 1/4 \quad (9)$$

$$\delta = 1/|I| \quad (10)$$

While Q remains sufficiently small, this limits the risk of overmerging for significant, “visible” regions. The threshold τ , computable in constant time, is finally:

$$\tau(R, R') = \frac{g|R||R'|}{2(|R| + |R'|)^2} \log \left(\frac{32|I|^2 g_I^{(|R|+|R'|)}}{U_R U_{R'}} \right) \quad (11)$$

Finally, each image is preprocessed in linear time and space to merge adjacent pixels having the same grey-level: this does not degrade optimal solutions, but increases the average region's size, and thus the efficiency of the merging test. The segmentation algorithm consists in making a single pass of the image, and testing (if possible) the merging of the current pixel (or region) with the upside pixel's region, and then the left one. As showed in [3], a particular data structure allows to obtain optimal (linear) time and space complexities. The running time for the C implementation over a Pentium II PC never exceeded the second for every experiment.

Figures 2, 3 and 4 present some experiments on various images. We emphasise that *all* segmentations were obtained using the *same* tuning for all parameters, even if the images are much different from each other (synthetic, medical, aerial images, portraits, etc.). In particular, the quality of the results does not stem from the optimisation of the algorithm on each image. Also, images are segmented without any preprocessing (e.g. noise filtering). Only **street** was Gaussian smoothed according to [2]'s experimental setup, to make accurate comparisons.

In **cornouaille**, the algorithm manages a great reduction in the boat's noise, particularly on its left part, whereas it keeps almost intact the overall shape and the name of the boat (note also the segmentation of the man in the bottom-right part). In image **synth**, which mainly features a ramp gradient on the left, and a noisy rectangle area on the right, the algorithm detects exactly three regions, approximating those most people would consider as perceptually distinct. The **street** image shows that our algorithm outperforms two other approaches on two aspects: the number of regions selected compared to [2] (e.g. the van), and the accuracy of the segmentation compared to [3] (e.g. the grass, highly noisy).

References

- [1] S. Dugelay, C. Graffigne, and J.-M. Augustin. Segmentation of multibeam acoustic imagery in the exploration of the deep-sea bottom. In *Int. Conf. on Pattern Recognition*, pages 437–446, 1996.

- [2] P. F. Felzenszwalb and D. P. Huttenlocher. Image segmentation using local variations. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pages 98–104, 1998.
- [3] Christophe Fiorio and Jens Gustedt. Two linear time Union-Find strategies for image processing. *Theoretical Computer Science*, 154:165–181, 1996.
- [4] W. G. Kropatsch and S. B. Yacoub. A revision of pyramid segmentation. In *Int. Conf. on Pattern Recognition*, pages 477–481, 1996.
- [5] C. McDiarmid. Concentration. In M. Habib, C. McDiarmid, J. Ramirez-Alfonsin, and B. Reed, editors, *Probabilistic Methods for Algorithmic Discrete Mathematics*, pages 1–54. Springer Verlag, 1998.
- [6] R. Watzel, K. Braun, A. Hess, W. Zuschratter, and H. Scheich. Restoration of dendrites and spines with the objective of Topologically Correct Segmentation. In *Int. Conf. on Pattern Recognition*, pages 472–476, 1996.
- [7] S. C. Zhu and A. Yuille. Region competition : Unifying snakes, region growing, and bayes/MDL for multiband image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 884–900, 1996.



cornouaille(256x256)



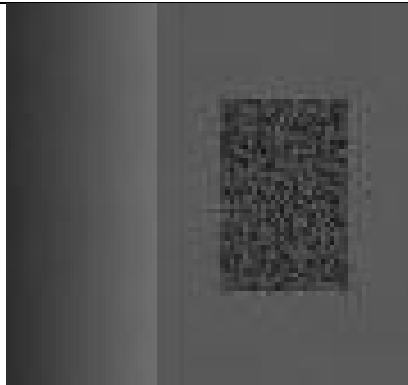
Our algorithm



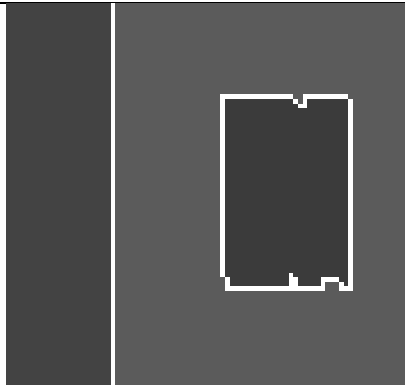
lena(512x512)



Our algorithm



synth(89x84)

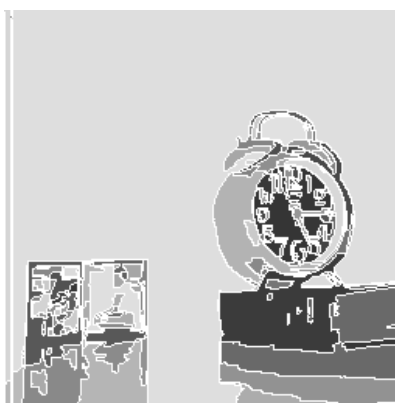


Our algorithm

Figure 2: Some experiments conducted (regions are white bordered, and averaged inside).



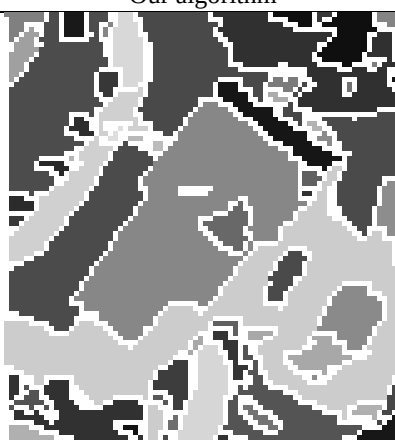
clock(256x256)



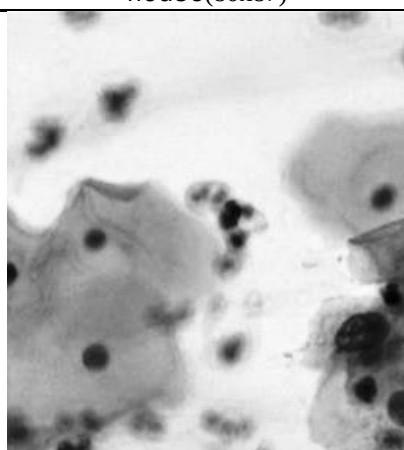
Our algorithm



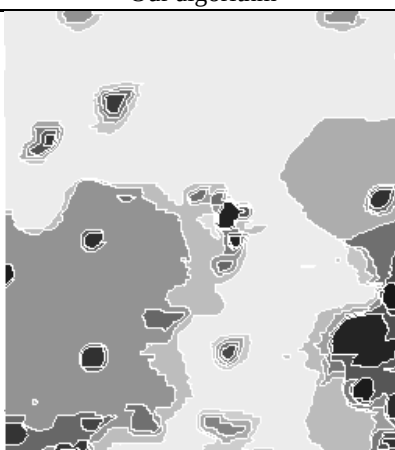
house(80x87)



Our algorithm

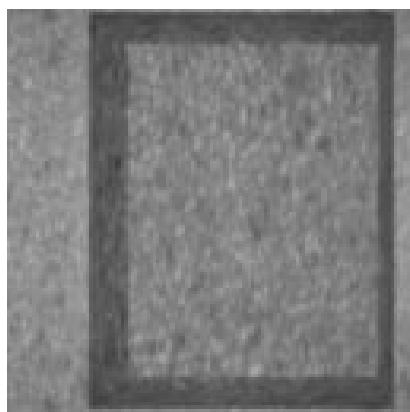


medical(255x285)

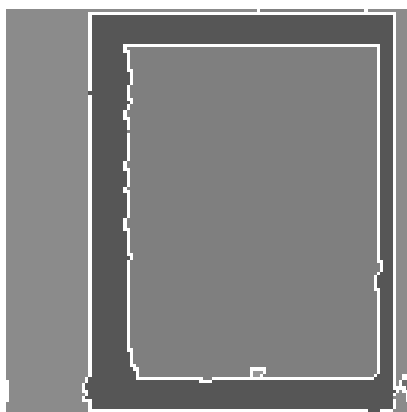


Our algorithm

Figure 3: More experiments (regions are white bordered, and averaged inside).



frame(128x128)



Our algorithm



street(320x240)



Our algorithm



Result of [2]



Result of [3]

Figure 4: More experiments (regions are white bordered, and averaged inside).

SORTED REGION MERGING TO MAXIMIZE TEST RELIABILITY

C. Fiorio

LIRMM, 161 rue Ada
161 rue Ada
34392 Montpellier, France

R. Nock

Université des Antilles-Guyane
UFR SEN-Dept of Maths and CS
97159 Pointe-A-Pitre Cedex, France

ABSTRACT

In this paper, we discuss an algorithmic approach to region merging which is built on a recent statistical work on the way to decide merging while keeping optimal complexity. In that latter work, a concentration-based statistical test is proposed, having the particularity to reduce the error occurring when rejecting the merging of two observed regions coming from the same true region. In this paper, we propose a preliminary ordered-based algorithmic procedure to cope with the errors occurring when merging two different regions in the first approach, thereby leading to a fast algorithm tailor-made for the reduction of both kinds of error. Experimentations proposed on images used without any preprocessing shed light on the quality of segmentations obtained.

1. INTRODUCTION AND RELATED WORK

In the early stage of a vision process it is necessary to isolate “objects” in the scene before recognizing them, a step called *image segmentation*. In this paper, we investigate image segmentation by region merging, which consists in building up regions by combining smaller regions, pixels being taken as primary regions. Our ultimate goal is to obtain a good partition of an image I into regions $\{R_1, R_2, \dots, R_n\} = P(I)$, each region being 4-connected: between any of its pixels, a path exists, lying in the region, with horizontal or vertical connections.

In the field of image segmentation, many algorithms use statistical or mathematical techniques, models, relying on hypotheses made *a priori* on the image [1, 2, 3]. Many of these hypotheses constrain the models by introducing distributional assumptions on the image. Also, the computational costs can be huge. Other approaches focus rather on topologic and algorithmic properties of the image [4, 5], striving to find fast segmentation algorithms. However, statistical confidence in the results is generally not ensured. In a recent work, we have been interested in a bridging technique between these two approaches [6]. It was based on recent results about concentration inequalities [7] adapted to image

segmentation, themselves derived from a particular model of image segmentation. Informally, this model supports the claim according to which an observed image I is principally a snapshot of an ideal object “scene” I^* , taken under particular conditions. Formally, in I^* , pixels are represented by distributions, used to generate the observed image. Ideal regions satisfy the homogeneity criterion that pixel’s expectations are equal, expectations that are in turn different between adjacent regions. This model gives an intuitive notion of what is the *optimal segmentation*. This could be obtained by infinite sampling of the distributions, thus under infinitely many “conditions of observations”. Averaging over the observed images would produce for each region its theoretical grey-level expectation, and we would observe the exact frontiers between regions.

From that model and recent concentration results [7], we derived [6] the probability that the intra-region variability (*IRV*) increase when merging two regions deviates up to “large values” when the two regions come from the same object in I^* . Solving the minimum variability τ for some (small) probability α , we obtain a simple test allowing to decide if two regions can be merged or not: if the increase in *IRV* exceeds τ , then we do not accept to merge the regions, while being confident up to risk α that the two regions are *not* element of the same object in I^* . The notion of *IRV* we use borrows almost everything to its equivalent variability notions in Variance Analysis, yet we have adapted it to our model of image generation. It is interesting to note that it can be computed in optimal (constant) time, and represent therefore a convenient choice of metric for our purposes [6]. Though our test handles the *first order risk*, there is no way to be formally confident when accepting to merge two regions, that is, we cannot be sure that two regions merged come from the same true region. This risk is crucial when testing small regions.

In this paper, we propose an algorithmic approach to solve this problem. We sort the pixels prior to their processing, according to the (increasing) *IRV* measured in their neighborhood, a task at a reasonable computational expense. By this way, the algorithm, which repeatedly tests the merging of two adjacent pixel’s regions, begins by merging regions

sharing similar local properties (variability), thus likely to come from the same true region. As regions grow, the test becomes stronger, and is more accurate whenever it rejects a merging. Due to the space constraints, we refer the reader to [6] for the bases of the model, the concentration bounds, and more generally all the statistical material. We concentrate here on the algorithmic material.

2. THE ALGORITHM ψ SCANLINE

We use as subprocedure the optimal-time and space algorithm SCANLINE (algorithm 1, [5]). It is based on a union-find structure which represents regions by trees of pixels belonging to the same region (see [8] for more information on union-find algorithms). The predicate $\mathcal{P}$ returns “Yes” or “No” depending on whether the two tested regions can be merged or not. The complete modified

Algorithm 1: ScanLine(I)

Input: an image I with r rows and c columns

```

for  $i = 0$  to  $r - 1$  do
  for  $j = 0$  to  $c - 1$  do
    if  $j > 0$  then
      left = Find( $I[i, j - 1]$ );
      current = Find( $I[i, j]$ );
      if  $\mathcal{P}(\text{left}, \text{current})$  then
        Union( $\text{left}, \text{current}$ );
    if  $i > 0$  then
      up = Find( $I[i - 1, j]$ );
      current = Find( $I[i, j]$ );
      if  $\mathcal{P}(\text{up}, \text{current})$  then Union( $\text{up}, \text{current}$ );

```

algorithm, ψ SCANLINE (for Probabilistic Sorted Iterated SCANLINE), works as follows. Before the first **for**, we sort the pixels $p, \dots \in I$ according to a function $f(p), \dots$. Denote as $R_k(p)$ the $k \times k$ squared region centered on pixel p , or centered as possible if p lies near the border of I . The function $f(p)$ is:

$$f(p) = \sum_{p' \in R_k(p)} \left(I(p') - \overline{R_k(p)} \right)^2$$

Here, $I(p')$ is the grey-level of pixel p' . In the case of RGB images, $f(p)$ sums three similar functions computed over each coordinate. Second, the predicate $\mathcal{P}(R, R')$ is the same as [6]. It is based on the IRV increase, $\Delta_i(R, R')$, which equals $\frac{|R| \times |R'|}{\sqrt{g}(|R| + |R'|)} (\overline{R} - \overline{R'})^2$ in our model, where $|\cdot|$ denotes the size in pixels and $\overline{R}$ is the grey-level average of R (see remark in [6] on the fact that this is not the usual definition from statistics). In RGB images, we calculate three IRV s in each coordinate. The predicate $\mathcal{P}$ returns true iff

$\Delta_i(R, R')$ does not exceed a quantity $\tau(R, R')$:

$$\tau(R, R') = \frac{g|R||R'|}{2(|R| + |R'|)^2} \log \left(\frac{32|I|^2 g_I^{(|R| + |R'|)}}{U_R U_{R'}} \right)$$

where for any region R , $U_R = 0.5 * \frac{|R|!}{|R|/2!}$ if $|R| < g_I$ (g_I is the effective number of grey-levels in I), and $U_R = 0.5 * \frac{|R|!}{(|R|/2)!}$ otherwise. As shown in [6], this predicate is actually constant time and space computable, a very convenient property. Finally, algorithm 1 is ran until convergence, *i.e.* until there is no more merging on the current iteration (experimentally, the number of iterations did not exceed 6, even on 1024x1024 images). Some experiments conducted with ψ SCANLINE are shown in figure 1 ($k = 13$). Note that the images were used without any preprocessing, and with the same parameter setting for all. The results show the quality of the segmentations obtained, even when using such raw material. They also show that the algorithm is able to handle regions with smooth gradients (mug). This partly comes from the fact that $f(p)$ reasonably fluctuates in regions with uniform (or smooth) gradients. Therefore, all pixels are grouped in the ordered list, and if they do not have too different colors, they will be merged by ψ SCANLINE.

3. REFERENCES

- [1] S. C. Zhu and A. Yuille, “Region competition : Unifying snakes, region growing, and bayes/MDL for multi-band image segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 884–900, 1996.
- [2] S. Dugelay, C. Graffigne, and J.-M. Augustin, “Segmentation of multibeam acoustic imagery in the exploration of the deep-sea bottom,” in *Int. Conf. on Pattern Recognition*, 1996, pp. 437–446.
- [3] R. Watzel, K. Braun, A. Hess, W. Zuschratter, and H. Scheich, “Restoration of dendrites and spines with the objective of Topologically Correct Segmentation,” in *Int. Conf. on Pattern Recognition*, 1996, pp. 472–476.
- [4] W. G. Kropatsch and S. B. Yacoub, “A revision of pyramid segmentation,” in *Int. Conf. on Pattern Recognition*, 1996, pp. 477–481.
- [5] Christophe Fiorio and Jens Gustedt, “Two linear time Union-Find strategies for image processing,” *Theoretical Computer Science*, vol. 154, pp. 165–181, 1996.
- [6] C. Fiorio and R. Nock, “A Statistical Approach to Region Merging of Optimal Complexity,” Tech. Rep. LIRMM-IFA-99101, Laboratoire

d'Informatique, de Robotique et de Microélectronique de Montpellier, 1999, Submitted, available at: <http://www.lirmm.fr/~fiorio>.

- [7] C. McDiarmid, "Concentration," in *Probabilistic Methods for Algorithmic Discrete Mathematics*, M. Habib, C. McDiarmid, J. Ramirez-Alfonsin, and B. Reed, Eds. 1998, pp. 1–54, Springer Verlag.
- [8] Kurt Mehlhorn, *Data Structures and Algorithms 1: Sorting and Searching*, Springer, 1984.

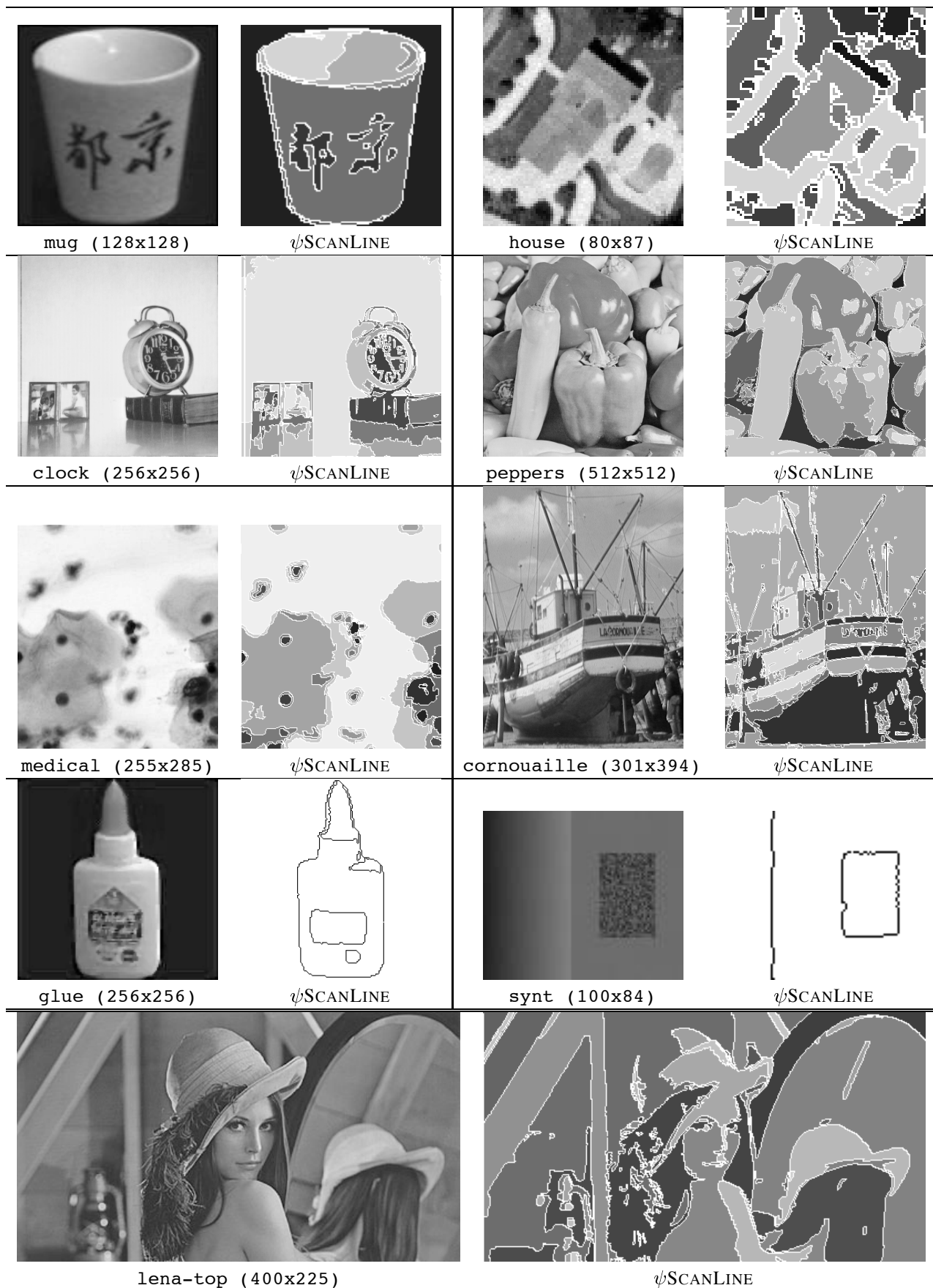


Fig. 1. Experiments conducted with ψ SCANLINE (regions are either white bordered and averaged inside, or black bordered and white inside).

Annexe C

Topological model for
two-dimensional image
representation : definition and
optimal extraction algorithm [[55](#)]



ELSEVIER

Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE @ DIRECT®

Computer Vision
and Image
Understanding

Computer Vision and Image Understanding 93 (2004) 111–154

www.elsevier.com/locate/cviu

Topological model for two-dimensional image representation: definition and optimal extraction algorithm

Guillaume Damiand,^{a,*} Yves Bertrand,^a and Christophe Fiorio^b

^a IRCOM-SIC, bât SP2MI, BP 30179, 86962 Chasseneuil Cedex, France

^b LIRMM, 161 rue Ada, 34392 Montpellier Cedex 5, France

Received 26 June 2001; accepted 3 September 2003

Abstract

In this paper, we define the two-dimensional *topological map*, a model which represents both topological and geometrical information of a two-dimensional labeled image. Since this model is minimal, complete, and unique, we can use it to define efficient image processing algorithms. The topological map is the last level of a map hierarchy. Each level represents the region boundaries of the image and is defined from the previous one in the hierarchy, thus giving a simple constructive definition. This model is similar to two existing structures but the main innovation of our approach is the progressive definition based on the successive map levels. These different maps can easily be extended in order to define the topological map in any dimension. Furthermore we provide an optimal extraction algorithm which extracts the different maps of the hierarchy in a single image scan. This algorithm is based on local configurations called precodes. Due to our constructive definition, different configurations are factorized which simplifies the implementation.

© 2003 Elsevier Inc. All rights reserved.

Keywords: Topological model; Image representation; Interpixel boundaries; Combinatorial map; Segmentation

* Corresponding author. Fax: +33-467-418-500.

E-mail addresses: damiand@sic.univ-poitiers.fr (G. Damiand), bertrand@sic.univ-poitiers.fr (Y. Bertrand), fiorio@lirmm.fr (C. Fiorio).

1. Introduction

In this paper we present a combinatorial structure which represents interpixel boundaries of a two-dimensional labeled image: the *topological map*. First we give a formal definition of the topological map, then we propose an optimal algorithm which builds this structure from a labeled image by a single image scan and a minimal number of operations. The problem of the definition of such a structure has been widely studied [5,7,9,12,22,29,35,36]. Indeed, it allows an efficient implementation of the main geometrical and topological operations used in principal image processing algorithms and more particularly in image segmentation.

There are many different data structures which represent the region boundaries of a labeled image. These structures are all more or less derived from the Region Adjacency Graph (RAG) [36]. This graph is composed of a set of vertices, one by region of the image, and there is an edge between two vertices if and only if the two corresponding regions are adjacent. However, the RAG has several drawbacks: it does not represent multiple adjacencies, does not make the difference between adjacency and inclusion, does not keep the order of the edges around a vertex, does not represent the faces but only vertices and edges and last it is not unique, i.e., two different images could be represented by the same RAG [28]. To solve these problems, the RAG model has been extended.

For example [29,31] define dual graphs structure, which are two multi-graphs representing inclusion relations. The first graph is equivalent to the RAG, but with multi-edges in order to represent multi-adjacency. The second graph is the dual of the first one. In order to avoid disconnection, edges are added between distinct boundaries of the same region in the primal graph. These special edges become loops in the dual graph. This allows the differentiation of the inclusion relation and the adjacency relation. But dual graphs are not topological representatives of the images: we can have two topologically different images having the same dual graphs. Another drawback of this structure is that each operation has to be applied twice (once to the primal graph and another to the dual one) in order to maintain the correspondence between the two graphs. At last, this structure is difficult to extend in higher dimension.

Two similar approaches have been proposed in order to give a solution to the problem of defining a structure representing all information resulting from a region segmentation. These two approaches are defined in the works of Domenger et al. [7,10–12,19] and Fiorio et al. [1,21,22]. The basic idea is to use combinatorial maps as a basis to represent the topology of the image. Indeed, combinatorial maps are a good model of space subdivision representation. They are defined in any dimension and represent all the cells of the subdivision and all the adjacency relations. Moreover, they can easily be linked to a geometrical model in order to represent the object geometry. Last, they are an efficient model to retrieve and to update information contained in the image [33].

The main difference between the two approaches is the algorithm which builds the topological map for Domenger and the topological graph of frontiers for Fiorio. The first one uses a contour tracking algorithm, the second one uses the notion of pre-

codes (Section 7, see also [13,21]) allowing to compute the structure in a single image scan. We can also quote the Frontier Graph defined by Jolion and Palloncy [35] which is very similar to the two previous structures. These structures have been defined in two dimensions. The need to work with images of higher dimension and specially in three dimensions has led to study how to extend previous works to higher dimension. But we rapidly reach the conclusion that structures are strongly related to the dimension two and so prevent their direct extension to the n -dimensional case.

To solve this problem, Bertrand et al. [5] proposes the definition of a new structure in n dimensions: the *border map*. This structure, also derived from combinatorial maps, represents the interpixel boundaries of a labeled image. But its major drawback is that this structure is not stable according to geometric transformations (translation, rotation, homothety, and local modifications): border maps of two isomorphic partitions for these transformations can be completely different. This is a major drawback since the structure does not characterize images. This implies problems for matching algorithms and shows that the border map model does not provide a minimal encoding.

More recent works [8,9] extend the topological map in three dimensions. But the contour tracking algorithm which allows to construct the topological map in two dimensions is difficult to extend in three dimensions. They need to avoid particular configurations of voxels in the image: when two voxels with same label are adjacent by an edge. So before extracting the topological map, they first modify the image in order to remove all these configurations. This is an heavy time-consuming pre-processing which is not satisfying since it modifies the initial image.

To propose a solution, we reexamine the two dimensions topological map in order to give a formal definition which easily allows to extend it to higher dimension. This model has to be:

- *minimal* in the number of cells, to optimize memory occupation but also complexity of algorithms that have a direct access to the adjacency information;
- *complete*: it must represent both topology and geometry of images;
- *unique*: it must be invariant for geometric transformations: two topological equivalent images must have the same representation in order to facilitate matching algorithms.

To propose a new definition, we introduce the new notion of *simplification levels* which allows us to give a simple and constructive definition of the two dimensions topological map based on the removal operations. Moreover, due to the genericity of the constructive definition and the removal operations, the extension to the three-dimensional case is facilitated [2,3].

First we give in Section 2 a brief recall on combinatorial maps which are the basic model used in this work. Then we give in Section 3 some notations and some recalls on the interpixel notion. We introduce in Section 4 our new notion of simplification level, and we define the topological map by using several intermediary levels. This section presents only the topological part of our model in order to simplify the understanding. The geometrical part and the link between the two models is the object of Section 5. Then we present a first extraction algorithm in Section 6 which comes directly from the constructive definition. In Section 7 we give the optimal extraction

which uses the precode notion. Precodes were already used in previous works [13,21] (only in two dimensions), but we show here a parallel between precodes to consider and the level we want to extract. We study all the different cases to process in order to extract any map level in a single image scan, and we show that some configurations can be factorized. In Section 8 we give our algorithm that computes the inclusion tree. In Section 9, we present more precisely the two similar existing approaches in order to understand the main differences and to see why our solution can be easier be extended in higher dimension. We present some experimental results in Section 10. Finally we conclude this paper in Section 11 and give some perspectives for future works.

2. Combinatorial maps

The combinatorial map is a mathematical model of space subdivision representation based on planar map [14,15,20,24,37]. The subdivision of a two dimensions topological space is a partition of the space into three subsets whose elements are *cells* of 0, 1, and 2 dimensions (respectively, called vertices, edges, and faces, and noted *i*-cell for a *i*-dimensional cell). Border relations are defined between these cells, where the border of a *i*-cell is a set of ($j < i$)-cells. We say that two cells are *incident* when one belongs to the border of the second, and that two *i*-cells are *adjacent* if they are both incident to the same ($j < i$)-cell. Combinatorial maps encode all the subdivisions and the incidence relations between all the different cells of the space, and so represent the topology of this space. They are defined formally for any dimension, and we call *n-map* an *n*-dimensional combinatorial map. The *n*-maps can encode all orientable manifold subdivisions of an *n*-dimensional space without boundary. They were generalized in [32,34] in order to encode all *n*-dimensional, orientable or not and with or without boundary subdivisions (see [33] to find a connection between maps and several other models).

A combinatorial map can be obtained intuitively by successive decompositions as we can see in Fig. 1. To represent the two dimensions object shown in Fig. 1A, we first distinguish the faces of this object (Fig. 1B) then the edges of these faces (Fig. 1C). Adjacency relation between the two faces is represented by the black segment and adjacency relations between each couple of edges is represented by grey arc circles. The obtained elements after all the decompositions are called *darts* and are the single basic elements of the combinatorial map definition. To obtain the map,

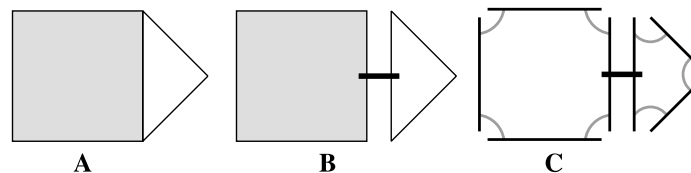


Fig. 1. The successive decompositions of an object to obtain the corresponding 2-map. (A) A 2D object. (B) Disjoined faces. (C) Disjoined edges.

we only report each adjacency relation onto darts, and call β_i the relation between two i -dimensional elements. Let us see now the formal definition of two dimensions combinatorial maps that we can find for example in [33]:

Definition 1 (*Two dimensions combinatorial map*). A two-dimensional combinatorial map, (or 2-map) is a triplet $M = (D, \beta_1, \beta_2)$ where:

- (1) D is a finite set of darts;
- (2) β_1 is a *permutation*¹ on D ;
- (3) β_2 is an *involution*² on D .

When two darts d_1 and d_2 are such that $\beta_i(d_1) = d_2$ ($1 \leq i \leq n$), we say that d_1 is i -sewn to d_2 . We speak about i -sewing (resp. i -unsewing) for the operation that puts two darts in relation for β_i (resp. that removes an existing β_i relation). We note β_0 for β_1^{-1} , and β_{ji} for $\beta_i \circ \beta_j$ ($\beta_i \circ \beta_j(d) = \beta_i(\beta_j(d))$), we first apply β_j then β_i , the permutations are applied in the same order as they are read in the notation β_{ji}).

We can see in Fig. 2A the 2-map which represents the object shown in Fig. 1. The β_1 relation connects an edge and the following edge in the same face, and the β_2 relation connects the two faces incident to the same edge. In order to simplify the figures, we use the graphical representation presented in Fig. 2B where the β_i are not explicitly drawn. Each dart is represented by an arrow that shows the face orientation. With this orientation, we can retrieve, for each dart, the following dart in the same face and so deduce the β_1 permutation. Moreover, two darts 2-sewn are drawn near and parallel and the β_2 permutation can also be deduced from the graphical representation.

Within the combinatorial map framework, all space cells are represented implicitly using the notion of *orbit*:

Definition 2 (*orbit*). Let $\Phi = \{f_1, \dots, f_k\}$ some permutations on D . We note $\langle \Phi \rangle$ the permutation group generated by Φ . This is the set of permutations obtained by any composition and inversion of permutations contained in Φ . The *orbit* of a dart d relatively to Φ is defined by $\langle \Phi \rangle(d) = \{\Phi(d) | \Phi \in \Phi\}$.

Intuitively, an orbit $\langle f_1, \dots, f_k \rangle(d)$ is the set of darts that we can reach with a traversal starting with d and using all combinations of all the f_i or f_i^{-1} permutations. Given a 2-map and a dart d , we can retrieve all the cells incident to d by using particular orbits. The vertex incident to d is defined by $\langle \beta_{21} \rangle(d)$, the edge by $\langle \beta_2 \rangle(d)$ and the face by $\langle \beta_1 \rangle(d)$. A dart is said to be incident to a cell if it belongs to the set of darts that represents the cell. Moreover, two cells are incident if the two corresponding sets have a non-empty intersection. In the 2-map shown in Fig. 2, the vertex incident to the dart 8 is the orbit $\langle \beta_{21} \rangle(8) = \{5, 8, 11\}$, the edge incident to the same dart is $\langle \beta_2 \rangle(8) = \{8, 10\}$ and the face is $\langle \beta_1 \rangle(8) = \{6, 7, 8, 9\}$.

¹ A *permutation* on a set S is a one to one mapping from S onto S .

² An *involution* f on a set S is a one to one mapping from S onto S such that $f = f^{-1}$.

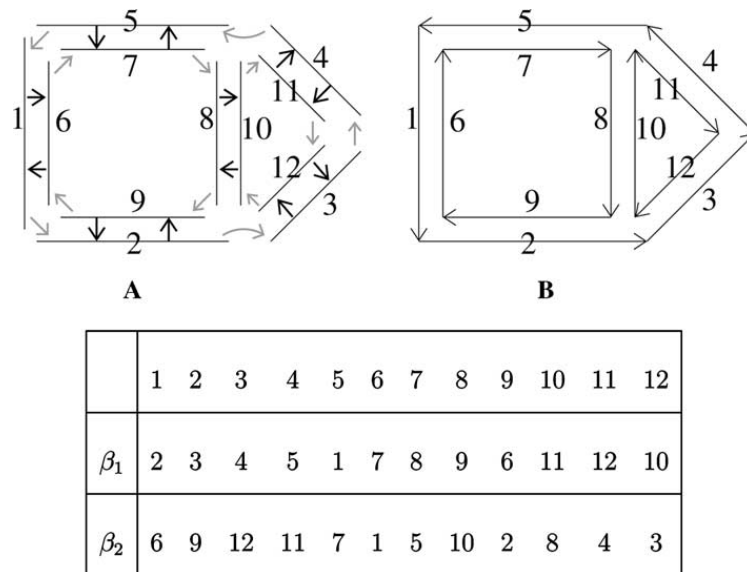


Fig. 2. Two different representations of the same combinatorial map. (A) Full representation. Darts are represented by numbered black segments, β_1 relations are represented by grey arrows and β_2 by thick black arrows. (B) Implicit representation where the β_i are not explicitly drawn. Darts are represented by numbered black arrows (to represent the orientation). Two darts 1-sewn are drawn consecutively, and two darts 2-sewn are concurrently drawn and in reverse orientation.

Combinatorial maps only represent object topology, and not their geometry. But it is easy to add some geometric elements to some (even all) orbits of the combinatorial map: this operation is called *embedding*.³ The separation of topological and geometrical models is one of the main advantages of combinatorial maps. Indeed, such a separation allows us to classify each operation in three different categories:

- Topological operations, which exclusively modify the topological model so the combinatorial map.
- Geometrical operations, which exclusively modify the geometrical model, the combinatorial map remaining unchanged.
- Mixed operations, which modify both models. But even in this case, we can usually decompose this operation in two distinct steps: first the topological modifications then the geometrical ones.

The separation of these two models allows us to modify either the embedding only or the topological model only. Moreover, operations become easier to define when the modifications of the topological and the geometrical model are separated.

³ One embedding example of a combinatorial map consists in linking to each topological vertex of the map the coordinates of an Euclidean space point.

3. Using combinatorial maps for image representation

Let us here recall some usual notations. A pixel is a point of discrete plan $\mathbb{Z}^2$ associated with a value which could be a color, a grey level. . . A two-dimensional image is a finite set of pixels. We use the classical notion of 4-connectivity because combinatorial maps can not represent non-manifold and so the 8-connectivity can not be considered without some tricks. In this work, we use combinatorial maps to represent pixel sets of *labeled images* having same values and which are 4-connected.

Definition 3 (*labeled image*). A *labeled image* is a set of labeled pixels such as two pixels with the same label belong to the same 4-connected component.

Note that we can represent any type of image simply by labeling all the 4-connected pixel sets that have the same value, for example with a growing algorithm. Considering only labeled images is an optimization that allows us to immediately retrieve, given the label of a pixel, the connected component it belongs to. But this work can be extended in order to consider any types of images. We speak about *region* for a set of same labeled pixels. Two pixels with same label belong to the same region, and two different regions have two different labels. We use this property only for the inclusion tree (cf. Section 8) in order to use the label of regions as a unique identifier, and so it is possible to extend this work to consider regions made of several 4-connected components. To avoid particular processes of the image border pixels, we consider an infinite region R_0 that contains all the pixels that do not belong to the image. With this region, each pixel has exactly four neighbors (for the 4-adjacency). Moreover this infinite region allows us to process any type of image, not only the ones rectangular and without holes.

We say that a region R_i is *included* into a region R_j if and only if any 4-connected path going from a pixel of R_i to a pixel of R_0 (the infinite region) has at least one pixel belonging to region R_j . Intuitively, this inclusion notion corresponds to the fact that a region is around another one without any constraint on the way this surrounding is done. We can notice that each region is at least included into the infinite region, and that this relation is a partial order relation.

Combinatorial maps represent the borders of the regions contained in the image. Several works have been done on the notion of boundary in a discrete image and have shown that using a topology based on the *interpixel* notion [21,23,25,26,28] enables to define these borders so that they verify classical topological properties, as in the Jordan theorem [27].

In the interpixel framework, an image is not considered only as a matrix of pixels, but as a subdivision of a two-dimensional space in a set of cells: *pixels* (the 2-cells called sometimes *faces*), *linels*, the 1-cells in between two 2-cells (also called *cracks*), and *pointels*, the 0-cells in between 1-cells (or *points*). These different types of cells can be seen in Fig. 3. With these cells, the notion of curve is topologically correctly defined: a *curve* is a list of alternated pointels and linels (with a pointel on each end) such as two consecutive cells in the list are incident. A curve is said *simple* when all

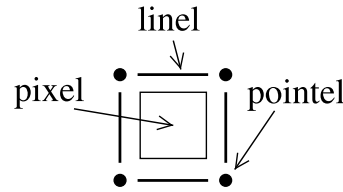


Fig. 3. All the cells of the interpixel two-dimensional space.

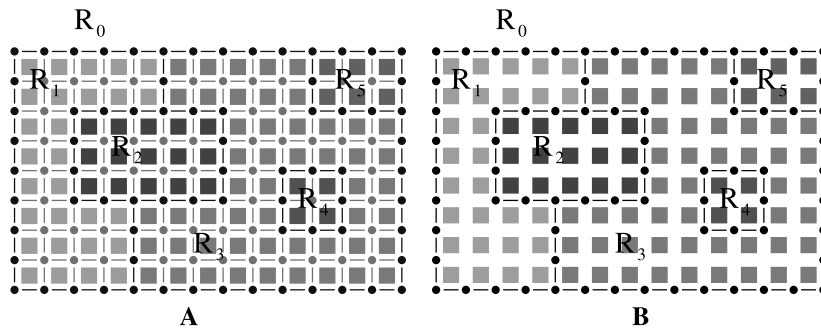


Fig. 4. Interpixel and boundaries of a labeled image. (A) An image with all the interpixel elements. (B) Interpixel boundaries of the image regions.

the cells of the curve are distinct (a curve is simple if it is not self-intersected), and *closed* if the first cell is the same as the last one.

The *interpixel boundary* between two regions R_i and R_j is the set of simple curves two by two disjointed, such as each linel of these curves is incident to exactly one pixel of R_i and one pixel of R_j . We call *boundary curve* a curve belonging to one of the image boundaries. These boundary curves are maximal: each linel of the image incident to one pixel of R_i and one pixel of R_j belongs to one boundary curve, and two adjacent lines belonging to the same boundary belongs to the same boundary curve (they are separated in this curve by the pointel incident to these two lines).

Fig. 4 represents a labeled image and its interpixel boundaries. The boundary between regions R_1 and R_3 is composed of two distinct curves, the one between R_3 and R_4 is composed of one closed curve, and the boundary between R_1 and R_4 is empty since these two regions are not adjacent. Note that this definition of interpixel boundary is valid for any region, due to the existence of the infinite region.

4. Topological map: last level of several maps definition

Combinatorial maps are well suited to represent interpixel boundaries since they represent space subdivisions and all the incidence relations between the different cells. For this reason they were used in several previous works [5,7,9,12,22,35]. But different combinatorial maps can represent the same image, and these works have used different map variants according to their needs.

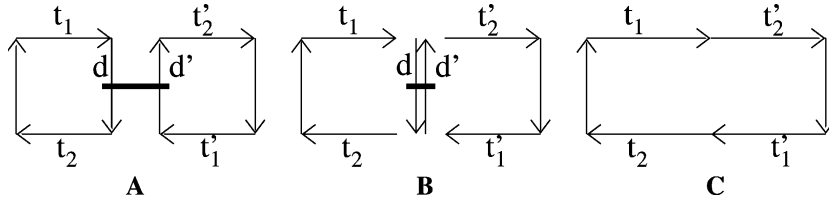


Fig. 5. 1-removal of the edge incident to the dart d . (A) Initial configuration. (B) 0- and 1-unsewing of d and d' . (C) 1-sewing of t_1 with t_2' and t_1' with t_2 .

In this paper we introduce the notion of *simplification levels* which allows us to give a simple and constructive definition of the two-dimensional topological map. These levels can be formally defined with simple constructive definitions, and can be extended in higher dimension. The topological map is the last simplification level since it is minimal; other levels are only intermediate concepts. But it is possible to use an intermediate level in a particular application where we have no memory constraint (for example [4] uses a map equivalent to our level 2 in a three-dimensional modeler). First, we only present the topological part of this work without looking at the link with a geometrical model. It allows us to help with the understanding of our model and not to mix the two parts up. The link with a geometrical model is the purpose of Section 5.

The main idea of our approach is first to build a complete combinatorial map, that represents all the interpixel cells of the image, and then to progressively simplify it as long as no topological information is lost. This construction scheme allows us, in the end, to define the minimal map that represents the interpixel boundaries. At every step of the process, we only need to verify that there is no topological information loss in order to ensure the validity of the new map. To perform the successive simplifications, we use the *removal operations*.

4.1. Removal operations in map

The i -dimensional removal operation (noted i -removal) consists in removing a i -cell. This leads to the merging of the $(i + 1)$ -cells incident to the removed cell. In two dimensions, we can remove an edge or 1-cell (1-removal, see an example in Fig. 5) or remove a vertex or 0-cell (0-removal, see an example in Fig. 6). We only present here the main notions of these operations. A more complete description can be found in [17] where we give general definitions of removal and contraction⁴ operations.

We can remove any edge in a 2-map without constraint (see example in Fig. 5 where we remove the edge incident to dart d). In order to perform this operation, we first 0- and 1-unsew the edge incident to the dart d (d and d' in Fig. 5B). Then we sew again “correctly” the darts which were previously sewn to this edge (t_1 with t_2' and t_1' with t_2 , see Fig. 5C) and remove the free darts d and d' . We can prove this

⁴ Contraction is the dual operation of the removal. It consists in contracting a i -cell into a $(i - 1)$ -cell.

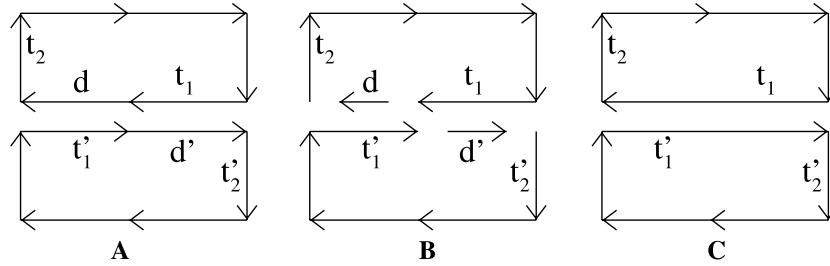


Fig. 6. 0-removal of the vertex incident to the dart d . (A) Initial configuration. (B) 0, 1 and 2-unsewing of d and d' . (C) 1-sewing of t_1 with t_2 and t_1' with t_2' . 2-sewing of t_1 with t_1' .

operation is correct whatever the initial configuration and the edge to remove (even in degenerated cases, as for example the removal of an isthmus, see [17]).

The 0-unsewing operation of a dart d consists only in removing the existing relation between d and $\beta_0(d)$. For that, we can use a particular dart (called for example NIL ⁵) and define the 0-unsew(d) by the two affectations: $\beta_1(\beta_0(d)) = NIL$ and $\beta_0(d) = NIL$, the 1-unsew(d) by $\beta_0(\beta_1(d)) = NIL$ and $\beta_1(d) = NIL$ and the 2-unsew(d) by $\beta_2(\beta_2(d)) = NIL$ and $\beta_2(d) = NIL$. We need to modify two values in order to preserve the properties of combinatorial maps ($\beta_0 = \beta_1^{-1}$ and β_2 is an involution). Using the 0-unsewing to define the 0-removal allows us to be sure that we always preserve these properties. Indeed, without unsewing, we can for example 1-sew a dart d , already 1-sewn to d_2 , to another dart d_3 and thus obtain an object which is no longer a combinatorial map since $\beta_0(d_2)$ is equal to d and $\beta_0(d_3)$ too. Moreover, 0-unsewing is also useful to update eventual embeddings (see Section 5).

The vertex removal is possible only for degree two vertices (the degree of a i -cell c is the number of distinct $(i + 1)$ -cells incident to c). Otherwise, it is not possible to automatically decide how to connect the different edges around the removed vertex. We can see an example in Fig. 6 where we remove the vertex incident to dart d . The operation is performed with a similar algorithm to the edge removal. First we unsew the darts incident to the vertex (d and d' in Fig. 6B) then we sew the free darts (t_1 with t_2 and t_1' with t_2' , see Fig. 6C) and remove free darts d and d' .⁶

4.2. Level 0: Complete map

Level 0 map is the starting point of our process and represents all the interpixel cells of a labeled image.

⁵ If we want to formally define this particular value, we can modify the combinatorial map definition (Definition 1) by adding a particular dart called NIL such as $\forall i, \beta_i(NIL) = NIL$. This modified definition can also be used to represent objects with boundaries.

⁶ Vertex removal can be achieved with edge contraction. Indeed, in the example shown in Fig. 6, removing the vertex incident to d is equivalent to contracting the edge incident to d (and also equivalent to contracting the edge incident to d'). The difference between vertex removal and edge contraction is the precondition of the operation: edge contraction can be achieved without constraint, and vertex removal can be done only for degree two vertices.

Definition 4 (*level 0 map*). Level 0 map corresponding to an $n_1 \times n_2$ pixels labeled image, is the map having $n_1 \times n_2$ square faces 2-sewn between them, each face corresponding to a pixel, plus an enclosing face which represents the infinite region.

Fig. 7B shows the level 0 map of the image shown in Fig. 7A. For an $n_1 \times n_2$ image, this map is composed of $(n_1 \times n_2) + 1$ faces. $n_1 \times n_2$ square faces, each one representing a pixel of the image, made of four darts, and an additional face that represents the infinite region, made of $2 \times (n_1 + n_2)$ darts. Two darts are 1-sewn when they represent two consecutive linels of the same face, and are 2-sewn when they represent the same linel.

4.3. Level 1: Linel map

Level 0 map represents all the interpixel cells of an image. In order to obtain a combinatorial map which represents only the interpixel boundaries, it is necessary to remove all linels corresponding to inner boundaries. This operation is done with the edge removal.

Definition 5 (*Level 1 map*). Level 1 map is the map obtained from level 0 map by removing each edge between two pixels having the same label.

We can see in Fig. 8A the image of our example and in Fig. 8B the corresponding level 1 map that represents all the interpixel boundaries of the image. Each edge of this map corresponds exactly to one linel of a boundary curve.

It should be noticed that each region is represented in level 1 map by an external border and zero or several internal borders. A region R has internal borders when there are some regions included into R . In our example in Fig. 8B, only region R_3 has one internal border (without considering the infinite region that is always represented by an internal border). The map is disconnected into several connected components (in the example, two connected components), and we have lost the topological information necessary to place and link the connected components. Without this information, it is not possible to distinguish two non-adjacent regions

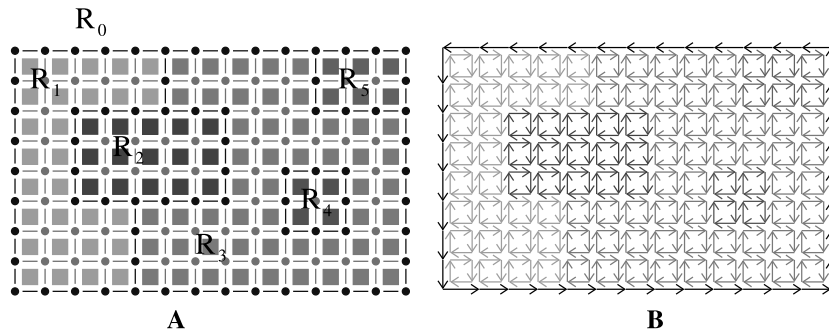


Fig. 7. (A) An image and its interpixel elements. (B) The corresponding level 0 map.

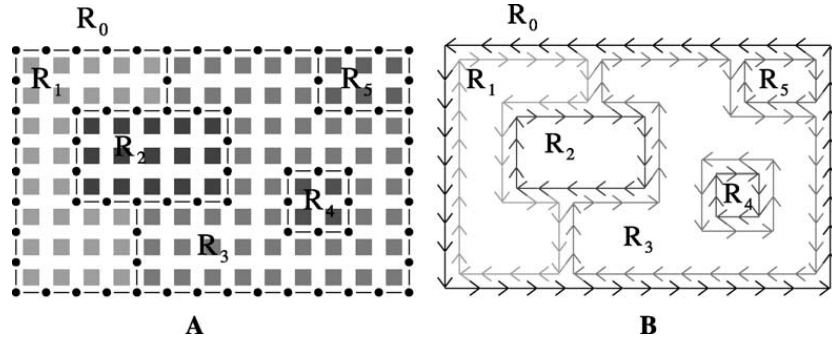


Fig. 8. (A) Interpixel boundaries. (B) The corresponding level 1 map.

and one region included into another one. In order to keep this information, we introduce a region inclusion tree of the (more precisely described in Section 8). This tree has one node for each image region of the. Its root is the infinite region, and the set of regions included in a given region R defines the sons of R in the tree. This tree allows us to retrieve the inclusion information and so to retrieve all the boundaries of a given region.

This level 1 is the first map that represents interpixel boundaries of a labeled image. But it is clear that this structure is not optimal, neither for space, nor for time complexity. Indeed, due to the number of darts having to be traverse to retrieve adjacency information, operations on this map require heavy computations. This is why we simplify this map and define level 2.

4.4. Level 2: Border map

Level 2 map is the most intuitive map encoding interpixel boundaries of an image. Indeed, each edge of this map represents a straight part of a boundary (this map is called *border map* in [5]). We can see level 2 map of our image in Fig. 9.

Definition 6 (level 2 map). Level 2 map is the map obtained from level 1 map by removing each degree two vertex between two aligned edges.

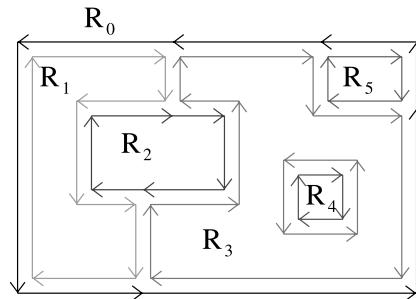


Fig. 9. Level 2 map.

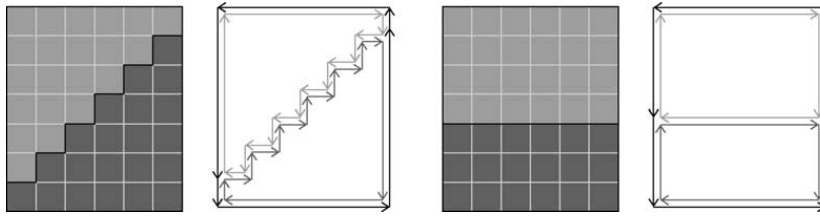


Fig. 10. Two topological equivalent partitions and their corresponding level 2 maps.

Each edge of level 2 map corresponds to a maximal series of aligned linels of a boundary curve. But these edges are now of any length, contrarily to level 1 map where edges are of length one. We can prove that level 2 map is topologically equivalent to level 1 map. The only modification achieved on the level 1 map is the removal of each degree two vertex between two aligned edges. Each couple of aligned edges incident to a degree two vertex is necessary between the same two regions R_i and R_j . So the two edges represent the same topological adjacency information and deleting one of the two does not lead to any loss of topological information. Moreover, the vertex removal cannot lead to the disconnection of the map in several connected components.

The main problem of level 2 map is its non-unicity. Indeed, the number of darts used to represent a boundary curve depends on the geometry of this boundary. In the worst case, when a boundary does not have any consecutive aligned linel (cf. the stair case in Fig. 10), the number of darts of level 2 map is exactly twice the number of linels of the boundary. This non-unicity implies that two topologically equivalent partitions can have two completely different level 2 maps.

4.5. Level 3: Topological map

To solve the non-unicity problem of level 2 map, we need to perform the same simplifications achieved to define level 2 map but for non-aligned edges. Indeed, after these simplifications, we obtain a model which do not depend on the geometry of the image boundaries but only on its topology. Level 3 map of our example is shown in Fig. 11.

Definition 7 (level 3 map). Level 3 map is the map obtained from level 2 map by removing each degree two vertex.

Each edge of level 3 map corresponds exactly to a boundary curve of the image. This gives the final model the name of "topological map": level 3 map plus the inclusion tree. This map is minimal, we cannot do any additional removal without modifying the topology. To ensure that level 3 map contains the same topological information as level 2 map (and so as level 1 map), we use the same proof as the previous map and we show that two edges incident to a degree two vertex represent the same adjacency information.

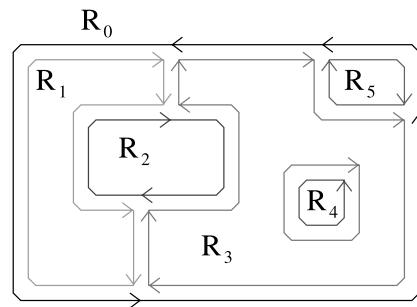


Fig. 11. Level 3 map.

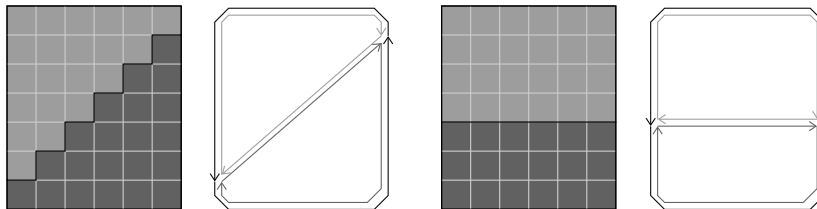


Fig. 12. Two topologically equivalent partitions and their corresponding level 3 maps.

Level 3 map only depends on the topology of the represented image. That is why two topologically equivalent partitions are represented by two isomorphic maps as we can see in Fig. 12. Level 3 map represents each boundary curve on a minimal way (one edge) and thus we can no longer simplify the map. This map needs much less memory than the others, it is consistent with the topology of the represented objects and it is stable for geometric transformations of the corresponding image. Moreover, image processing algorithms need less computational time when they are defined on level 3 map because of the direct access of the adjacency information.

5. What geometrical model for each map level

Combinatorial maps only represent the topological part of images. But in most applications, it is necessary to represent also geometry. There are several ways to associate a geometrical model to a combinatorial map. We present here one solution for each map level and show how these models have to be modified during our main operations: sewings, unsewings, and removals.

5.1. Level 1 and level 2 maps

For level 1 and level 2 maps, we associate a two-dimensional geometrical point (given by its coordinates) to each topological vertex of the map. Each topological vertex is represented in the map by a set of darts, but only one of these darts is linked

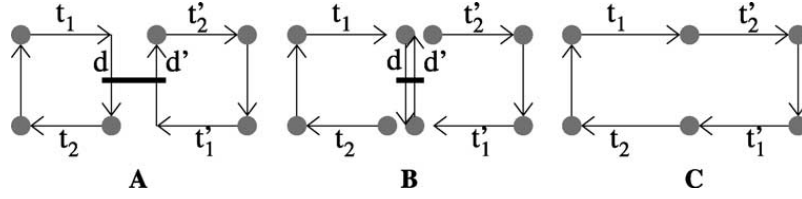


Fig. 13. 1-removal of the edge incident to the dart d with embedding modification. (A) Initial configuration. (B) 0- and 1-unsewing of d and d' . The two vertices incident to d and d' are duplicated. (C) 1-sewing of free darts: t_1 with t'_1 and t_2 with t'_2 . Geometrical points linked with d and d' are removed.

with the geometrical point. Given a dart, we have to traverse the vertex orbit in order to find this particular dart and so the geometrical point.

We use the same geometrical model for level 1 and level 2 maps in order to simplify algorithms. Indeed, we do not need a conversion of the geometrical model between these two levels. But this leads to a more important complexity in memory space for level 1 since we do not use its particular properties.

This embedding is simple since only one type of cell is embedded. Moreover, retrieving the embedding of the edges or the faces can easily be achieved since it is sufficient to traverse topological cells and to keep the coordinates of each vertex. To ensure the validity of this embedding, we must verify that each topological vertex has exactly one dart that is linked with a geometrical point.

This property can be preserved automatically by modifying the sewing and unsewing operations. Indeed, we test when a sewing operation groups together two initial distinct vertex orbits and then remove one⁷ of the two geometrical points when it is the case. We also check when an unsewing operation breaks an initial topological vertex into two distinct orbits, and then we duplicate the geometrical point. We can see in Fig. 13 the 1-removal operation (already presented in the previous section) with the embedding modifications. After the 0- and 1-unsewing of d and d' , the two incident geometrical points have been duplicated as the corresponding vertices have been broken into two orbits (Fig. 13B). Then, after the 1-sewing of free darts (Fig. 13C), the two geometrical points associated with d and d' are removed along with the deletion of the two darts.

Note that this is not the optimal embedding updating. Indeed, if we know performing a 1-removal operation (as the example shown in Fig. 13), we know that darts d and d' are going to be deleted and thus it is not required to duplicate the two geometrical points. We can sometimes choose to use the general sewing and unsewing operations that check and update the vertex embeddings, and some other times the basic sewing and unsewing operations (for example when we perform a set of sewing operations and know exactly how the different orbits are going to be modified).

⁷ When we i -sew(a, b), we keep the geometrical point associated with a . For this reason, 0-sew(a, b) or 1-sew(b, a) give the same topological result but not the same geometrical one.

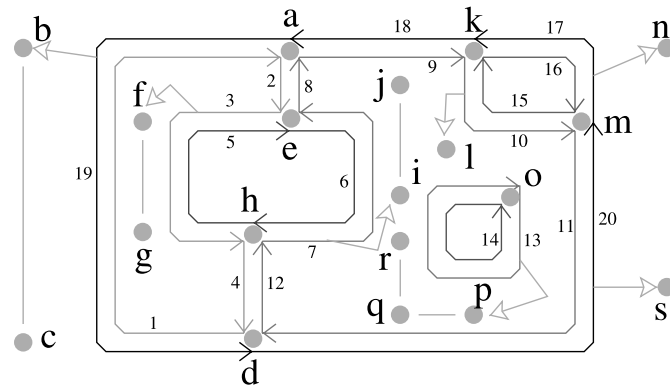


Fig. 14. Topological map with open curves and vertices embedding.

But this is only an optimization and in the following we do not detail when we can use basic operations instead of general ones. Moreover, the additional cost in complexity of general sewing and unsewing operations is not important. Indeed in the worst case there are four darts incident to the same geometrical point (one dart by line incident to the corresponding point) and thus the test on the vertex orbit can be performed in four elementary operations that is to say in constant time.

5.2. Level 3 map

For level 3 map it is not possible to use the same embedding as for the two other levels. Indeed, we cannot retrieve the embedding of an edge with only the embedding of its end vertices because edges can now have any geometrical shape.

In order to embed this map, we associate a two-dimensional geometrical point to each topological vertex (as for the previous levels), and we associate to each topological edge the one-dimensional curve corresponding to the interpixel boundary without its two extremities (we call these curves *open curves*). These curves are represented by one-dimensional combinatorial maps.⁸ These 1-maps represent maximal straight line segments in order to minimize the number of elements. Only one dart of the two that compose an edge is linked to the 1-map, and the orientation of the 1-map is the same as the one of this dart.

The main advantage of this embedding is to not encode twice the coordinates of the vertices. The main drawback is that we do not have immediately the embedding of a boundary curve but we need to reconstruct it from one open curve and two geometrical points. But this can be performed in linear complexity for the number of points of the boundary curve.

⁸ A 1-map is equivalent to a double-linked list but this allows us to consider the extension in higher dimension where each i -dimensional topological cell could be embedded with a i -dimensional combinatorial map.

We can see in Fig. 14 the embedding of level 3 map of our example image used in this paper. This map has seven topological vertices (embedded by geometrical points a, d, e, h, k, m , and o) and 10 topological edges. Among them, three do not have an edge embedding (edges $\{2, 8\}$, $\{4, 12\}$, and $\{9, 18\}$). Indeed, when a topological edge corresponds to a straight line segment, the one-dimensional open curve is empty and the two geometrical end points are enough to reconstruct the edge geometry.

The method to reconstruct the embedding of an edge incident to a dart, for example dart numbered 19 in the figure, consists in retrieving in a first time the embedding of the vertex incident to 19 (geometrical point a), then in retrieving the embedding of the edge (geometrical points b and c), and at last in retrieving the embedding of the dart $\beta_2(19)$ (geometrical point d). The one-dimensional curve made from the concatenation of these three embeddings is the embedding of the edge (list of geometrical points a, b, c , and d).

During modification operations, we need to update both embeddings (vertex and edge embeddings). For the vertex embedding, we can exactly perform the same modifications of the sewing and unsewing operations as for level 1 and level 2 maps. The edge embedding has to be updated only for the 0-removal operation. Indeed, when we remove an edge, the corresponding open curve is simply destroyed. But when we remove a vertex, we need to keep the geometry of the two merged edges in the unique embedding of the resulting new edge.

In order to do that, we process in two steps (shown in Fig. 15). So as to remove the vertex incident to dart d , we first add at the beginning of the 1-map associated to d the geometrical point of the removed vertex (depending on the dart that is linked with the 1-map, this is equivalent to adding the geometrical point at the end of the 1-map associated to $\beta_2(d)$). Note that this creates such a map if the edge incidents to d does not already have an edge embedding (case of straight line segments).

In a second step we merge the edge embedding of $\beta_0(d)$ and the edge embedding of d (with a particular case to take into account when one edge does not have an edge embedding). When the two 1-maps do not have the same orientation, we need to reverse one of the two before merging them in a unique 1-map. Moreover, in order to represent maximal straight line segments, we check during this merging if the three

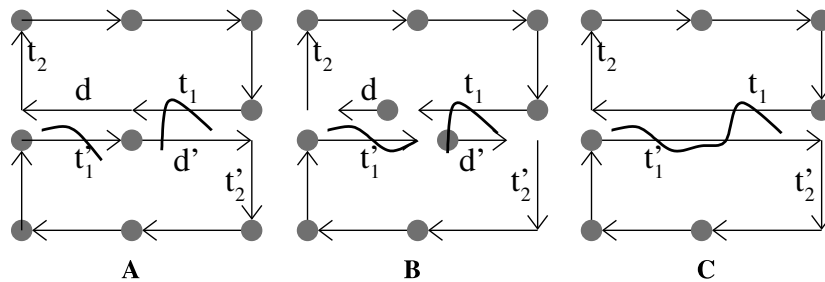


Fig. 15. 0-removal of the vertex incident to the dart d with embedding modification. (A) Initial configuration. (B) 0- and 1-unsewing of d and d' . The geometrical point incident to d is added in the beginning of the 1-map associated to the edge incident to d . (C) 1-sewing of free darts and merging of the two 1-maps associated with the edges incident to d and d' .

geometrical points around the removed vertex are aligned, and when it is the case we remove the geometrical point inserted in the previous step.

As for the embedding of level 1 and level 2 maps, the updating of the embedding can automatically be achieved by the sewing and unsewing operations and the 0-removal operation, but it can also be achieved manually for particular operations in order for example to optimize and avoid several tests of the same orbit. But in this last case, we must make sure that the embedding is correctly updated in order to avoid a loss of geometrical information.

6. Basic generic extraction algorithm of any map level

The successive definitions of each map level seen in Section 4 immediately give a first extraction algorithm presented in Algorithm 1. The main advantage of this algorithm is its simplicity. Indeed, it follows the definitions of the different map levels. Moreover, it is generic: the same algorithm can extract any map level. First, this algorithm builds level 0 map of the labeled image (line 0). Then, depending on l , the level we want to extract, we progressively simplify the map by following the level definitions. Each line numbered i of the algorithm constructs level i map of the image. At last (line 4) we compute the inclusion tree with the algorithm presented in Section 8.

Algorithm 1 (*Basic extraction of level l map*).

Input: A labeled image I of $n_1 \times n_2$ pixels.
 l the level of the map we want to extract ($l \in \{1, 2, 3\}$).
Output: The level l map corresponding to I .
0 Build a 2-map M of $n_1 \times n_2$ squares sewn between them
1 Remove each edge between two same labeled regions
 if $l > 1$ **then**
2 Remove each degree two vertex between two aligned edges
 if $l > 2$ **then**
3 └ Remove each degree two vertex
4 Compute the region inclusion tree

The embedding modifications are achieved during the removal operations by the way we have explained in the previous section. For level 1 and level 2 maps only for the vertex embedding, and for level 3 by adding edge embeddings (when we remove a vertex between two non-aligned edges).

This algorithm is linear in number of image pixels (both in time and memory). Indeed, level 0 map has a dart number which is linear in pixel number (see Section 4.2) and other levels have less (or equal) darts. In order to construct each level, we only test each cell exactly once. But in order to construct level 3 map, we traverse each dart three times, and even if the complexity is linear it is not optimal. Moreover, we begin for level 0 map by creating many darts that are going to be eventually destroyed during the simplifications. This leads to an important and useless time

consumption due to memory allocation and de-allocation. Moreover, memory space required for the construction of level 0 map can be too important, and can prevent the construction of other map levels. To solve these drawbacks, we now propose an improvement of this basic algorithm.

7. Precodes to define an optimal extraction algorithm

The basic principle of precodes [13,21,22] consists in scanning the image from top to bottom and from left to right, with a 2×2 pixels window, and in executing an algorithm which depends on the local configuration of pixels in this window. We have used this principle, adapted it to our different map levels, and precisely studied what configurations have to be treated for each level. The main idea of our approach is to compute the map by an incremental and direct way, with only one image scan. Furthermore, precode algorithms depend only on the local pixel configuration, and create only the necessary darts.

7.1. Precodes

We scan the image from top to bottom and from left to right by using a 2×2 pixels window. During this scan, the *current pixel* is the bottom right pixel of the 2×2 window. A *precode* is a local configuration of pixels.

Definition 8 (precode). A *precode* is a partition of the set of the four pixels of the 2×2 window.

A precode fixes the pixels that have the same label and those that have different labels. Given a partition $\{p_1, \dots, p_k\}$ of the 2×2 window, two pixels belonging to the same p_i have the same label, and reciprocally two pixels belonging to distinct p_i and p_j have different labels. There exist 15 different precodes in two dimensions shown in Fig. 16. We draw a precode by affecting a different color to each partition element. Precode Nos. 13, 14, and 15 in Fig. 16 are *non-manifold* precodes.

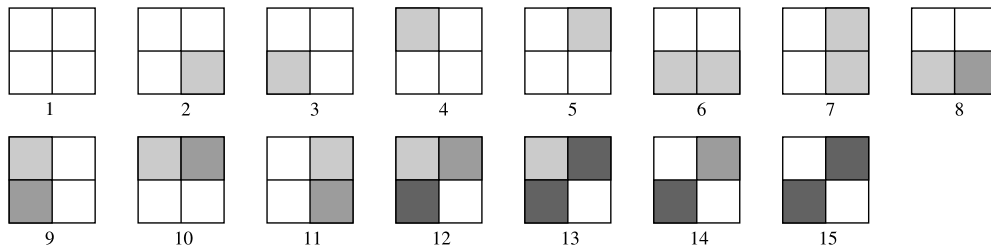


Fig. 16. The 15 different precodes.

Definition 9 (*non-manifold precode*). A precode is called *non-manifold* if and only if it exists an element of the partition which is not 4-connected. A precode which is not a non-manifold precode is called *manifold*.

The precode notion allows to define the operations to perform for each configuration met during the image scan. But we need another notion, which allows us to group several precodes in order to factorize similar processings: this is the *partial precode* notion.

Definition 10 (*partial precode*). A *partial precode* is a partition of any subset of the four pixels of the 2×2 window, where each element of the partition is a 4-connected set of pixels.

We can see in Figs. 17A and B two examples of partial precodes and a counter example in Fig. 17C. This last partition is not a partial precode, since one element of this partition (white pixels) is not a 4-connected set of pixels.

Precodes and partial precodes do not have exactly the same interpretation. Given a partition $\{p_1, \dots, p_k\}$ representing a partial precode, pixels not present in the partition can belong to any region. Two pixels belonging to the same p_i have the same label (as for a precode). But two pixels belonging to distinct p_i and p_j can have the same label or not, depending if $p_i \cup p_j$ is a 4-connected set or not:

- $p_i \cup p_j$ is 4-connected: the two pixels have the same label;
- $p_i \cup p_j$ is not 4-connected: the two pixels can have the same label or not.

Intuitively, a partial precode fixes the pixels that have the same label or those that have different labels, but only for 4-neighbor pixels. This is why a partial precode regroups several precodes (and also because pixels not present in the partition can belong to any region).

The partial precode shown in Fig. 17B only fixes that the bottom right pixel has a different label from its left pixel neighbor, and a different label from its top pixel neighbor. But it fixes no constraint between the top right pixel and the bottom left pixel, because the union of the two partition elements is not a 4-connected set. On the contrary, the two pixel sets of the partial precode shown in Fig. 17A cannot belong to the same region, because their union is a 4-connected set.

A partial precode regroups the precodes possible to be obtained by labeling pixels not present in the partial precode, and by possibly grouping non 4-connected element partitions in a same element. For example, partial precode shown in Fig. 17A

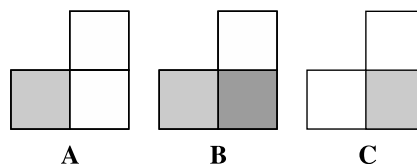


Fig. 17. Two partial precodes (A and B) and a counter example (C).

regroups precodes 3, 7, and 9, and partial precode Fig. 17B regroups precodes 2, 8, 11, 12, 13, 14, and 15.

7.2. Optimal and generic extraction algorithm

The extraction algorithm based on the precode notion is presented in Algorithm 2. It is generic since it can extract any map level, only by considering different precodes, and it is optimal since the map is extracted in a single image scan (we process each pixel exactly once) and the processings performed for each precode is done in a minimal number of operations. Moreover, this algorithm is simple because it can be summarized by an image scan where we execute the code corresponding to the current precode.

Algorithm 2 (*Optimal extraction of the level l map*).

Input : A labeled image I of $n_1 \times n_2$ pixels.
 l the level of the map we want to extract.
Output : The level l map corresponding to I .
1 $last \leftarrow$ Build the upper left border of the image
2 **for** $j = 1$ **to** $n_2 + 1$ **do**
 for $i = 1$ **to** $n_1 + 1$ **do**
 $last \leftarrow$ Process the code corresponding to precode (i, j)
3 Compute the inclusion tree

The image contains pixels having coordinates between $1, \dots, n_1$ in x and $1, \dots, n_2$ in y (other pixels are in the infinite region). Precode (i, j) is the precode having as current pixel (the bottom right pixel of the 2×2 window) pixel (i, j) . During the image scan, we overflow the image by one pixel in any direction. Indeed, when we process precodes $(1, j)$ (resp. $(n_1 + 1, j)$), $(i, 1)$, $(i, n_2 + 1)$), that processes the pixels on the left (resp. right, up, and bottom) of the image.

The first step of the optimal algorithm consists in building the upper left border of the image. Indeed, we have this invariant: the map corresponding to the pixels already scanned, is already built. This invariant allows us to be sure, at any time, that it exists a dart on the left and at the top of the current pixel. To satisfy this invariant for the first pixel of the image (pixel $(1, 1)$), we begin by building the upper left border of the image, as we can see in Fig. 18. With this initial map, each pixel in the first line of the image has an upper dart, and the first pixel of this first line has also a left dart.

During the image scan, we need to know the darts on the left and at the top of the current pixel, in order to locally modify the map already built. We respectively call these darts *last* and *up* (or l and u in figures), as we can see in Fig. 19. In this figure, we do not know if the darts *last* and *up* are sewn or if another darts between these two darts exist (this is represented by the dash lines and the question mark in the figure). But this has no influence on our algorithms since the map is only locally modified. We only keep the *last* dart since the *up* dart can be computed from this dart, starting from the *last* dart, and turning over the vertex until we find a dart not 2-sewn

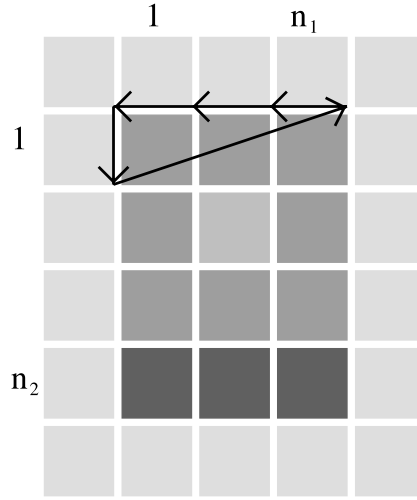


Fig. 18. The initial upper left border of an image with 3×4 pixels.

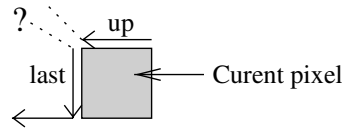


Fig. 19. Darts *last* and *up* by relation with the current pixel.

(note that during the extraction, the map is not closed, the darts in the border of the map are not 2-sewn, but after the extraction we have a correct map where all darts are sewn).

The *last* dart is initialized with the unique vertical dart of the initial border. Then, during the extraction, we execute the algorithm corresponding to the current precode. Each precode algorithm locally modifies the map, and return the vertical dart, to the bottom right of the current precode, which is the *last* dart for the next precode. Note that we do not need particular processings for the border of the image since it is plated on a cylinder (the last pixel of each line is the same as the first pixel of the next line). At the end of the algorithm, we have to compute the inclusion tree, as for the basic algorithm, by using the algorithm given in Section 8.

7.3. Precodes for level 1 map

The optimal extraction is generic. Indeed, to extract a particular map level, we only need to define what precodes we have to consider, and give the operations to perform for each of them. To find the precodes to consider for a particular level, we have to look at the operations to be achieved to obtain this level and see the consequences of these operations on the local precodes. To extract level 1 map, we need to remove each edge of the complete map between two pixels having the same label.

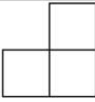
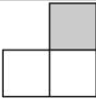
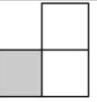
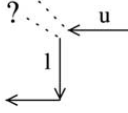
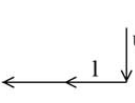
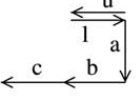
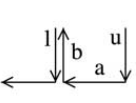
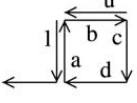
	l_1	l_2	l_3	l_4
Precode				
Before				
After				

Fig. 20. The four precodes to extract level 1 map.

Considering the current pixel and the map already built, we need to remove the left edge when the left pixel has the same label as the current pixel and we also need to remove the upper edge when the upper pixel has the same label as the current one. So there are four precodes to consider: they are all the ways of removing zero, one or two of these edges. These precodes are shown in Fig. 20.

In this figure, the first line gives the precode name, the second the precode itself, the third shows the current configuration of the map before executing the current precode algorithm, and the last line shows for each precode, the map we want to obtain after this execution. This map corresponds to the map obtained locally by the basic algorithm when we create the complete map and then we perform 1-removals. Maps represented in the figures are partial representation of the local configuration. For example, for the map after precode l_2 in Fig. 20, there are necessarily a dart 0-sewn to l , another 0-sewn and 1-sewn to u , but since this precode is local, we do not know where these darts are and we can not draw them.

In order to write the algorithm corresponding to each precode, we only have to look at the map before, compare it with the map we want to obtain, and write the sequence of operations which transform the first map into the second. We can see in Algorithm 3, which gives the algorithm of precode l_2 , that this code is easy to write. Indeed it is mainly composed of successive unsewn and sewn operations. We can verify in Fig. 20, that this algorithm actually transforms the current map into the map we want to obtain for the corresponding precode.

Algorithm 3 (Code corresponding to precode l_2).

Input: *last* and *up*, (x,y) coordinates of the new vertex.

Output: The “next” *last*.

$a \leftarrow$ a new dart; $b \leftarrow$ a new dart

$c \leftarrow \beta_1(last)$; 1-unsew(*last*)

```

1-sew(last,a); 1-sew(a,b)
1-sew(b,c); 2-sew(up,last)
mbedding of the vertex incident to b  $\leftarrow$  (x,y)
return a

```

The initial map is modified in order to include the current pixel. For that, we can modify all the darts of the initial map, even *last* and *up* darts. We only need at the end of the algorithm to return the dart which will be the next *last* dart for the next precode. The definition of the four precode algorithms (given in annex) totally defines the optimal extraction of level 1 map, which during the image scan, tests what is the current precode, and only executes the corresponding algorithm.

7.4. Precodes for level 2 map

To extract level 2 map, we have to remove each degree 2 vertex between two aligned edges. But since we construct the map in a single scan, we also need to remove each edge between two pixels having the same label (simplification to obtain level 1 map). If we look at the 15 different possible precodes, we can see that there are only two cases with aligned edges that are incident to a degree two vertex: the two precodes given in Fig. 21.

To extract level 2 map, we need to find what is the current precode between these two precodes plus the four level 1 precodes. We can note that these two level 2 precodes are included in precodes l_2 and l_3 . To find what is the current precode, we first test if it is a level 2 precode (f_1 or f_2), and when it is not the case we search in the level 1 precodes. We can see in Fig. 21 these two precodes, the initial maps and the maps we want to obtain. As for level 1 map, the algorithms corresponding to these two precodes are simple as we can see for precode f_1 in Algorithm 4.

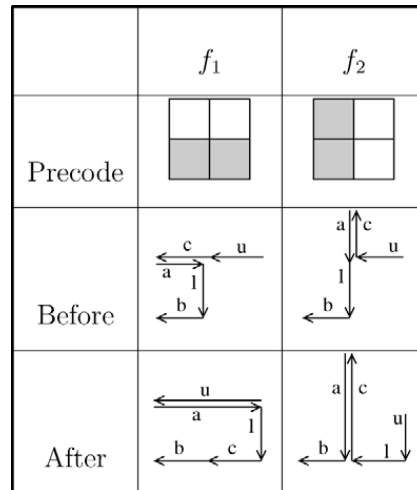


Fig. 21. The two additional precodes to extract level 2 map.

Algorithm 4 (Code corresponding to precode f_1).

Input: *last* and *up*, (x, y) coordinates of the new vertex.

Output: The “next” *last*.

$a \leftarrow \beta_0(\textit{last}); b \leftarrow \beta_1(\textit{last})$

$c \leftarrow \beta_1(\textit{up}); d \leftarrow \beta_1(c)$

1-unsew(*last*); 2-unsew(*a*)

0-unsew(*c*); 1-unsew(*c*)

1-sew(*last*, *c*); 1-sew(*c*, *b*)

1-sew(*up*, *d*); 2-sew(*up*, *a*)

Embedding of the vertex incident to $c \leftarrow (x, y)$

return *last*

7.5. Precodes for level 3 map

For this level, we have to remove each degree 2 vertex. By looking at the 15 precodes, we can note that there are only four cases where such a removal has to be achieved. These precodes are given in Fig. 22. We have not represented the edge embeddings in this figure, but they are updated during the sewing, unsewing and 0-removal operations, by the way explained in Section 5. We can see in Algorithms 5 and 6 the two algorithms that correspond to precodes t_1 and t_3 .

Algorithm 5 (Code corresponding to precode t_1).

Input: *last* and *up*, (x, y) coordinates of the new vertex.

Output: The “next” *last*.

$a \leftarrow$ a new dart; $b \leftarrow$ a new dart

$c \leftarrow \beta_1(\textit{last}); 2\text{-sew}(\textit{up}, a)$

Add geometrical point incident to *last* at the end of the edge embedding of *a*

0-unsew(*last*); 1-unsew(*last*)

1-sew(*a*, *last*); 1-sew(*last*, *b*)

1-sew(*b*, *a*); 0-sew(*c*, *up*)

Embedding of the vertex incident to $b \leftarrow (x, y)$

return *last*

Algorithm 5 is similar to previous algorithms. The unique difference is the explicit embedding modification of the edge incident to *a*. Indeed, since we remove the vertex incident to dart *c*, we need to keep the corresponding geometrical point in the edge embedding. Note that the edge incident to *a* has initially no embedding because only closed edges (for β_2) can have such an embedding. So when we add the geometrical point incident to *last* at the end of the edge embedding of *a*, this creates an edge embedding made of this unique geometrical point.

Algorithm 6 (Code corresponding to precode t_3).

Input: *last* and *up*, (x, y) coordinates of the new vertex.

Output: The “next” *last*.

```

 $a \leftarrow \beta_0(last); b \leftarrow \beta_2(a); c \leftarrow \beta_0(b)$ 
 $d \leftarrow \beta_2(c); t_1 \leftarrow \beta_1(b); t_2 \leftarrow \beta_1(d)$ 
Add geometrical point incident to last at the end of the edge embedding of a
Merge edge embedding of d at the end of edge embedding of a
0-unsew(b); 1-unsew(b); 2-unsew(b); delete b
0-unsew(d); 1-unsew(d); 2-unsew(d); delete d
0-unsew(last); 1-sew(up,last)
2-sew(a, c); 1-sew(a, t_2); 1-sew(c, t_1)
Embedding of the vertex incident to last  $\leftarrow (x, y)$ 
return up

```

The algorithm of precode t_3 is a little more complex since this is the unique case where the two edges along the removed vertex (edges incident to *a* and *b*) can both have an edge embedding. In order to keep the geometry, we first add the geometrical point incident to *last* at the end of the edge embedding of *a*, then we merge the edge embedding of *d* at the end of edge embedding of *a*. This merging has to consider the special case where one embedding is empty, and also the case where the two embeddings have not the same orientation. In this case, we need to reverse one of the two embeddings before the merging (see Section 5 for more explanations).

The definition of the four algorithms, corresponding to the four level 3 precodes, allows us to extract level 3 map in a single image scan, by the way of the generic extraction algorithm. First we check if the current precode is a level 3 precode, if not if it is a level 2 precode and otherwise we find the current precode in the level 1 precodes. Note that we only have ten cases to consider amongst the 15 possible precodes. This is due to our simplification levels that allows us to factorize immediately similar processings.

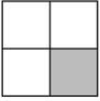
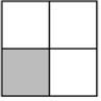
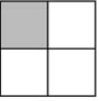
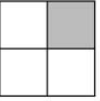
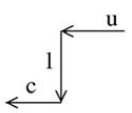
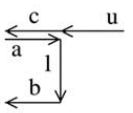
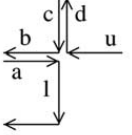
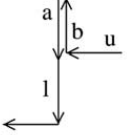
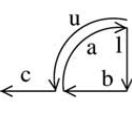
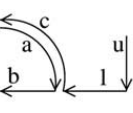
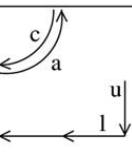
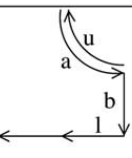
	t_1	t_2	t_3	t_4
Precode				
Before				
After				

Fig. 22. The four additional precodes to extract level 3 map.

8. Inclusion tree

The computation of the inclusion tree is the last step of our two extraction algorithms, after the extraction of a level l map. This tree is necessary to keep a relation between each different connected components of the map. The inclusion relation used here is the one presented in Section 3, but we only represent direct inclusions (relations we cannot retrieve by transitivity). The inclusion tree is always rooted with the infinite region. We keep relations between a node and its sons and with its father, in order to be able to go through the tree from root to leaves and from leaves to root.

There are many possibilities to represent inclusion relations: we can give for each region the list of regions adjacent to an internal border; we can add fictive edges between all the internal and external borders of each region; we can give for each region one region per connected set of included regions (one region for each hole). . . Our choice allows us to obtain an inclusion tree and not a forest, where each region is present. This is interesting when we need to traverse each region: so it is enough to traverse the inclusion tree without considering the map. Otherwise, this solution is not a good one to easily check if two regions can be merged or not, only by looking at the inclusion tree. But of course, each solution has its own advantages and disadvantages, and our proposed solution need to be reconsidered for each application.

To compute the inclusion tree, we traverse all the darts of the map connected component by connected component. Indeed, a connected component is composed of regions that all have the same father. In order to retrieve the different inclusions we need some particular information:

- Each dart is linked with its belonging region noted $region(d)$: $\forall d$ and d' , $region(d) = region(d')$ if and only if d and d' belong to the same region.
- Each region R knows one dart of the map, belonging to this region. This dart has to be the dart with the corresponding vertex in the upper left of this region. This property ensures us we can easily retrieve the region containing R . We call such a dart the *representative* of its region.
- We also have to know the list of all the image regions, without the infinite region. This list must be sorted so that a region R_i is smaller than another region R_j if and only if R_i is met before R_j during the image traversal from top to bottom and from left to right.

These properties are given by our extraction algorithms. For the basic algorithm, we need to perform a first image scan in order to initialize the list of sorted regions. The dart labeling is achieved during the construction of level 0 map, since simplification operations do not lead to modification of a dart label. The representative dart of each region is fixed during the initial image scan, and possibly modified during the simplifications. Indeed, when the representative dart of a region is removed, we choose as new representative a neighboring dart of the former one. All these operations can be achieved in linear complexity, but lead nevertheless to additional time for the initial image scan.

For optimal algorithm, the list can be computed at the same time as the extraction since we use the same image scan. During this scan we can fix the representative dart

of each region when we meet a region for the first time. The dart labeling is realized at the same time. When we meet a new region, we label the new darts with this new region (this case can occur only for precodes l_4 and t_1). For the other cases, we copy the label of the same region existing darts (we can verify that for each precode except l_4 and t_1 , the current pixel has a neighbor with the same label).

Algorithm 7 (*Inclusion tree computation*).

Input : The map M of a labeled image I .
The list L of regions of the image I .
Output : An inclusion tree T .
Unmark each region of L
Add a node corresponding to the infinite region as root of T
1 **foreach** region R of L not marked **do**
 $d_{init} \leftarrow \text{representative}(R)$
 $father \leftarrow$ the node corresponding to $\text{region}(\beta_2(d_{init}))$
2 **foreach** dart d of the connected component incident to d_{init} **do**
 if $\text{region}(d)$ not marked **then**
 Add a node corresponding to $\text{region}(d)$ in T
 Set this node as son that $father$ in T
 Mark $\text{region}(d)$
return T

The computation of the inclusion tree is presented in Algorithm 7. This algorithm's input is a map, of any level, and a list of regions, sorted as explained above, and its return is the corresponding inclusion tree. This algorithm is made of two loops. The first loop traverses unmarked regions of list L . Indeed, when a region is marked, this is because it was already considered and placed in the inclusion tree. For an unmarked region, we initialize the first dart of the traversal of the second loop with the representative of this region (called d_{init}). The region which contains this region, is inevitably the region of dart β_2 sewn to dart d_{init} . Indeed, list L is sorted so that, when we meet the first region of a connected component, we know that this region has no region belonging to the same connected component at its top and on its left. So $\beta_2(d_{init})$ is obviously a dart of the region containing the current region, and each region belonging to the same connected component is also included in this same region.

The second loop of the algorithm traverses each dart of the connected component incident to d_{init} , and for each region not already met, it puts this region as son of $father$. Since we traverse each dart of the connected component, we are going to consider exactly all regions of this component, and not the included regions which will be treated during a next round of the first loop.

We can see in Fig. 23 a labeled image and its inclusion tree. The sorted list of regions is A, B, C, D, E, F (first sorted by y coordinate then by x). So the first region considered is A which has for representative the dart that contains the point at the top left corner of this region. d_{init} is initialized to this dart and so $father$ to the infinite

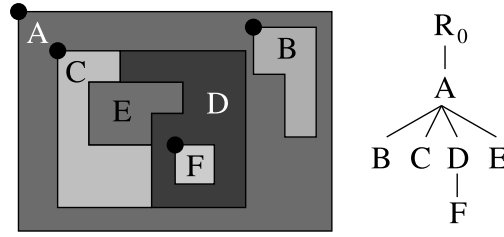


Fig. 23. A labeled image and its inclusion tree.

region. Then, the second loop of the algorithm traverses the connected component of A but here there is no region adjacent to A (not included) so we only put A as son of R_0 .

The second region considered is B , its representative dart contains the point at the top left of this region (d_{init}) and so $\beta_2(d_{\text{init}})$ belongs to region A . Region B is put as son of A in the inclusion tree. As for region A , there is no region adjacent to B so the second loop marks only darts of B . The next region is C , d_{init} is initialized to the dart that contains the top left point of C . Here, the second loop traverses regions C , D and E because they are in the same connected component. Each of these regions is put as son of A in the inclusion tree (that corresponds to the notion of inclusion used in this work). The last region considered is F (because others regions of the list are already marked) which is put as son of D in the tree.

The complexity of this algorithm is linear in number of darts of the map, and in number of regions of the image. Since the number of regions is lower than the number of darts, the complexity is so linear in number of darts of the map. Indeed, we traverse each dart exactly once, because we traverse the connected components one by one, and two different connected components are inevitably disjointed. We also traverse each region only once because we traverse the list of regions and mark each treated region. This algorithm works for every map level. Moreover, its execution time decreases when the level increases, even if the global complexity does not change, because the number of darts to traverse decreases more and more for higher levels.

9. Comparison with similar existing approaches

In order to compare our solution with existing approaches, we are going to describe more precisely the two that are very near to our solution: the topological map of Domenger et al. [19,11] and the topological graph of frontier (TGF) of Fiorio et al. [21,22]. These two solutions are both based on combinatorial maps, and both define the minimal model that represents an image partition. For these reasons, the three methods (solution of Domenger et al., Fiorio et al. plus the new method presented in this paper) have the same topological representation. The main differences are about the embedding and about the algorithms used to extract each model from an image.

9.1. The topological map of Domenger et al.

We present here the definition given in [10]. Combinatorial maps used in this work are not exactly the same as the ones presented in this paper (cf. Section 2). Differences are only about the name of the applications. They use a combinatorial map with σ a permutation that gives for a dart the next dart of the same vertex, and an involution α which connects the two darts incident to the same edge. α is equal to our β_2 , and σ can be defined by $\beta_2 \circ \beta_1$. Indeed, to represent a two-dimensional combinatorial map, we can choose to encode the permutation around vertices or the one around faces. Our notation can be extended more easily in higher dimension than their notation since we can add any β_i that puts in relation two i -cells.

In order to be addressed, each face is associated with a unique label. But contrary to our label, two darts have the same label if and only if they belong to the same face orbit. So an internal contour does not have the same label as its corresponding external contour. These labels are used to represent the inclusion relations by using two functions called *prnt* for father and *chld* for children. An external contour has no father (which is represented by the particular value 0 for the function *prnt*), and an internal contour has no child. At last, another function called β gives a canonical dart for each label. This last function is equivalent to our representative dart.

Comparing this solution to our inclusion tree, we can see that the two solutions are very near. We use only a label for each region that allows us to add other information about region (mean grey level, number of pixels...). But we cannot say in constant time if there is a boundary between a region and its father region, since we need to traverse the face orbit. This information is given immediately by the solution of Domenger et al. But there are some cases where their solution need more operations than ours. For example when an edge removal leads to the creation of an internal border, we only need to modify the position of one region in the inclusion tree, but they need to add a new label and to traverse the face orbit in order to put this new label onto all the darts of the new internal contour.

They define the topological map by introducing the notion of *boundary points* which are pointels with more than two incident linels that belong to a boundary. The boundary points are then connected with edges when there is a contour linking them in the image. We can see that this leads to the same map as our solution since we remove all degree two vertices. At the end, we thus obtain only points that have more than two incident linels (if we except the cases of loops).

Their solution for the embedding of the topological map consists in a matrix that represents all interpixels cells (pointels and linels) that belong to the image boundaries, and which is called *boundary image*. They use an efficient coding in order to compress the memory necessary to encode this matrix. But they also need to keep some additional information in order to be able to retrieve a dart that contains a given pointel, and to retrieve, given a dart, the corresponding one-dimensional curve in the boundary image. This is made with two additional data structures that are correspondence tables in the both ways. There are so many additional structures that need to be updated according to each operation that modifies the map.

This leads to time consumption since we have more structures to update, and leads to more complex algorithms since we need to study how an operation on the map possibly modifies other structures. Comparing to our solution, we only have the map plus an inclusion tree with an unique labelling function that gives for each dart its belonging region. Moreover, our embedding can be updated automatically by the sewing and unsewing operations since it is quite independent of the map, which is not the case for their solution.

The last difference is about the algorithm used to extract the topological map. They use a contour tracking algorithm that follows the boundaries in the image. They start from an initial loop that represents the boundary between the image and the infinite region. Then, they select boundary points by looking at the number of pixels having different labels around each point. These points are inserted into the boundary images by splitting existing edges. Then, edges are inserted between boundary points by following the outer contours. Last, it is necessary to scan each region to look for possible new included connected components. Compared with our approach, this algorithm is more complex since it needs a more complex image scan and several distinct steps. Another advantage of our solution is that we can choose to use the optimal algorithm with precodes in order to optimize time, or to use the basic algorithm to give a simple algorithm easy to develop. Moreover, our optimal solution with precodes is more efficient as we can see in Section 10. At last, our model and our extraction algorithm can be extended more easily in higher dimension since we do not have many different structures and our algorithm is only based on the removal operations (which are defined in any dimension), and since we have a solution which is a lot simpler than their extraction algorithm.

9.2. The topological graph of frontier of Fiorio et al.

We present here the definition that we can find for example in [22]. They see their model as an extension of the RAG (region adjacency graph) with multi-adjacency relations encoded by a two-dimensional combinatorial map. This leads to a difference for the representation of the structure, where edges are drawn between the barycenter of regions instead of our representation where edges are drawn on the image boundaries. But this is only a graphical representation and not a main difference between the two approaches. They use the same combinatorial maps notations as Domenger et al. with α and σ .

This is the definition of the FTG that we can find in [22]:

Definition 11. A FTG is an extended combinatorial map $G(V, D, \alpha, \sigma)$ where V is the set of vertices of the graph, D the set of darts, α a permutation on D and σ an involution on D such that:

- (1) each region is represented by one and only one vertex;
- (2) a dart e is always incident to a vertex R of V . The notation e^R denotes the incident vertex of e ;

- (3) an edge of the graph is a non-ordered pair $(e, \sigma(e))$ where e belongs to D . Let E be the set of edges, $(e_1^R, e_2^{R'}) \in E$ if and only if there exists a frontier between R and R' . Thus an edge represents a frontier. A dart symbolizes the frontier “as seen from” the region to which it is incident;
- (4) the cycles of permutation α correspond to a contour of a given region and respect the order induced by the sequence of frontiers making the contour;
- (5) inf is a particular vertex of the graph symbolizing the exterior of the image;
- (6) to each vertex is associated the list of the cycles of α related to the contours of the region represented by this vertex. By convention, the exterior contour will always be the first of the list.

This definition defines directly the minimal combinatorial map that represents an image (item 3, each edge corresponds exactly to a frontier). Moreover, it fixes the relation between regions (the vertices V) and the combinatorial map (item 2). Item 4 is necessary in order to guaranty that the order of the edges in the map is the same as the order of frontiers in the image. At last, item 6 is equivalent to our inclusion tree. Indeed, each region has an external contour (the first element of the list), then possibly some internal contours that are holes in the regions. This implicit encoding is equivalent to the explicit function that puts in relation the contours in the work of Domenger et al. We can note that this definition is more complex than ours, since the minimal model is directly defined. Moreover, we can do the same criticism as for the previous approach. The fact that the combinatorial map is linked with other additional structures, leads to more complex algorithms when we need to update the model, and thus leads more important complexity. Moreover, this TGF is only a topological model and does not keep the region geometry.

To extract the TGF, they use an algorithm based on precodes (this algorithm was our initial inspiration to our algorithm). But since TGF is a unique structure, they need to build at the same time the map and the relations between frontiers (which are equivalent to our inclusion tree). This leads to more complex algorithms since they do not only need to give operations that modify the map, but also operations that update the inclusion tree. They keep a list of pending contours (contours that touch the current line of the image scan) and updating the type of contour (internal or external contour) depending of the current precode and a test to see if darts belong to the same orbit or not.

For these reasons, the TGF is difficult to extend in higher dimension. First the definition is complex to extend since it is based on the order of boundaries in the image, and this order is not the same in three dimensions for surfaces. Moreover the extraction algorithm needs to consider all precodes (15 in two dimensions and 4140 in three dimensions) to build the inclusion tree, contrary to our approach with simplification levels that allows us to factorize many cases (only 10 precodes in two dimensions and 129 precodes in three dimensions). Moreover, our computer software which extracts our model is more efficient than their software since we made less operations (see Section 10).

10. Experiments and analysis

We have implemented our optimal extraction algorithm based on precodes for the three different map levels. We have used the embedding presented in Section 5 (for levels 1 and 2 a vertex embedding, and for level 3 a vertex embedding plus an open edge embedding). Moreover, we have compared our software with the software of Brun⁹ which implements the method first proposed by Domenger et al. [11,19], and the software of Fiorio which constructs the TGF [21,22]. Experiments were made on a personal computer with a 1500 MHz CPU with 256 Mb of memory and a Linux system.

We choose to represent each dart by a structure where each β_i relation is simply a pointer to the dart i -sewn. There are many different ways to represent maps in computer memory, and the choice of one of them depends on what we would like to prioritize: memory cost or time consumption. Our choice clearly prioritizes time since we access to each β_i of a given dart in direct access, and we can also easily modify all the β_i relation without constraint. But of course this is to the detriment of memory space occupation. This choice needs to be studied more precisely for each particular need of a given application, but this question could not be answered in a general way. Moreover, our software is developed in C++ without particular optimization.

We have used for our experiments the six labeled images well known in image processing shown in Figs. 24 and 25. These figures show, firstly, the labeled image (one color by different label), then the maps obtained which represent the interpixel boundaries. Note that the segmentation algorithm used to obtain the label image is very simple (only based on the mean grey value of regions). We do not try to obtain here a good segmentation but only to compare the different map levels on the same images. We can find in the legend of the figures the size of each image in pixels and the number of regions obtained by the segmentation.

First we only compare the three simplification levels. We can see in Fig. 26 the evolution of memory space, number of darts and number of vertices for level 1, 2, and 3 maps, in level 1 percentage. The memory space evolution shows that the gain is very significant. It is about 36% between level 1 and level 2, and about 34% between level 2 and level 3 (this is a mean between the six considered images). We thus obtain that the memory gain is about 58% between level 1 and level 3. This is not a surprise since a lot of darts are removed during the successive simplifications.

But the memory gain is not the unique advantage of level 3 maps. Indeed, if we look at the number of darts, we can observe that level 3 map has only about 30% of darts of level 1 map. This significant difference leads to decrease the computational times for algorithms defined on level 3 map due to a more direct access to the adjacency information.

We can see in Table 1 the extraction time in seconds, and the memory space occupation in mega-bytes for levels 1, 2, and 3 by using our method, and for the two other softwares of Brun and Fiorio. First, we can notice, on these different extraction examples, that the execution time of our algorithm is about the same to extract any

⁹ Thanks to Luc Brun for his help which allowed us to use his software and to realize this comparison.

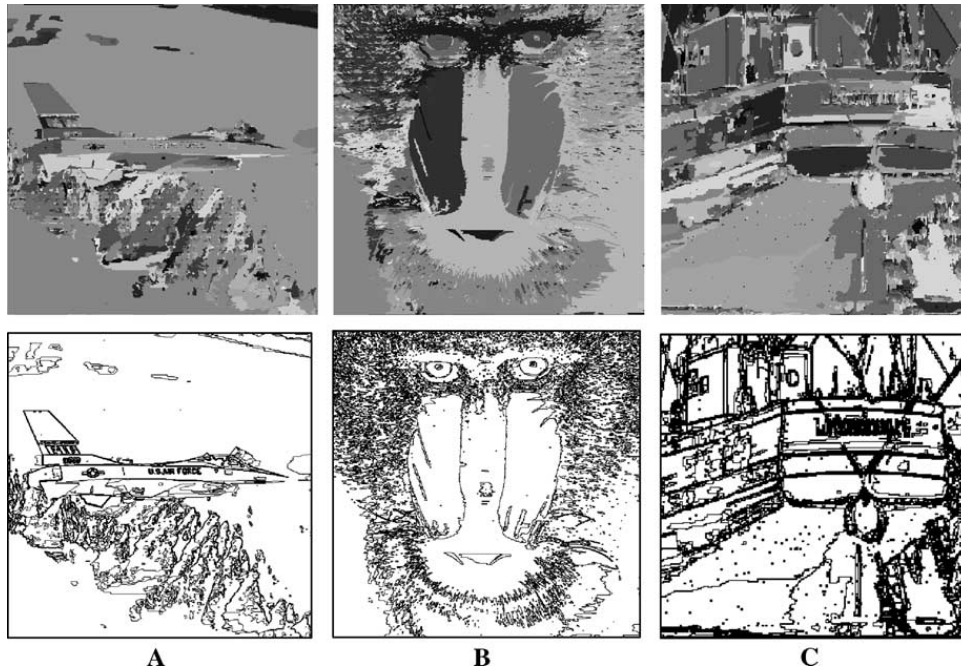


Fig. 24. (A) Airplane, 512×512 pixels, 6361 regions. (B) Baboon, 512×512 pixels, 6705 regions. (C) Cornouaille, 256×256 pixels, 5410 regions.

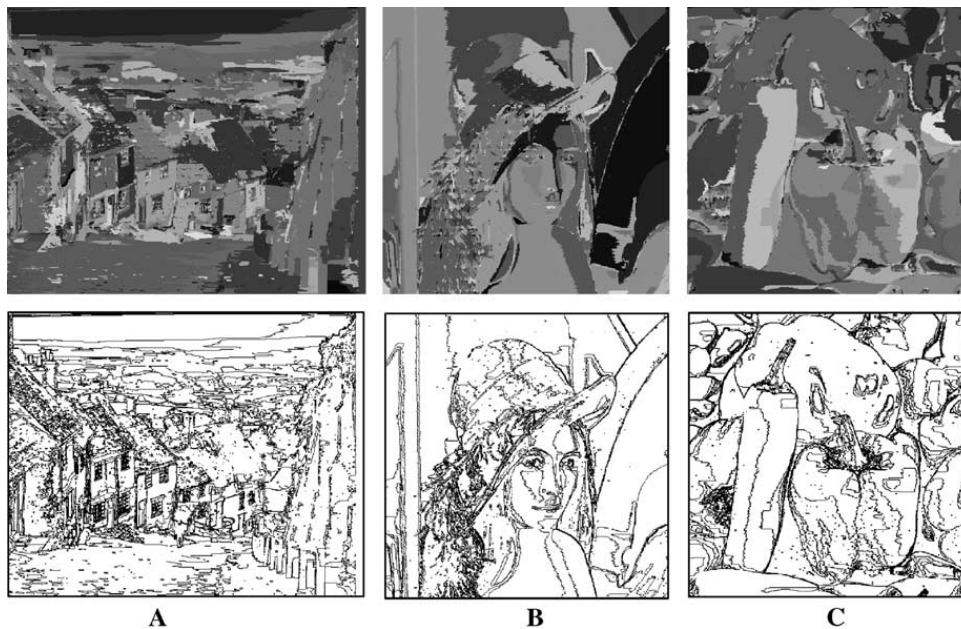


Fig. 25. (A) Goldhill, 720×576 pixels, 8006 regions. (B) Lena, 512×512 pixels, 6198 regions. (C) Peppers, 512×512 pixels, 5765 regions.

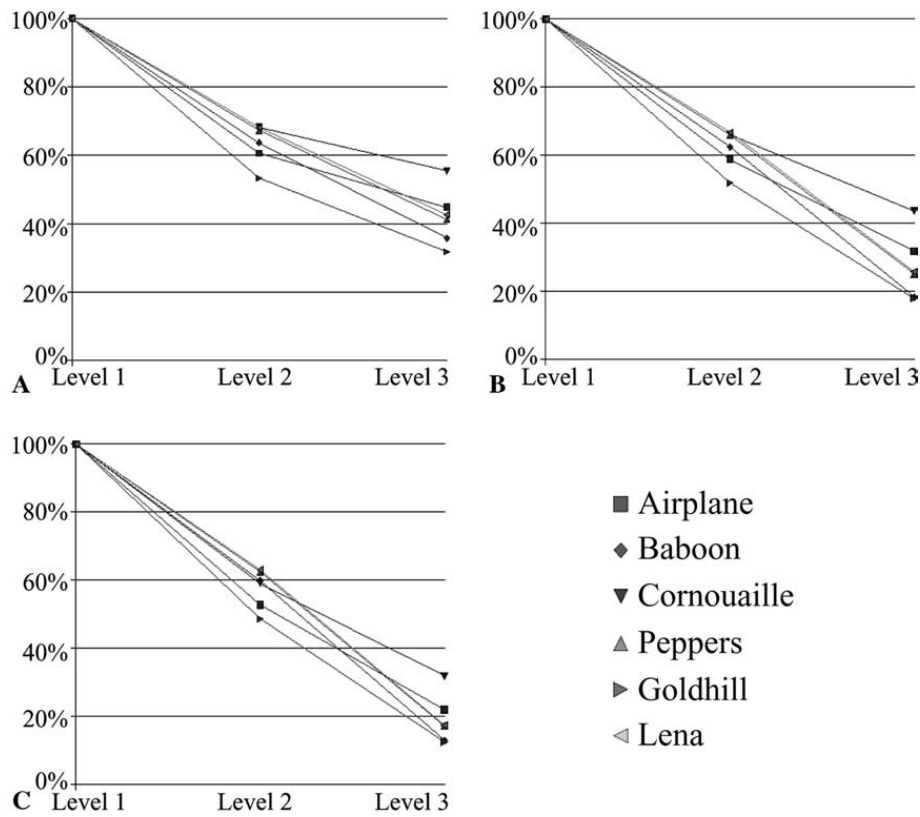


Fig. 26. Characteristics evolution for the three different map levels, in level 1 percentage. (A) Memory space. (B) Number of darts. (C) Number of vertices.

Table 1

Comparison of extraction time (in seconds) and memory space occupation (in mega-bytes) for each image.

Image	Level 1	Level 2	Level 3	L. Brun	C. Fiorio
Airplane	0.08 s	0.06 s	0.05 s	0.71 s	0.25 s
	3.85 Mb	2.34 Mb	1.6 Mb	1.64 Mb	1.49 Mb
Baboon	0.11 s	0.08 s	0.09 s	6.96 s	0.35 s
	6.11 Mb	3.9 Mb	1.89 Mb	2.75 Mb	1.41 Mb
Cornouaille	0.05 s	0.03 s	0.04 s	0.27 s	0.20 s
	2.46 Mb	1.69 Mb	1.3 Mb	1.36 Mb	1.3 Mb
Goldhill	0.14 s	0.11 s	0.11 s	2.7 s	0.43 s
	7.97 Mb	4.26 Mb	2.24 Mb	2.76 Mb	3.62 Mb
Lena	0.08 s	0.08 s	0.08 s	0.97 s	0.35 s
	4.36 Mb	2.98 Mb	1.66 Mb	1.79 Mb	1.4 Mb
Peppers	0.09 s	0.07 s	0.08 s	1.5 s	0.31 s
	4.36 Mb	2.94 Mb	1.61 Mb	1.63 Mb	1.37 Mb

simplification level. Indeed, all precode algorithms are similar (mainly made of some sewing and unsewing operations) and so take about the same time. The execution time mainly depends on the image size and on the number of regions. Indeed, when there are less regions, there are also less darts and the computation of the inclusion

tree requires less time. We can also see that for some images (for example airplane or goldhill) the execution time decreases for higher levels. This is due to the less significant number of memory allocation and disallocation for higher levels.

Note that these times are the full execution times including initialization, image reading, topological and geometrical part and inclusion tree computation. This time is divided in about 7% for the inclusion tree computation, 26% for the geometrical part and 37% for the topological one. The remaining 30% are for the image scan and the current precode test. These percentages are approximative since they depend on the images.

Now if we compare our solution with the two existing ones, we can see that we are really more efficient in computation times, and that the three methods are quite similar for memory space occupation (of course without considering level 1 and level 2 maps). For memory space, this is due to the fact that the three maps are similar, and differences only concern the embedding. The computation time is larger for the method of Brun since its extraction algorithm is a contour tracking which makes much more operations than our method. The difference with the method of Fiorio is probably due to the inclusion tree computation which is made in the same time as the map extraction (cf. Section 9).

These results are a good surprise since our first goal was not to improve existing methods but only to revisit these methods to propose a definition that could be easily extended in higher dimension. Our works made in order to simplify algorithms lead to simpler computer software and thus also more efficient. This shows another advantage of our approach.

In order to compare more precisely our simplification levels, we have made some experiments with artificial regular images. This allows to show the extreme cases, for example when all regions are made of squares or otherwise when all regions have only stair boundaries. For that, we have made four types of images:

- (1) Square images where each region is a square of length l .
- (2) Line images where each region is a line, indeed a rectangle with its height equal to the height of the image and its width equal to l .
- (3) Column images which are the same as line images but with a 90° rotation.

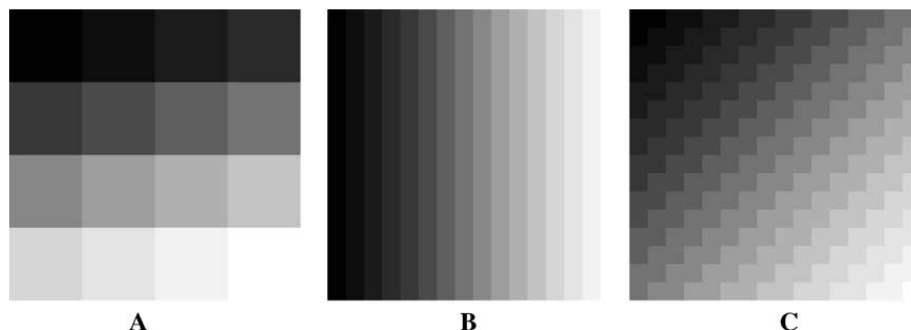


Fig. 27. (A) Square image, 128×128 pixels with $l = 32$ so with 16 regions. (B) Column image, 128×128 pixels with $l = 8$ so with 16 regions. (C) Stairs image, 128×128 pixels with $l = 8$ so with 16 regions.

- (4) Stair images: each region is a 45° oriented rectangle. So its borders are made of successive stairs of length l .

In order to obtain regular images, we have worked with a 128×128 pixels image, and only use for regions the lengths which are a power of two. For each type of image, we generate each image starting with l equal to 1 and multiply by two until length equal to 128. Thus we can observe the relation between the length of regions and the memory space occupied by each level. We can see in Fig. 27 some examples of obtained images. Since images are regular, we can compute, given the length of each region, the number of regions. For example, when we have a square image with 128×128 pixels, the number of regions is $(128 \times 128)/l^2$.

We give in Table 2 the memory space required for square images for each size of squares between 1 and 128. We can observe that level 2 and level 3 are very near since almost all region boundaries are straight line segments (in fact all boundaries except for the four corners of the image).

In Table 3, we make the same experiment with column images, for each length between 1 and 128. Again, level 2 and level 3 are very near since almost all region boundaries are straight line segments. We obtain exactly the same results for line images since the maps are exactly the same in both cases.

But if we look in Table 4 where we make the same experiment with stair images, for each length between 1 and 128, we can see that it is the opposite. Indeed, it is now

Table 2

Comparison of memory space occupation (in kilo-bytes) for each length of squares between 1 and 128

Length	1	2	4	8	16	32	64	128
Nb regions	16,384	4096	1024	256	64	16	4	1
Level 1	2462	1278	662	348	182	100	60.1	40.1
Level 2	2462	628	167	50.5	13.3	3.67	1.10	0.39
Level 3	2461	627	166	50.3	13.1	3.45	0.89	0.21

Table 3

Comparison of memory space occupation (in kilo-bytes) for each length of columns between 1 and 128 (exactly the same results for line images)

Length	1	2	4	8	16	32	64	128
Nb regions	128	64	32	16	8	4	2	1
Level 1	1314	672	351	190	110	70.2	50.1	40.1
Level 2	34.1	17.1	8.62	4.37	2.25	1.18	0.65	0.39
Level 3	33.9	16.9	8.39	4.14	2.01	0.95	0.42	0.21

Table 4

Comparison of memory space occupation (in kilo-bytes) for each length of step stairs between 1 and 128

Length	1	2	4	8	16	32	64	128
Nb regions	128	64	32	16	8	4	2	1
Level 1	1314	672	351	190	110	70.2	50.1	40.1
Level 2	1294	327	83.7	21.9	6.07	1.89	0.73	0.39
Level 3	286	79.4	23.7	7.78	2.85	1.13	0.44	0.21

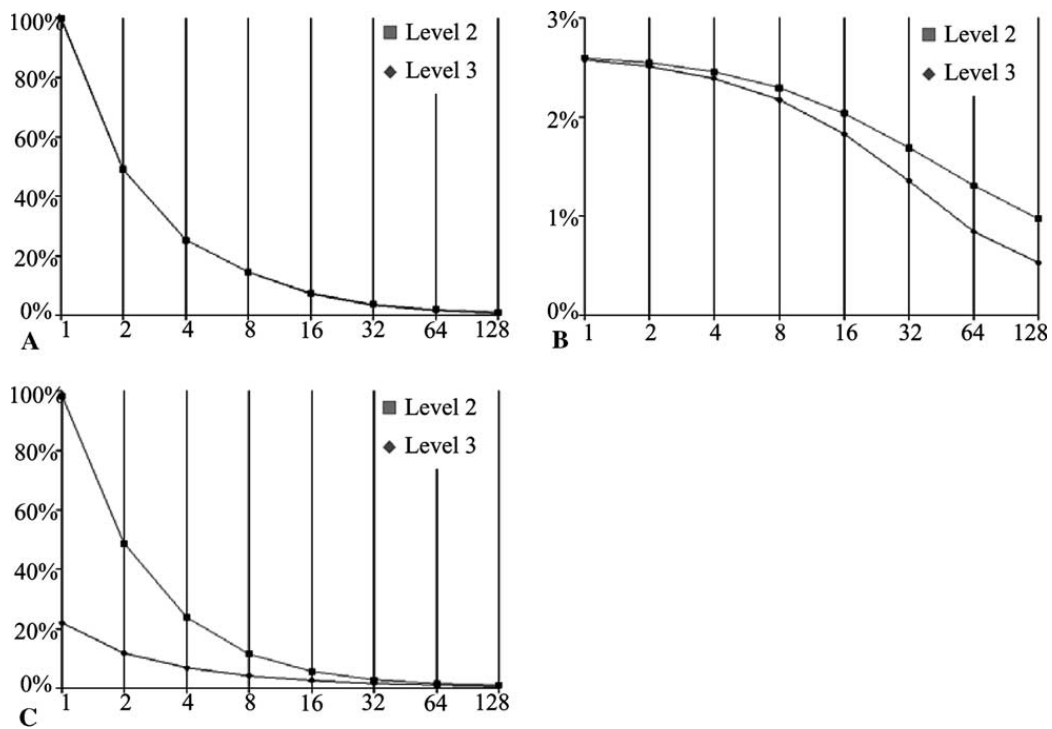


Fig. 28. Memory space evolution for level 2 and level 3 maps, in level 1 percentage. (A) Square images. (B) Line (resp. column) images. (C) Stair images.

level 1 and level 2 that are mainly near, since almost all region boundaries are not straight line segments. These boundaries become straighter and straighter when l becomes larger, and so level 2 becomes closer to level 3 when l is large.

We can see in Fig. 28 three curves that show the memory space evolution for each type of images in function of the length of regions. On each figure, there are two curves that show the memory space occupation for level 2 and level 3 map, in percentage of the memory of level 1 map. Curves shown in Fig. 28A show that level 2 and level 3 are very near for square images. It is also the case for line images (see Fig. 28B), even if we can see differences between the two curves, but this is due to the y-axis which is very precise (between 0 and 3%). Indeed, even when the length of lines is equal to one, regions have very large straight boundaries and so the memory occupied by level 2 and level 3 is much smaller than the memory required by level 1. Otherwise, we can see in Fig. 28C that memory occupation of level 3 is much smaller than the one of level 2, even if the difference decreases when the length of the stairs increases.

These four experiments with regular images show that the difference between level 2 and level 3 can be very significant or not, depending on the images content. But in general, if we consider “real images,” boundaries are often not straight and the difference between level 2 and level 3 memory occupation is significant. This is confirmed by our experiments made with real images (see Table 1).

11. Conclusion

In this paper we have presented a model for two-dimensional image representation, and have given an optimal extraction algorithm which computes this model from a labeled image.

Several works have already studied the use of combinatorial maps to represent interpixel boundaries of two-dimensional images. But our approach is original due to the new notion of simplification level which allows to give a formal and constructive definition of the topological map. Moreover, these levels can be extended easily in higher dimension [1,16], whereas for other approaches this extension is much more difficult, even not possible.

Of course, some problems still remain in higher dimension since we do not know the topological classification of objects ([30] discusses about these problems and a possible solution). Nevertheless, we can define the topological map in any dimension and use this tool in order to try to characterize discrete objects. For that, we are studying how to compute homology groups on topological map. Another idea is to add fictive cells in the topological map in order to keep each i -cell homeomorphic to an i -sphere. This can be achieved without particular problems, thanks again to our progressive simplification levels, where we can control if a removal operation leads to a disconnection. Then we need to study how obtained objects can be interpreted and how modification algorithms have to process these particular elements.

The topological map is the last simplification level. We have proved that this map is *complete* by showing that no information is lost during the different simplifications, *minimal* because we cannot remove anything in this map without changing the topology, and *unique* because this map only depends on the topology of the image and so two topological equivalent partitions have the same level 3 map. For these reasons, topological map is a good model for image processings. Indeed, it allows to retrieve most of the information which may be required by an image processing algorithm with a low computational cost. Thus we can consider many different image processing algorithms that only work on topological map and that use many kind of information.

We also have presented in this paper an optimal extraction algorithm which uses the notion of precode. It can extract any map level in one image scan. Moreover this algorithm is simple to implement since it is based on the definition of a limited number of operations to do, according to a limited number of local configurations. We have given for each simplification level the precodes required to extract the corresponding map and the algorithms associated to each of them. This has allowed us to show that there is a correspondence between the levels and the number of precodes required. At last, we have shown in different experiments that the topological map needs less memory than other levels and we have computed the gain observed comparatively to each level. Moreover, we have compared our solution with other existing approaches and showed that our software is more efficient to construct the topological map than other ones. This is due to our optimal extraction algorithm and to our effort to simplify the definition of our model.

We are currently working on the use of our model to propose new segmentation algorithms or to improve existing techniques [6]. The first results are encouraging

and let us hope to achieve interesting solutions. But our main goal is to extend this work in three dimensions and so to propose new three-dimensional segmentation algorithms based on the three-dimensional topological map [18]. This work is quite advanced and we already obtained first results [1,16]. At last, we are also working on operations allowing to modify the topological map. Indeed by mixing the segmentation part of our work, the extraction of the map, and the different interactive or automatic operations, we could define a complete image analysis software based on the topological map that takes into account topological information of the analyzed objects.

Acknowledgments

The authors are grateful to reviewers for their useful comments and suggestions which greatly improved the paper. We also especially thank all Gral family for his careful reading and correction of this paper.

Appendix A. All the precodes algorithms

Algorithm 8 (Code corresponding to precode l_1).

Input: *last* and *up*, (x, y) coordinates of the new vertex.
Output: The “next” *last*.
 $a \leftarrow \beta_0(\textit{last}); b \leftarrow \beta_1(\textit{up})$
 0-unsew(*last*); 1-unsew(*up*)
 1-sew(*up*,*last*); 1-sew(*a*, *b*)
 Embedding of the vertex incident to *last* $\leftarrow (x, y)$
return *up*

Algorithm 9 (Code corresponding to precode l_2).

Input: *last* and *up*, (x, y) coordinates of the new vertex.
Output: The “next” *last*.
 $a \leftarrow$ a new dart; $b \leftarrow$ a new dart
 $c \leftarrow \beta_1(\textit{last});$ 1-unsew(*last*)
 1-sew(*last*,*a*); 1-sew(*a*, *b*)
 1-sew(*b*, *c*); 2-sew(*up*,*last*)
 Embedding of the vertex incident to *b* $\leftarrow (x, y)$
return *a*

Algorithm 10 (Code corresponding to precode l_3).

Input: *last* and *up*, (x, y) coordinates of the new vertex.
Output: The “next” *last*.
 $a \leftarrow$ a new dart; $b \leftarrow$ a new dart
 $c \leftarrow \beta_1(\textit{up});$ 1-unsew(*up*)
 1-sew(*up*,*a*); 1-sew(*a*, *b*)

1-sew(b, c); 2-sew($last, b$)
 Embedding of the vertex incident to $a \leftarrow (x, y)$
return up

Algorithm 11 (Code corresponding to precode l_4).

Input: $last$ and up , (x, y) coordinates of the new vertex.

Output: The “next” $last$.

$a \leftarrow$ a new dart; $b \leftarrow$ a new dart
 $c \leftarrow$ a new dart; $d \leftarrow$ a new dart
 1-sew(a, b); 1-sew(b, c)
 1-sew(c, d); 1-sew(d, a)
 2-sew($last, a$); 2-sew(up, b)
 Embedding of the vertex incident to $d \leftarrow (x, y)$
return c

Algorithm 12 (Code corresponding to precode f_1).

Input: $last$ and up , (x, y) coordinates of the new vertex.

Output: The “next” $last$.

$a \leftarrow \beta_0(last)$; $b \leftarrow \beta_1(last)$
 $c \leftarrow \beta_1(up)$; $d \leftarrow \beta_1(c)$
 1-unsew($last$); 2-unsew(a)
 0-unsew(c); 1-unsew(c)
 1-sew($last, c$); 1-sew(c, b)
 1-sew(up, d); 2-sew(up, a)
 Embedding of the vertex incident to $c \leftarrow (x, y)$
return $last$

Algorithm 13 (Code corresponding to precode f_2).

Input: $last$ and up , (x, y) coordinates of the new vertex.

Output: The “next” $last$.

$a \leftarrow \beta_0(last)$; $b \leftarrow \beta_1(last)$; $c \leftarrow \beta_1(up)$
 0-unsew($last$); 1-unsew($last$); 1-unsew(up)
 0-sew(b, a); 1-sew($up, last$); 1-sew($last, c$)
 Embedding of the vertex incident to $last \leftarrow (x, y)$
return up

Algorithm 14 (Code corresponding to precode t_1).

Input: $last$ and up , (x, y) coordinates of the new vertex.

Output: The “next” $last$.

$a \leftarrow$ a new dart; $b \leftarrow$ a new dart
 $c \leftarrow \beta_1(last)$; 2-sew(up, a)
 Add geometrical point incident to $last$ at the end of the edge embedding of a
 0-unsew($last$); 1-unsew($last$)
 1-sew($a, last$); 1-sew($last, b$)
 1-sew(b, a); 0-sew(c, up)

Embedding of the vertex incident to $b \leftarrow (x, y)$
return $last$

Algorithm 15 (Code corresponding to precod t_2).

Input: $last$ and up , (x, y) coordinates of the new vertex.

Output: The “next” $last$.

$a \leftarrow \beta_0(last)$; $b \leftarrow \beta_1(last)$; $c \leftarrow \beta_1(up)$

Add geometrical point incident to $last$ at the end of the edge embedding of a

0-unsew($last$); 1-unsew($last$); 1-unsew(up)

0-sew(b, a); 1-sew($up, last$); 0-sew($c, last$)

Embedding of the vertex incident to $last \leftarrow (x, y)$

return up

Algorithm 16 (Code corresponding to precod t_3).

Input: $last$ and up , (x, y) coordinates of the new vertex.

Output: The “next” $last$.

$a \leftarrow \beta_0(last)$; $b \leftarrow \beta_2(a)$; $c \leftarrow \beta_0(b)$

$d \leftarrow \beta_2(c)$; $t_1 \leftarrow \beta_1(b)$; $t_2 \leftarrow \beta_1(d)$

Add geometrical point incident to $last$ at the end of the edge embedding of a

Merge edge embedding of b at the end of edge embedding of a

0-unsew(b); 1-unsew(b); 2-unsew(b); delete b

0-unsew(d); 1-unsew(d); 2-unsew(d); delete d

0-unsew($last$); 1-sew($up, last$)

2-sew(a, c); 1-sew(a, t_2); 1-sew(c, t_1)

Embedding of the vertex incident to $last \leftarrow (x, y)$

return up

Algorithm 17 (Code corresponding to precod t_4).

Input: $last$ and up , (x, y) coordinates of the new vertex.

Output: The “next” $last$.

$a \leftarrow \beta_0(last)$; $b \leftarrow \beta_1(up)$; $c \leftarrow \beta_1(b)$

Add geometrical point incident to $last$ at the end of the edge embedding of a

0-unsew($last$); 0-unsew(b)

1-unsew(b); 2-unsew(b)

2-sew(up, a); 1-sew(up, c)

1-sew(a, b); 1-sew($b, last$)

Embedding of the vertex incident to $last \leftarrow (x, y)$

return b

References

- [1] E. Ahronovitz, C. Fiorio, S. Glaize, Topological operators on the topological graph of frontiers, in: Discrete Geometry for Computer Imagery, Lecture Notes in Computer Science, vol. 1568, Marne-la-Vallée, France, 1999, pp. 207–217.

- [2] Y. Bertrand, G. Damiand, C. Fiorio, Topological encoding of 3d segmented images, in: *Discrete Geometry for Computer Imagery, Lecture Notes in Computer Science*, vol. 1953, Uppsala, Sweden, 2000, pp. 311–324.
- [3] Y. Bertrand, G. Damiand, C. Fiorio, Topological map: minimal encoding of 3d segmented images, in: *Workshop on Graph Based Representations, Ischia, Italy*, 2001, pp. 64–73, IAPR-TC15.
- [4] Y. Bertrand, J.F. Dufourd, Algebraic specification of a 3d-modeler based on hypermaps, *Comput. Vision, Graph., Image Process.: Graph. Models Image Process.* 56 (1) (1994) 29–60.
- [5] Y. Bertrand, C. Fiorio, Y. Pennaneach, Border map: a topological representation for n d image analysis. In: *Discrete Geometry for Computer Imagery, Lecture Notes in Computer Science*, vol. 1568, Marne-la-Vallée, France, 1999, pp. 242–257.
- [6] P. Bourdon, O. Alata, G. Damiand, C. Olivier, Y. Bertrand, Geometrical and topological informations for image segmentation with Monte Carlo Markov chain implementation, in: *Vision Interface, Calgary, Canada*, May 2002, pp. 413–420.
- [7] J.-P. Braquelaire, L. Brun, Image segmentation with topological maps and inter-pixel representation, *J. Visual Commun. Image Represent.* 9 (1) (1998) 62–79.
- [8] J.-P. Braquelaire, P. Desbarats, J.-P. Domenger, 3d split and merge with 3-maps, in: *Workshop on Graph Based Representations, Ischia, Italy*, May 2001, pp. 32–43, IAPR-TC15.
- [9] J.-P. Braquelaire, P. Desbarats, J.-P. Domenger, C.A. Wüthrich, A topological structuring for aggregates of 3d discrete objects, in: *Workshop on Graph Based Representations, Austria*, May 1999, pp. 193–202, IAPR-TC15.
- [10] J.-P. Braquelaire, J.-P. Domenger, Representation of segmented images with discrete geometric maps, *Image Vision Comput.* 17 (10) (1999) 715–735.
- [11] L. Brun, J.-P. Domenger, A new split and merge algorithm with topological maps and inter-pixel boundaries, in: *The Fifth International Conference in Central Europe on Computer Graphics and Visualization*, February 1997.
- [12] L. Brun, J.-P. Domenger, J.-P. Braquelaire, Discrete maps: a framework for region segmentation algorithms, in: *Workshop on Graph Based Representations, Lyon*, April 1997, IAPR-TC15, published in *Advances in Computing*, Springer.
- [13] P. Charnier, Outils algorithmiques pour le codage interpixel et ses applications, Thèse de doctorat, Université Montpellier II, janvier, 1995.
- [14] R. Cori, Un code pour les graphes planaires et ses applications, PhD thesis, Université Paris VII, 1973.
- [15] R. Cori, Un code pour les graphes planaires et ses applications, in: *Astérisque*, volume 27, Soc. Math. de France, Paris, France, 1975.
- [16] G. Damiand, Définition et étude d'un modèle topologique minimal de représentation d'images 2D et 3D, Thèse de doctorat, Université Montpellier II, décembre, 2001.
- [17] G. Damiand, P. Lienhardt, Removal and contraction for n -dimensional generalized maps, in: *Computer Vision Winter Workshop, Bad Aussee, Austria*, February 2002, pp. 208–221.
- [18] G. Damiand, P. Resch, Topological map based algorithms for 3d image segmentation, in: *Discrete Geometry for Computer Imagery, Lecture Notes in Computer Science*, vol. 2301, Bordeaux, France, April 2002, pp. 220–231.
- [19] J.P. Domenger, Conception et implémentation du noyau graphique d'un environnement 2D1/2 d'édition d'images discrètes, Thèse de doctorat, Université Bordeaux I, avril 1992.
- [20] J. Edmonds, A combinatorial representation for polyhedral surfaces, *Notices Am. Math. Soc.* 7 (1960).
- [21] C. Fiorio, Approche interpixel en analyse d'images: une topologie et des algorithmes de segmentation, Thèse de doctorat, Université Montpellier II, novembre 1995.
- [22] C. Fiorio, A topologically consistent representation for image analysis: the frontiers topological graph, in: *Discrete Geometry for Computer Imagery, Lecture Notes in Computer Science*, vol. 1176, Lyon, France, November 1996, pp. 151–162.
- [23] J. Françon, Topologie de khalimski et kovalévski et algorithmique graphique, Rapport de recherche 91-10, Université Louis-Pasteur, Centre de Recherche en Informatique, Strasbourg, France, 1991.
- [24] A. Jacques, Constellations et graphes topologiques, in: *Combinatorial Theory and Applications*, vol. 2, 1970, pp. 657–673.

- [25] E. Khalinsky, R. Kopperman, P.R. Meyer, Boundaries in digital planes, *J. Appl. Math. Stochastic Anal.* 3 (1) (1990) 27–55.
- [26] T.Y. Kong, R. Kopperman, P.R. Meyer, A topological approach to digital topology, *Am. Math. Monthly* 98 (10) (1991) 901–917.
- [27] T.Y. Kong, A. Rosenfeld, Digital topology: introduction and survey, *Comput. Vision, Graph., Image Process.* 48 (3) (1989) 357–393.
- [28] V.A. Kovalevsky, Finite topology as applied to image analysis, *Comput. Vision, Graph., Image Process.* 46 (1989) 141–161.
- [29] W.G. Kropatsch, Building irregular pyramids by dual-graph contraction, *Vision, Image Signal Process.* 142 (6) (1995) 366–374.
- [30] W.G. Kropatsch, Abstraction pyramids on discrete representations, in: *Discrete Geometry for Computer Imagery*, vol. 2301 in LNCS, Bordeaux, France, April 2002, pp. 1–21.
- [31] W.G. Kropatsch, H. Macho, Finding the structure of connected components using dual irregular pyramids, in: *Discrete Geometry for Computer Imagery, Invited Lecture*, September 1995, pp. 147–158.
- [32] P. Lienhardt, Subdivision of n -dimensional spaces and n -dimensional generalized maps. in: *Proc. 5th Annual ACM Symp. on Computational Geometry*, Saarbrücken, Germany, 1989, pp. 228–236.
- [33] P. Lienhardt, Topological models for boundary representation: a comparison with n -dimensional generalized maps, *Comput. Aided Design* 23 (1) (1991).
- [34] P. Lienhardt, N -Dimensional generalized combinatorial maps and cellular quasi-manifolds, *Int. J. Comput. Geom. Appl.* 4 (3) (1994) 275–324.
- [35] J.-G. Paillancy, J.-M. Jolion, The frontier-region graph, in: *Workshop on Graph Based Representations, Computing Supplementum*, vol. 12, Springer, 1997, pp. 123–134.
- [36] A. Rosenfeld, Adjacency in digital pictures, *Informat. Control* 26 (1) (1974) 24–33.
- [37] W.T. Tutte, A census of planar maps, *Canad. J. Math.* 15 (1963) 249–271.

Annexe D

On some applications of
generalized functionality for
arithmetic discrete planes [[19](#)]

On some applications of generalized functionality for arithmetic discrete planes

Valérie Berthé, Christophe Fiorio, Damien Jamet ^{*}, Fabrice Philippe

LIRMM, UMR 5506, Univ. Montpellier II, 161 rue Ada, F-34392 Montpellier Cedex 05, France

Abstract

Naive discrete planes are well known to be functional on a coordinate plane. The aim of our paper is to extend the functionality concept to a larger family of arithmetic discrete planes, by introducing suitable projection directions $(\alpha_1, \alpha_2, \alpha_3)$ satisfying $\alpha_1 v_1 + \alpha_2 v_2 + \alpha_3 v_3 = w$. Several applications are considered. We first study certain local configurations, that is, the (m, n) -cubes introduced in Ref. [J. Vittone, J.-M. Chassery, (n, m) -cubes and Farey Nets for Naive Planes Understanding, in: DGCI, 8th International Conference, Lecture Notes in Computer Science, vol. 1568, Springer-Verlag, 1999, pp. 76–87.]. We compute their number for a given (m, n) and study their statistical behaviour. We then apply functionality to formulate an algorithm for generating arithmetic discrete planes, inspired by Debled-Rennesson [I. Debled-Rennesson, Reconnaissance des Droites et Plans Discrets, Thèse de doctorat, Université Louis Pasteur, Strasbourg, France, 1995.]. We also prove that an arithmetic discrete plane may be endowed with a combinatorial surface structure, in the spirit of Ref. [Y. Kenmochi, A. Imiyam Combinatorial topologies for discrete planes, in: DGCI, 11th International Conference, DGCI 2003, Lecture Notes in Computer Science, vol. 2886, Springer-Verlag, 2003, pp. 144–153.].

© 2006 Elsevier B.V. All rights reserved.

Keywords: Digital planes; Arithmetic planes; Local configurations; Functionality of discrete planes

1. Introduction

Let $\mathbf{v} = (v_1, v_2, v_3) \in \mathbb{Z}^3$ and $\mu, w \in \mathbb{Z}^2$. The arithmetic discrete plane $\mathfrak{P}(\mathbf{v}, \mu, w)$ is the set of all points $\mathbf{x} = (x_1, x_2, x_3) \in \mathbb{Z}^3$ satisfying

$$0 \leq v_1 x_1 + v_2 x_2 + v_3 x_3 + \mu < w.$$

Arithmetic discrete planes with a common normal vector $\mathbf{v}$ are mainly characterized by their thickness w . For example, in such a class the *naive* planes ($w = \max(|v_1|, |v_2|, |v_3|)$) are the thinnest 2-separating ones, while the standard planes ($w = |v_1| + |v_2| + |v_3|$) are the thinnest 0-separating ones (see [1]).

Arithmetic discrete planes play a key role in polyedrisation of discrete objects [15, 8, 12]. Indeed tridimensional discretized objects are usually described as pieces, generally

triangular, of arithmetic discrete planes. A reasonable polyedrisation mainly requires that the involved objects have *good* topological properties, such as connectedness and/or absence of κ -tunnels, for $\kappa \in \{0, 1, 2\}$ [1] (intuitively, a κ -tunnel occurs if two voxels, one above the plane and the other below, have a k -dimensional intersection). If naive arithmetic planes are rather good candidates for this approach as being the thinnest ones without 2-tunnels, one major withdraw is that the intersection of two naive arithmetic discrete planes is usually not an arithmetic discrete line. Similar problems occur when considering standard arithmetic discrete planes. V. Brimkov and R. Barneva introduced a new class of arithmetic discrete planes [9, 11], the so-called *graceful planes*, defined as $\mathfrak{P}(\mathbf{v}, \mu, w)$ with $0 \leq v_1 \leq v_2 \leq v_3$ and $w = \max\{v_3, v_1 + v_2\}$, which allow for constructing thin discrete triangular tunnel-free meshes admitting analytical description. In order to take into account noise during digitization, Debled-Rennesson et al. initiated a new approach of segmentation of discrete curves with arithmetic discrete lines of various thicknesses

^{*} Corresponding author. Tel.: +33467418682.

E-mail addresses: berthe@lirmm.fr (V. Berthé), fiorio@lirmm.fr (C. Fiorio), jamet@lirmm.fr (D. Jamet), philippe@lirmm.fr (F. Philippe).

[14]. The latter works indicate that a better understanding of the topological, arithmetic and geometric structure of arithmetic discrete linear objects of *any thickness* would be very useful.

Naive planes have been widely studied (see for instance [27,16,15, 1, 31, 3,10,11, 13]) and are well known to be *functional*, that is, in a one-to-one correspondence with the integer points of one of the coordinate planes by an orthogonal projection map. In other words, given a naive arithmetic discrete plane $\mathfrak{P}$ and the suitable coordinate plane, for any integer point P of this coordinate plane there exists a unique point of $\mathfrak{P}$ obtained by adding a third coordinate to P .

The aim of the present paper is to extend the notion of functionality for naive arithmetic discrete planes to a larger family of arithmetic discrete planes. For that purpose, instead of projecting on a coordinate plane, we introduce a suitable orthogonal projection map on a plane along a direction $\alpha = (\alpha_1, \alpha_2, \alpha_3) \in \mathbb{Z}^3$, in some sense dual to the normal vector of the discrete plane $\mathfrak{P}(\mathbf{v}, \mu, w)$, that is, $\alpha_1 v_1 + \alpha_2 v_2 + \alpha_3 v_3 = w$, so that the projection of $\mathbb{Z}^3$ and the points of the discrete plane $\mathfrak{P}(\mathbf{v}, \mu, w)$ are in one-to-one correspondence. Functionality allows us to reduce a three-dimensional problem to a two-dimensional one, thus leads to a better understanding of the combinatorial and geometric properties of arithmetic discrete planes.

The present paper is organized as follows. We first recall some basic notions on arithmetic discrete planes, while extending their definition to the case of *real* $(\mathbf{v}, \mu, w)$ -parameters (we call *rational* the classic case).

Generalized functionality is introduced in Section 3, with the main result:

Theorem 1. *Let $\mathfrak{P}(\mathbf{v}, \mu, w)$ be an arithmetic discrete plane, and let $\alpha \in \mathbb{Z}^3$ be such that $\gcd\{\alpha_1, \alpha_2, \alpha_3\} = 1$. Let $\pi_\alpha^\perp : \mathbb{R}^3 \rightarrow \{\mathbf{x} \in \mathbb{R}^3, \langle \alpha, \mathbf{x} \rangle = 0\}$ be the affine orthogonal projection map onto the plane $\mathbf{x} \in \mathbb{R}^3, \langle \alpha, \mathbf{x} \rangle = 0$ along the vector α . Then the map $\pi_\alpha^\perp : \mathfrak{P}(\mathbf{v}, \mu, w) \rightarrow \pi_\alpha^\perp(\mathbb{Z}^3)$ is a bijection if, and only if, $|\langle \alpha, \mathbf{v} \rangle| = w$.*

Up to the sign, such α -vectors are called *functional* for the plane. We show that any rational arithmetic discrete plane admits functional vectors.

In Section 4, we study the functional lattice Γ_α obtained by projecting an arithmetic discrete plane on one of the coordinate planes along a functional vector α , and we compute $\mathbb{Z}$ -basis of such lattices. We exhibit the converse function $\pi_\alpha^{-1} : \Gamma_\alpha \rightarrow \mathfrak{P}(\mathbf{v}, \mu, w)$ of the projection map π_α . This function admits a very simple expression when $\alpha_3 = 1$.

Several applications of generalized functionality are given in the rest of the paper, under the assumption that a *functional vector exists with a coordinate equal to 1*. We first apply in Section 5 the generalized functionality property to the enumeration of some local configurations, in particular we generalize results on (m, n) -cubes.

Section 6 provides one with two further applications of the generalized functionality property. We first extend Isabelle Debled-Rennesson's algorithm for computing the geometric representation of naive arithmetic discrete planes

[15] to any rational arithmetic discrete plane $\mathfrak{P}(\mathbf{v}, \mu, w)$ with $w - v_3 \in \gcd(v_1, v_2)\mathbb{Z}$. We then show that one can provide any arithmetic discrete plane $\mathfrak{P}(\mathbf{v}, \mu, w)$ with a structure of two-dimensional combinatorial manifold, under the assumption $w \in v_1\mathbb{Z} + v_2\mathbb{Z} + v_3$ and $w \geq \max\{|v_1|, |v_2|\}$. Section 7 concludes the paper.

For clarity issues, we have chosen to work here in a three-dimensional space but all the results and methods presented extend in a natural way to $\mathbb{R}^n$, with $n \geq 2$, as well as to arithmetic discrete lines. Let us note that we have tried to make this paper essentially self-contained. This paper is an extended version of [BFJ05], but also contains some new applications of functionality.

2. Arithmetic discrete planes

In the present paper, the $\mathbb{R}$ -vector space $\mathbb{R}^3$ is endowed with its canonical basis $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$. Let $\mathbf{v}$ and $\mathbf{v}'$ be two vectors of $\mathbb{R}^3$, their usual scalar product in $\mathbb{R}^3$ is denoted by $\langle \mathbf{v}, \mathbf{v}' \rangle$. The subsets of all nonzero elements in $\mathbb{R}$ and $\mathbb{N}$ are denoted by $\mathbb{R}^\star$ and $\mathbb{N}^\star$, respectively.

Let $\mathbf{v} \in \mathbb{R}^3$, $\mu \in \mathbb{R}$, $w \in \mathbb{R}_+^\star$, and define

$$\mathfrak{P}(\mathbf{v}, \mu, w) = \{\mathbf{x} \in \mathbb{Z}^3, 0 \leq \langle \mathbf{v}, \mathbf{x} \rangle + \mu < w\}. \quad (1)$$

An *arithmetic discrete plane* is a subset $\mathcal{P}$ of $\mathbb{Z}^3$ such that

$$\exists (\mathbf{v}, \mu, w) \in \mathbb{R}^3 \times \mathbb{R} \times \mathbb{R}_+^\star, \quad \mathcal{P} = \mathfrak{P}(\mathbf{v}, \mu, w). \quad (2)$$

The following lemma provides us with basic properties of arithmetic discrete planes. In particular, the vector $\mathbf{v}$ in Eq. (2) is shown to be unique up to scale. For $\mathbf{v} = (v_1, v_2, v_3) \in \mathbb{Z}^3$, let $\gcd\{v_1, v_2, v_3\}$ be denoted by $\gcd(\mathbf{v})$, and for $\mathbf{v} \in \mathbb{R}^3$, let the dimension of the $\mathbb{Q}$ -vector space spanned by $\{v_1, v_2, v_3\}$ be denoted by $\dim_{\mathbb{Q}} \mathbf{v}$ (for instance, the following $\mathbf{v}$ -values: $(1, 2, 3)$, $(\sqrt{2}, 2\sqrt{2}, 3\sqrt{2})$, $(1, \sqrt{2}, 2\sqrt{2})$, and $(1, \sqrt{2}, \sqrt{3})$ yield $\dim_{\mathbb{Q}} \mathbf{v} = 1, 1, 2$, and 3, respectively).

Lemma 2. *Let $\mathcal{P} = \mathfrak{P}(\mathbf{v}, \mu, w)$ be an arithmetic discrete plane with $(\mathbf{v}, \mu, w) \in \mathbb{R}^3 \times \mathbb{R} \times \mathbb{R}_+^\star$.*

- (i) *If $\mathcal{P} = \mathfrak{P}(\mathbf{v}', \mu', w')$ then there exists $t \in \mathbb{R}$ such that $\mathbf{v}' = t\mathbf{v}$.*
- (ii) *The set $\{\langle \mathbf{v}, \mathbf{x} \rangle + \mu, \mathbf{x} \in \mathcal{P}\}$ is dense in $[0, w[$ if, and only if, $\dim_{\mathbb{Q}} \mathbf{v} > 1$.*
- (iii) *There exist $\mathbf{v}' \in \mathbb{Z}^3$ with $\gcd(\mathbf{v}') = 1$, $\mu' \in \mathbb{Z}$, and $w' \in \mathbb{N}^\star$ such that $\mathcal{P} = \mathfrak{P}(\mathbf{v}', \mu', w')$ if, and only if, $\dim_{\mathbb{Q}} \mathbf{v} = 1$. If so, then $\text{Card}\{\langle \mathbf{v}, \mathbf{x} \rangle + \mu, \mathbf{x} \in \mathcal{P}\} = w'$.*

Proof. Let us assume $\dim_{\mathbb{Q}} \mathbf{v} > 1$. There exist $i, j \in \{1, 2, 3\}$ such that $v_i \neq 0$ and $\frac{v_j}{v_i} \notin \mathbb{Q}$. By Kronecker's Approximation Theorem, the set $\{at + b, (a, b) \in \mathbb{Z}^2\}$ is dense in $\mathbb{R}$ if $t \in \mathbb{R} \setminus \mathbb{Q}$, that is, for every interval $I \subset \mathbb{R}$, there exists $(a, b) \in \mathbb{Z}^2$ such that $at + b \in I$. Let I be a sub-interval of $[0, w[$; one has $\frac{v_j}{v_i} \notin \mathbb{Q}$; hence there exists $(x_i, x_j) \in \mathbb{Z}^2$ such that $x_i + x_j \frac{v_j}{v_i} + \frac{\mu}{v_i} \in I$, that is, $x_i v_i + x_j v_j + \mu \in I$. Hence the set $\{\langle \mathbf{v}, \mathbf{x} \rangle + \mu, \mathbf{x} \in \mathcal{P}\}$ is dense in $[0, w[$. If $\dim_{\mathbb{Q}} \mathbf{v} = 1$, then there exists $t \in \mathbb{R}_+^\star$ such that $t\mathbf{v} \in \mathbb{Z}^3$ and $\gcd(t\mathbf{v}) = 1$. A straightfor-

ward calculation shows that in (iii), one can take $\mathbf{v}' = t\mathbf{v}, \mu' = \lfloor t\mu \rfloor$, and $w' = \lceil tw - t\mu \rceil + \lfloor t\mu \rfloor$. Furthermore note that the set $\{\mathbf{x} \in \mathbb{Z}^3, \langle \mathbf{v}', \mathbf{x} \rangle + \mu' = k\}$ is not empty if, and only if, $k \in \mathbb{Z}$. Thus $\text{Card}\{\langle \mathbf{v}', \mathbf{x} \rangle + \mu', \mathbf{x} \in \mathcal{P}\} = w'$, which completes the proof of (ii) and (iii) since the sets $\langle \mathbf{v}', \mathbf{x} \rangle + \mu', \mathbf{x} \in \mathcal{P}$ and $\langle \mathbf{v}, \mathbf{x} \rangle + \mu, \mathbf{x} \in \mathcal{P}$ are homothetical.

Let us next prove (i). First assume that $\mathcal{P} = \mathfrak{P}(\mathbf{v}, \mu, w) = \mathfrak{P}(\mathbf{v}', \mu', w')$ with the further condition $0 \leq \mu < w$, so that $\mathfrak{P}$ contains $(0,0,0)$. Let V be the $\mathbb{R}$ -vector space spanned by $\{\mathbf{v}, \mathbf{v}'\}$, and define

$$B = \{\mathbf{x} \in \mathbb{R}^3, \quad 0 \leq \langle \mathbf{v}, \mathbf{x} \rangle + \mu < w\}, \quad B' = \{\mathbf{x} \in \mathbb{R}^3, \quad 0 \leq \langle \mathbf{v}', \mathbf{x} \rangle + \mu' < w'\}.$$

Suppose $\dim_{\mathbb{R}} V = 2$. Then $B \cap B' \cap V$ is bounded (a parallelogram), so that its discrete subset $\mathcal{P} \cap V$ is finite. Let $\mathbf{x} = a\mathbf{v} + b\mathbf{v}'$ with $(a, b) \in \mathbb{Z}^2$; then $\mathbf{x} \in \mathcal{P}$ if, and only if, $0 \leq \langle \mathbf{v}, \mathbf{x} \rangle + \mu < w$ and $0 \leq \langle \mathbf{v}', \mathbf{x} \rangle + \mu' < w'$. Therefore, $\mathcal{P} \cap V$ is the arithmetic discrete line with normal vector $(\langle \mathbf{v}, \mathbf{v} \rangle, \langle \mathbf{v}', \mathbf{v}' \rangle)$, translation parameter μ and thickness w' . Since it is not empty, it is not finite, a contradiction. Finally, let us show that the conclusion still holds for arbitrary μ . If $\dim_{\mathbb{Q}} \mathbf{v} = 1$, we may assume that $\mathbf{v} \in \mathbb{Z}^3$ with $\gcd(\mathbf{v}) = 1$ and $\mu \in \mathbb{Z}$. By using Bezout's Lemma let us choose $\mathbf{u}$ in $\mathbb{Z}^3$ such that $\langle \mathbf{u}, \mathbf{v} \rangle = 1$ and let $\mathbf{u}' = \mu\mathbf{u}$. If $\dim_{\mathbb{Q}} \mathbf{v} > 1$, choose $\mathbf{u}' \in \mathbb{Z}^3$ such that $\langle \mathbf{u}', \mathbf{v} \rangle \in [\mu, \mu + w']$. Let $\tau : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ be the translation by $-\mathbf{u}'$, we have $\tau(\mathcal{P}) = \mathfrak{P}(\mathbf{v}, \mu - \langle \mathbf{u}', \mathbf{v} \rangle, w) = \mathfrak{P}(\mathbf{v}', \mu - \langle \mathbf{u}', \mathbf{v} \rangle, w')$. Since $0 \leq \mu - \langle \mathbf{u}', \mathbf{v} \rangle < w'$, we still conclude that $\mathbf{v}$ and $\mathbf{v}'$ are colinear. $\square$

An arithmetic discrete plane that may be represented by $\mathcal{P} = \mathfrak{P}(\mathbf{v}, \mu, w)$ with $\dim_{\mathbb{Q}} \mathbf{v} = 1$ is called *rational*, otherwise it is called *irrational*. According to Lemma 2, the distinction is exclusive. From now on, we shall agree that any representation $\mathfrak{P}(\mathbf{v}, \mu, w)$ of a rational arithmetic discrete plane satisfies:

$$\mathbf{v} \in \mathbb{Z}^3 \text{ and } \gcd(\mathbf{v}) = 1, \quad \mu \in \mathbb{Z}, \quad w \in \mathbb{N}^*. \quad (3)$$

Exactly two such representations exist for a rational arithmetic discrete plane

$$\mathfrak{P}(\mathbf{v}, \mu, w) = \mathfrak{P}(-\mathbf{v}, w - 1 - \mu, w). \quad (4)$$

Indeed, only two choices are allowed for $\mathbf{v}$ by Lemma 2 (i), and w is unique by Lemma 2 (iii). Then other μ -values are easily ruled out. For an irrational arithmetic discrete plane, constraint $\langle \mathbf{v}, \mathbf{v} \rangle = 1$ and Lemma 2 (ii) result in similar conclusions: There are at most two representations of an irrational arithmetic discrete plane $\mathfrak{P}(\mathbf{v}, \mu, w)$, the second one being $\mathfrak{P}(-\mathbf{v}, w - \mu, w)$ and happening if, and only if, neither μ nor $w - \mu$ belong to the $\mathbb{Q}$ -vector space spanned by $\{v_1, v_2, v_3\}$.

Some among rational arithmetic discrete planes are usually given particular names [27,16,15,17]. Let $\mathbf{v} = (v_1, v_2, v_3) \in \mathbb{Z}^3$; if $w = \max\{|v_1|, |v_2|, |v_3|\}$ (resp. $w = |v_1| + |v_2| + |v_3|$), then $\mathcal{P} = \mathfrak{P}(\mathbf{v}, \mu, w)$ is called *naive* (resp. *standard*). It is important to remark that, according to the above discussion, these definitions do not depend on any specific representation of $\mathcal{P}$.

Let us next recall a classical property of naive discrete planes with normal vector $\mathbf{v}$. For a given $\alpha \in \mathbb{Z}^3$, let $\pi_{\alpha}^{\perp} : \mathbb{R}^3 \rightarrow \{\mathbf{x} \in \mathbb{R}^3 : \langle \alpha, \mathbf{x} \rangle = 0\}$ stand for the orthogonal projection map onto the plane $\langle \alpha, \mathbf{x} \rangle = 0$. We still denote by $\pi_{\alpha}^{\perp}$ its restriction to any subset of $\mathbb{R}^3$ or its corestriction to any subset of the plane $\langle \alpha, \mathbf{x} \rangle = 0$, as for instance $\pi_{\alpha}^{\perp} : \mathfrak{P}(\mathbf{v}, \mu, w) \rightarrow \pi_{\alpha}^{\perp}(\mathbb{Z}^3)$.

Theorem 3 [27]. *Let $\mathcal{P} = \mathfrak{P}(\mathbf{v}, \mu, w)$ be a naive arithmetic discrete plane, with $\mathbf{v} = (v_1, v_2, v_3) \in \mathbb{R}^3$. If $|v_i| = w$, for $i = 1, 2$ or 3 , then $\mathcal{P}$ is in bijection with the integer points of the plane $\langle \mathbf{e}_i, \mathbf{x} \rangle = 0$ by the projection map $\pi_{\mathbf{e}_i}^{\perp}$.*

The plane $\langle \mathbf{e}_i, \mathbf{x} \rangle = 0$ in Theorem 3 is called a *functional plane* of $\mathcal{P}$. An analogous result holds for standard discrete planes:

Theorem 4 ([17,5]). *Let $\mathcal{P}$ be a standard discrete plane and let $\alpha = \mathbf{e}_1 + \mathbf{e}_2 + \mathbf{e}_3$. Then the map $\pi_{\alpha}^{\perp} : \mathcal{P} \rightarrow \pi_{\alpha}^{\perp}(\mathbb{Z}^3)$ is a bijection.*

Both results are extended in several directions in the following section.

3. Generalized functionality

In both cases investigated in Theorems 3 and 4 the following property holds: there exists a vector $\alpha \in \mathbb{Z}^3$ such that the projection map $\pi_{\alpha}^{\perp} : \mathfrak{P}(\mathbf{v}, \mu, w) \rightarrow \pi_{\alpha}^{\perp}(\mathbb{Z}^3)$ is a bijection, and $\langle \alpha, \mathbf{v} \rangle = w$. In this section, it is extended to a large class of arithmetic discrete planes by means of *functional* directions:

Definition 5. A vector $\alpha \in \mathbb{Z}^3$ is called *functional* for an arithmetic discrete plane $\mathfrak{P}(\mathbf{v}, \mu, w)$ if it satisfies conditions $\gcd(\alpha) = 1$ and $\langle \alpha, \mathbf{v} \rangle = w$.

According to Lemma 2, functionality of a vector $\alpha \in \mathbb{Z}^3$ for an arithmetic discrete plane $\mathfrak{P}(\mathbf{v}, \mu, w)$ does not depend on the representation of the latter.

Furthermore, we improve Theorems 3 and 4 by showing that the vectors $\alpha \in \mathbb{Z}^3$ providing us with bijective projections $\pi_{\alpha}^{\perp}$ are exactly the functional ones (to be precise, this statement also holds for $-\alpha$ if α is functional since $\gcd(\alpha) = \gcd(-\alpha)$ and $\pi_{\alpha}^{\perp} = \pi_{-\alpha}^{\perp}$).

Theorem 6. *Let $\mathcal{P} = \mathfrak{P}(\mathbf{v}, \mu, w)$ be an arithmetic discrete plane, and let $\alpha \in \mathbb{Z}^3$ be such that $\gcd(\alpha) = 1$. Then the map $\pi_{\alpha}^{\perp} : \mathcal{P} \rightarrow \pi_{\alpha}^{\perp}(\mathbb{Z}^3)$ is a bijection if, and only if, $|\langle \alpha, \mathbf{v} \rangle| = w$.*

Proof. Throughout the proof, we assume w.l.o.g. that $\langle \alpha, \mathbf{v} \rangle \geq 0$. Let $\mathbf{x}, \mathbf{x}' \in \mathbb{Z}^3$, then $\pi_{\alpha}^{\perp}(\mathbf{x}) = \pi_{\alpha}^{\perp}(\mathbf{x}')$ if, and only if, there exists $(k, k') \in \mathbb{Z}^2$ such that $k'(\mathbf{x}' - \mathbf{x}) = k\alpha$. Since $\gcd(\alpha) = 1$, k' divides k . In other words, $\pi_{\alpha}^{\perp}(\mathbf{x}) = \pi_{\alpha}^{\perp}(\mathbf{x}')$ if, and only if, there exists $c \in \mathbb{Z}$ such that $\mathbf{x}' = \mathbf{x} + c\alpha$. Moreover, $\mathbf{x} + c\alpha \in \mathcal{P}$ if, and only if, $\frac{-\langle \mathbf{x}, \mathbf{v} \rangle + \mu}{\langle \alpha, \mathbf{v} \rangle} \leq c < \frac{w - \langle \mathbf{x}, \mathbf{v} \rangle + \mu}{\langle \alpha, \mathbf{v} \rangle}$. Accordingly, if $|\langle \alpha, \mathbf{v} \rangle| = w$ then c is allowed to take exactly one value, so that $\pi_{\alpha}^{\perp} : \mathcal{P} \rightarrow \pi_{\alpha}^{\perp}(\mathbb{Z}^3)$ is a bijection.

Conversely assume that $\pi_\alpha^\perp : \mathcal{P} \rightarrow \pi_\alpha^\perp(\mathbb{Z}^3)$ is a bijection. If $\mathcal{P}$ is rational, take $\mathbf{x} \in \mathbb{Z}^3$ such that $\langle \mathbf{x}, \mathbf{v} \rangle + \mu = 0$ by using Bezout's Lemma. On the one hand, $\mathbf{x} \in \mathcal{P}$ and $\langle \mathbf{x} + \alpha, \mathbf{v} \rangle + \mu = \langle \alpha, \mathbf{v} \rangle \geq 0$. Moreover, $\pi_\alpha^\perp(\mathbf{x} + \alpha) = \pi_\alpha^\perp(\mathbf{x})$. Since $\pi_\alpha^\perp$ is one-to-one, $\mathbf{x} + \alpha \notin \mathcal{P}$, thus $\langle \alpha, \mathbf{v} \rangle \geq w$. On the other hand, suppose that $\langle \alpha, \mathbf{v} \rangle > w$ and choose $\mathbf{y} \in \mathbb{Z}^3$ such that $\langle \mathbf{y}, \mathbf{v} \rangle + \mu = -1$. Then, for $c \in \mathbb{Z}$, $\langle \mathbf{y} + c\alpha, \mathbf{v} \rangle + \mu = c\langle \alpha, \mathbf{v} \rangle - 1$, which is negative if $c \leq 0$, and larger than or equal to $w - 1$ if $c > 0$. Since $\pi_\alpha^\perp$ is onto, this cannot happen. Thus $\langle \alpha, \mathbf{v} \rangle = w$.

If $\mathcal{P}$ is irrational, we parallel the previous proof by using a density argument instead of Bezout's Lemma. Consider any $\varepsilon \in \mathbb{R}$ and choose $\mathbf{x} \in \mathbb{Z}^3$ such that $\langle \mathbf{x}, \mathbf{v} \rangle + \mu \in [0, \varepsilon]$. We obtain as above $\langle \alpha, \mathbf{v} \rangle > w - \varepsilon$, thus $\langle \alpha, \mathbf{v} \rangle \geq w$. Suppose $\langle \alpha, \mathbf{v} \rangle > w$, let $v = \langle \alpha, \mathbf{v} \rangle - w$, and choose $\mathbf{y} \in \mathbb{Z}^3$ such that $\langle \mathbf{y}, \mathbf{v} \rangle + \mu \in]-v/2, 0]$. Then, for $c \in \mathbb{Z}$, $\langle \mathbf{y} + c\alpha, \mathbf{v} \rangle + \mu = c\langle \alpha, \mathbf{v} \rangle + \langle \mathbf{y}, \mathbf{v} \rangle + \mu$. This quantity is negative if $c \leq 0$, and is larger w , if $c > 0$. $\square$

If α is a functional vector for an arithmetic discrete plane $\mathcal{P}$, then Theorem 6 states that, looking at $\mathcal{P}$ along this direction, one can see all its points as if they were on the plane $\{\mathbf{x} \in \mathbb{R}^3, \langle \alpha, \mathbf{x} \rangle = 0\}$. We show later that a natural regular lattice structure emerges from this point of view. Let us first complete Theorem 6 by relaxing condition $\gcd(\alpha) = 1$ for rational planes.

Corollary 7. Let $\mathcal{P} = \mathfrak{P}(\mathbf{v}, \mu, w)$ be a rational arithmetic discrete plane and $\langle \alpha, \mathbf{v} \rangle = w$. If $\gcd(\alpha) = d$, where $d \in \mathbb{N}$ divides w , then $\pi_\alpha^\perp : \mathcal{P} \rightarrow \pi_\alpha^\perp(\mathbb{Z}^3)$ satisfies, for each $\mathbf{p} \in \pi_\alpha^\perp(\mathbb{Z}^3)$,

$$(\pi_\alpha^\perp)^{-1}(\{\mathbf{p}\}) = \left\{ \mathbf{x}_\mathbf{p} + k \frac{\alpha}{d}, 0 \leq k \leq d-1 \right\}, \quad (5)$$

where $\mathbf{x}_\mathbf{p}$ is the unique solution in $\mathfrak{P}(\mathbf{v}, \mu, w/d)$ of $\pi_\alpha^\perp(\mathbf{x}) = \mathbf{p}$.

Proof. Let $\mathcal{P}_k = \mathfrak{P}(\mathbf{v}, \mu - kw/d, w/d)$ and let us write $\mathfrak{P}$ as the disjoint union $\mathcal{P} = \bigcup_{k=0}^{d-1} \mathcal{P}_k$. One has $\pi_\alpha^\perp|_{\mathcal{P}_k} = \pi_\alpha^\perp$. By Theorem 6, the projection $\pi_\alpha^\perp : \mathcal{P}_k \rightarrow \pi_\alpha^\perp(\mathbb{Z}^3)$ is one-to-one for each k between 0 and $d-1$, so that the cardinality of $(\pi_\alpha^\perp)^{-1}(\{\mathbf{p}\})$ does not exceed d . Moreover, given $\mathbf{p} \in \pi_\alpha^\perp(\mathbb{Z}^3)$, $\mathbf{x}_\mathbf{p}$ is well defined. Since $\langle \alpha/d, \mathbf{v} \rangle = w/d$, for $0 \leq k \leq d-1$, each point $\mathbf{x}_\mathbf{p} + k \frac{\alpha}{d}$ is in $\mathcal{P}_k$. Whence the result. $\square$

In particular, taking $d = w$ in Corollary 7 one obtains:

Corollary 8. Let $\mathcal{P} = \mathfrak{P}(\mathbf{v}, \mu, w)$ be a rational arithmetic discrete plane and $\langle \alpha, \mathbf{v} \rangle = w$. If $\gcd(\alpha) = w$, then $\mathfrak{P}$ is the disjoint union $\mathcal{P} = \bigcup_{k=0}^{w-1} (\mathcal{P}_0 + k \frac{\alpha}{w})$, where $\mathcal{P}_0$ is the discrete plane $\{\mathbf{x} \in \mathbb{Z}^3, \langle \mathbf{v}, \mathbf{x} \rangle + \mu = 0\}$.

In case of an irrational discrete plane $\mathfrak{P}(\mathbf{v}, \mu, w)$, there is no reason for a vector $\alpha \in \mathbb{Z}^3$ to exist satisfying $\langle \alpha, \mathbf{v} \rangle = w$ – consider $\{v_1, v_2, v_3, w\}$ spanning a four-dimensional subspace of the $\mathbb{Q}$ -vector space $\mathbb{R}$. Nevertheless, if $\mathfrak{P}(\mathbf{v}, \mu, w)$ is rational then we deduce from Bezout's Lemma that such an α exists. We next prove that it can be chosen such that $\gcd(\alpha) = 1$.

Proposition 9. Any rational arithmetic discrete plane $\mathfrak{P}(\mathbf{v}, \mu, w)$ has a functional vector.

Proof. Let $\beta, \mathbf{u} \in \mathbb{Z}^3$ such that $\langle \beta, \mathbf{v} \rangle = 1$ and $\langle \mathbf{u}, \mathbf{v} \rangle = 0$. Let $d = \gcd(\mathbf{u})$ and let $\alpha = w\beta + \mathbf{u}/d$. An easy computation gives $\langle \alpha, \mathbf{v} \rangle = w$ and $\gcd(\alpha) = 1$. $\square$

Proposition 9 is illustrated in Fig. 1, in the case of a discrete line for a better visualisation. Moreover, rather than projecting on the line $\{\mathbf{x} \in \mathbb{R}^2, \langle \alpha, \mathbf{x} \rangle = 0\}$ we prefer projecting on the line $\{\mathbf{x} \in \mathbb{R}^2, \langle \mathbf{e}_2, \mathbf{x} \rangle = 0\}$, which amounts to the same as far as bijectivity is concerned. With Fig. 1 again, note that the projection is not one-to-one for $\alpha = \mathbf{e}_2$ since $\langle \mathbf{e}_2, \mathbf{v} \rangle = 2 < w$, and it is not onto for $\alpha = 2\mathbf{e}_1 + \mathbf{e}_1$ since $\langle \alpha, \mathbf{v} \rangle = 4 > w$ (see the proof of Th. 6).

4. Functional lattices

Let us next study how arithmetic discrete planes can be recoded in a functional way by a regular two-dimensional lattice, despite their three-dimensional structure. Let $\mathcal{P} = \mathfrak{P}(\mathbf{v}, \mu, w)$ be an arithmetic discrete plane and let $\alpha \in \mathbb{Z}^3$ be functional for $\mathcal{P}$, that is, $\gcd(\alpha) = 1$ and $\langle \alpha, \mathbf{v} \rangle = w$ (in case $\mathfrak{P}$ is rational, the existence of such a vector α is ensured by Proposition 9). Since one of the coordinates α_i , for $i \in \{1, 2, 3\}$, is non-zero, then we assume $\alpha_3 \neq 0$ without loss of generality.

Accordingly, we shall from now on prefer to $\pi_\alpha^\perp$ the projection π_α along α onto the plane $\{\mathbf{x} \in \mathbb{R}^3, \langle \mathbf{e}_3, \mathbf{x} \rangle = 0\}$, as it was done in Fig. 1. Indeed Theorem 6 also holds for π_α since π_α is a bijection if, and only if, $\pi_\alpha^\perp$ is also a bijection.

First note that, since $\pi_\alpha(\alpha) = 0$ and $\pi_\alpha(\mathbf{e}_i) = \mathbf{e}_i$ for $i \in \{1, 2\}$, one has $\alpha_3 \pi_\alpha(\mathbf{e}_3) = -\alpha_1 \mathbf{e}_1 - \alpha_2 \mathbf{e}_2$. Then, for all $\mathbf{x} = (x_1, x_2, x_3) \in \mathbb{Z}^3$, a straightforward calculation gives:

$$\pi_\alpha(\mathbf{x}) = \frac{\alpha_3 x_1 - \alpha_1 x_3}{\gcd\{\alpha_1, \alpha_3\}} \mathbf{f}_1 + \frac{\alpha_3 x_2 - \alpha_2 x_3}{\gcd\{\alpha_2, \alpha_3\}} \mathbf{f}_2, \quad \text{where} \quad \mathbf{f}_i = \frac{\gcd\{\alpha_i, \alpha_3\}}{\alpha_3} \mathbf{e}_i. \quad (6)$$

We deduce from Eq. (6) that $\Gamma_\alpha = \pi_\alpha(\mathbb{Z}^3) = \pi_\alpha(\mathcal{P})$ is a subset of $\mathbb{Z}\mathbf{f}_1 + \mathbb{Z}\mathbf{f}_2$. The description of Γ_α is made more precise below. In particular, it is shown to be a lattice, which shall be called a *functional lattice* of $\mathcal{P}$. This definition generalizes the concept of functionality defined for naive arithmetic discrete planes as a projection onto the integer points of one of the coordinate planes.

Let us first give a characterization of the points in Γ_α .

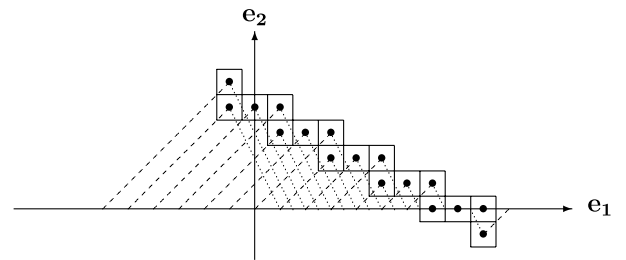


Fig. 1. Generalized functionality: the discrete line $0 \leq x_1 + 2x_2 - 7 < 3$ is projected on $\{\mathbf{x} \in \mathbb{R}^2, \langle \mathbf{e}_2, \mathbf{x} \rangle = 0\}$ along $\alpha = \mathbf{e}_1 + \mathbf{e}_2$ and $\alpha = -\mathbf{e}_1 + 2\mathbf{e}_2$.

Theorem 10. Assume that $\gcd\{\alpha\} = 1$, $\alpha_3 \neq 0$, and, for $i \in \{1, 2, 3\}$, let

$$\delta_i = \frac{\alpha_i}{\prod_{j \neq i} \gcd\{\alpha_i, \alpha_j\}} \in \mathbb{Z}.$$

A point with coordinates $(y_1, y_2) \in \mathbb{Z}^2$ in the basis $\{\mathbf{f}_1, \mathbf{f}_2\}$ belongs to Γ_α if, and only if, δ_3 divides $\delta_2 y_1 - \delta_1 y_2$.

Proof . According to Eq. (6), we have to show that the Diophantine system

$$\begin{cases} \gcd\{\alpha_1, \alpha_3\} y_1 = \alpha_3 x_1 - \alpha_1 x_3 \\ \gcd\{\alpha_2, \alpha_3\} y_2 = \alpha_3 x_2 - \alpha_2 x_3 \end{cases} \quad (7)$$

has a solution $(x_1, x_2, x_3) \in \mathbb{Z}^3$ if, and only if, $\delta_2 y_1 \equiv \delta_1 y_2 \pmod{\delta_3}$. The condition is clearly necessary: just rewrite Eq. (7) as

$$\begin{cases} y_1 = \delta_3 \gcd\{\alpha_2, \alpha_3\} x_1 - \delta_1 \gcd\{\alpha_1, \alpha_2\} x_3 \\ y_2 = \delta_3 \gcd\{\alpha_1, \alpha_3\} x_2 - \delta_2 \gcd\{\alpha_1, \alpha_2\} x_3. \end{cases}$$

Conversely, let us first note that each of the conditions $\alpha_1 = 0$, $\alpha_2 = 0$, or $\alpha_3 = 1$ easily yields solutions to Eq. (7). Assume none of them holds and denote, for $i = 1, 2$, by γ_i the inverse of $\delta_i \gcd\{\alpha_1, \alpha_2\}$ modulo $\alpha_3 / \gcd\{\alpha_i, \alpha_3\}$. Note that $\gamma_i \delta_i \gcd\{\alpha_1, \alpha_2\}$ is also 1 modulo δ_3 . Therefore, $\delta_2 y_1 \equiv \delta_1 y_2 \pmod{\delta_3}$ implies that $\gamma_1 y_1 \equiv \gamma_2 y_2 \pmod{\delta_3}$. Now let $\ell \in \mathbb{Z}$ such that $\gamma_1 y_1 - \gamma_2 y_2 = \ell \delta_3$, choose $p, q \in \mathbb{Z}$ such that $\gcd\{\alpha_2, \alpha_3\} p - \gcd\{\alpha_1, \alpha_3\} q = \ell$, and set:

$$\begin{aligned} x_3 &= -\gamma_1 y_1 + p \delta_3 \gcd\{\alpha_2, \alpha_3\} \\ &= -\gamma_2 y_2 + q \delta_3 \gcd\{\alpha_1, \alpha_3\}. \end{aligned} \quad (8)$$

For $i = 1, 2$, we have $\delta_i \gcd\{\alpha_1, \alpha_2\} x_3 + y_i \equiv 0 \pmod{\alpha_3 / \gcd\{\alpha_i, \alpha_3\}}$. Setting

$$\begin{aligned} x_1 &= \frac{\delta_1 \gcd\{\alpha_1, \alpha_2\} x_3 + y_1}{\delta_3 \gcd\{\alpha_2, \alpha_3\}}, \\ x_2 &= \frac{\delta_2 \gcd\{\alpha_1, \alpha_2\} x_3 + y_2}{\delta_3 \gcd\{\alpha_1, \alpha_3\}}, \end{aligned} \quad (9)$$

we thus get a solution $(x_1, x_2, x_3) \in \mathbb{Z}^3$ of Eq. (7). $\square$

Corollary 11. We have $\Gamma_\alpha = \mathbb{Z} \mathbf{g}_1 + \mathbb{Z} \mathbf{g}_2$, with

$$\mathbf{g}_1 = \delta_1 \mathbf{f}_1 + \delta_2 \mathbf{f}_2, \quad \mathbf{g}_2 = u \mathbf{f}_1 + v \mathbf{f}_2, \quad (10)$$

where $(u, v) \in \mathbb{Z}^2$ satisfies $\delta_1 v - \delta_2 u = \delta_3$.

Proof. Theorem 10 states that Γ_α is the union, for ℓ running through $\mathbb{Z}$, of the subsets $A_\ell = \{y_1 \mathbf{f}_1 + y_2 \mathbf{f}_2, (y_1, y_2) \in \mathbb{Z}^2, \delta_1 y_2 - \delta_2 y_1 = \ell \delta_3\}$ of the plane $\{\mathbf{x} \in \mathbb{R}^3, \langle \mathbf{e}_3, \mathbf{x} \rangle = 0\}$. Since $\gcd\{\delta_1, \delta_2\} = 1$ by definition, no set A_ℓ is empty. It follows that Γ_α is a sublattice of $\mathbb{Z}^2$, a basis of which may consist in a basis of A_0 together with any vector shifting A_0 to A_1 . $\square$

Another basis for Γ_α is of interest because direction $\mathbf{e}_2$ is conserved, namely:

$$\mathbf{h}_1 = \mathbf{f}_1 + s \delta_2 \gcd\{\alpha_1, \alpha_2\} \mathbf{f}_2, \quad \mathbf{h}_2 = \delta_3 \mathbf{f}_2, \quad (11)$$

where s is any integer satisfying $s \alpha_1 + t \alpha_3 = \gcd\{\alpha_1, \alpha_3\}$, together with a fitting integer t . We leave it to the reader to check that

$$\begin{pmatrix} \mathbf{g}_1 \\ \mathbf{g}_2 \end{pmatrix} = A \begin{pmatrix} \mathbf{h}_1 \\ \mathbf{h}_2 \end{pmatrix}, \quad A = \begin{pmatrix} \delta_1 & t \delta_2 \gcd\{\alpha_2, \alpha_3\} \\ u & s \gcd\{\alpha_1, \alpha_2\} + vt \gcd\{\alpha_2, \alpha_3\} \end{pmatrix}, \quad \det A = 1. \quad (12)$$

Considering an arithmetic discrete plane $\mathcal{P}$ with functional lattice Γ_α , a natural question arises: Given an element $\mathbf{y} \in \Gamma_\alpha$, how to recover the unique vector $\mathbf{x} \in \mathcal{P}$ such that $\pi_\alpha(\mathbf{x}) = \mathbf{y}$?

The height of $\mathcal{P}$ at $\mathbf{y} \in \Gamma_\alpha$ is defined as the third coordinate x_3 of $\mathbf{x} = \pi_\alpha^{-1}(\mathbf{y})$ and it is denoted by $H_{\mathcal{P}, \alpha}(\mathbf{y})$. Thus $H_{\mathcal{P}, \alpha}$ maps Γ_α to $\mathbb{Z}$. According to Eq. (9), the height of $\mathcal{P}$ at $\mathbf{y}$ entirely determines $\pi_\alpha^{-1}(\mathbf{y})$. It can be computed as follows.

Theorem 12. Let Γ_α be a functional lattice of $\mathcal{P} = \mathfrak{P}(\mathbf{v}, \mu, w)$ and let $\mathbf{y} \in \Gamma_\alpha$ with $\mathbf{y} = y_1 \mathbf{f}_1 + y_2 \mathbf{f}_2$. Then

$$H_{\mathcal{P}, \alpha}(\mathbf{y}) = -\lfloor m/w \rfloor + r \quad (13)$$

where

$$m = v_1 y_1 \gcd\{\alpha_1, \alpha_3\} + v_2 y_2 \gcd\{\alpha_2, \alpha_3\} + \mu \alpha_3, \quad (14)$$

and $r \in \{0, \dots, \alpha_3 - 1\}$ is given by any of the following expressions:

- (i) $r \equiv \lfloor -m/w \rfloor - a y_1 \gcd\{\alpha_1, \alpha_3\} - b y_2 \gcd\{\alpha_2, \alpha_3\} \pmod{\alpha_3}$, where a, b are any integers such that $a \alpha_1 + b \alpha_2 \equiv 1 \pmod{\alpha_3}$.
- (ii) If $\gcd\{w, \alpha_3\} = 1$ then $r \equiv \lfloor m/w \rfloor - m/w \pmod{\alpha_3}$.
- (iii) If $\alpha_3 = 1$ then $r = 0$.

Proof. Let $\mathbf{x} = (x_1, x_2, x_3) \in \mathcal{P}$ and let $\mathbf{y} = \pi_\alpha(\mathbf{x})$. According to Eq. (7) and $\langle \alpha, \mathbf{v} \rangle = w$, we have

$$\alpha_3 \langle \mathbf{v}, \mathbf{x} \rangle = m + x_3 w, \quad (15)$$

where m is given by Eq. (14). Substituting in Eq. (1) and dividing each member by w result in

$$0 \leq \frac{m}{w} + x_3 < \alpha_3. \quad (16)$$

Therefore, there exists a unique integer r such that $0 \leq r \leq \alpha_3 - 1$ and $x_3 = -\lfloor m/w \rfloor + r$. In particular, (iii) holds. Since $\gcd(\alpha) = 1$, integers a and b can be found such that $a \alpha_1 + b \alpha_2 \equiv 1 \pmod{\alpha_3}$. A linear combination of equations Eq. (7) gives

$$a y_1 \gcd\{\alpha_1, \alpha_3\} + b y_2 \gcd\{\alpha_2, \alpha_3\} \equiv -x_3 \pmod{\alpha_3},$$

so that (i) is proved. Finally, (ii) follows from Eq. (15). $\square$

In the following, we make use of (iii) in Theorem 12 in order to investigate the case of the best known classes of arithmetic discrete planes, namely the naive, the standard, and the graceful ones. Let $\mathcal{P} = \mathfrak{P}(\mathbf{v}, \mu, w)$ be an arithmetic discrete plane and let $\alpha \in \mathbb{Z}^3$ such that $\langle \alpha, \mathbf{v} \rangle = w$. We assume here that there exists $i \in \{1, 2, 3\}$ such that $\alpha_i = 1$,

say $\alpha_3 = 1$. Let us emphasize that, according to Eq. (11), $\Gamma_\alpha = \mathbb{Z}\mathbf{e}_1 + \mathbb{Z}\mathbf{e}_2$ in that latter case. Moreover, Theorem 12 yields a simple formula for the height of $\mathcal{P}$, that is,

$$H_{\mathcal{P},\alpha}(\mathbf{y}) = -\left\lfloor \frac{v_1 y_1 + v_2 y_2 + \mu}{w} \right\rfloor. \quad (17)$$

One thus obtains:

Corollary 13. *If $\alpha_3 = 1$, then the function $\pi_\alpha^{-1} : \Gamma_\alpha \rightarrow \mathcal{P}$ is defined, for all $\mathbf{y} \in \Gamma_\alpha$ with $\mathbf{y} = y_1 \mathbf{e}_1 + y_2 \mathbf{e}_2$, by*

$$\pi_\alpha^{-1}(\mathbf{y}) = \mathbf{y} - \left\lfloor \frac{v_1 y_1 + v_2 y_2 + \mu}{w} \right\rfloor \alpha. \quad (18)$$

Let us assume that $\mathbf{v} \in \mathbb{R}_+^3$ with $v_1 \leq v_2 \leq v_3$, which may be achieved by mild modifications of the coordinates axes. If $\mathcal{P}$ is either a naive or a standard discrete plane we can thus assume $\alpha_3 = 1$. If $\mathcal{P} = \mathfrak{P}(\mathbf{v}, \mu, w)$ is a graceful plane, that is, $w = \max(v_1 + v_2, v_3)$, then it is naive if $v_1 + v_2 \leq v_3$, otherwise we may assume $\alpha_1 = 1$. Therefore, we recover the following from Corollary 13, the formula being already known in the naive case:

Corollary 14. *Let $\mathcal{P} = \mathfrak{P}(\mathbf{v}, \mu, w)$ with $0 \leq v_1 \leq v_2 \leq v_3$.*

If $\mathcal{P}$ is naive, then $\alpha = \mathbf{e}_3$ is functional for $\mathfrak{P}$. For all $\mathbf{x} \in \mathcal{P}$, $\pi_\alpha(\mathbf{x}) = x_1 \mathbf{e}_1 + x_2 \mathbf{e}_2$ and for all $\mathbf{y} = y_1 \mathbf{e}_1 + y_2 \mathbf{e}_2 \in \Gamma_\alpha$,

$$\pi_\alpha^{-1}(\mathbf{y}) = y_1 \mathbf{e}_1 + y_2 \mathbf{e}_2 - \left\lfloor \frac{v_1 y_1 + v_2 y_2 + \mu}{v_3} \right\rfloor \mathbf{e}_3.$$

If $\mathcal{P}$ is standard, then $\alpha = \mathbf{e}_1 + \mathbf{e}_2 + \mathbf{e}_3$ is functional for $\mathcal{P}$. For all $\mathbf{x} \in \mathcal{P}$, $\pi_\alpha(\mathbf{x}) = (x_1 - x_3) \mathbf{e}_1 + (x_2 - x_3) \mathbf{e}_2$, and for all $\mathbf{y} = y_1 \mathbf{e}_1 + y_2 \mathbf{e}_2 \in \Gamma_\alpha$,

$$\pi_\alpha^{-1}(\mathbf{y}) = y_1 \mathbf{e}_1 + y_2 \mathbf{e}_2 - \left\lfloor \frac{v_1 y_1 + v_2 y_2 + \mu}{v_1 + v_2 + v_3} \right\rfloor (\mathbf{e}_1 + \mathbf{e}_2 + \mathbf{e}_3).$$

If $\mathcal{P}$ is graceful with $w = v_1 + v_2$, then $\alpha = \mathbf{e}_1 + \mathbf{e}_2$ is functional for $\mathfrak{P}$. For all $\mathbf{x} \in \mathfrak{P}$, $\pi_\alpha(\mathbf{x}) = (x_2 - x_1) \mathbf{e}_2 + x_3 \mathbf{e}_3$, and for all $\mathbf{y} = y_2 \mathbf{e}_2 + y_3 \mathbf{e}_3 \in \Gamma_\alpha$,

$$\pi_\alpha^{-1}(\mathbf{y}) = y_2 \mathbf{e}_2 + y_3 \mathbf{e}_3 - \left\lfloor \frac{v_2 y_2 + v_3 y_3 + \mu}{v_1 + v_2} \right\rfloor (\mathbf{e}_1 + \mathbf{e}_2).$$

5. Local configurations and m-cubes

5.1. First definitions

The aim of this section is to apply the previous results to the study of (m, n) -cubes and local configurations, general-

izing the study performed for rational naive planes in [31,30,20,32,13]. For the sake of consistency in the notation, we call them here **m**-cubes with $\mathbf{m} = (m_1, m_2)$ rather than (m, n) -cubes.

Let $\mathcal{P} = \mathfrak{P}(\mathbf{v}, \mu, w)$ be an arithmetic discrete plane and let $\alpha \in \mathbb{Z}^3$ such that $\gcd(\alpha) = 1$ and $\langle \alpha, \mathbf{v} \rangle = w$ (recall that if $\mathbf{v} \in \mathbb{Z}^3$ and $\gcd(\mathbf{v}) = 1$, then the existence of α is ensured by Proposition 9). The results in both the present and the following sections rely on Corollary 13. We shall assume that $\alpha_3 = 1$ in all that follows. Indeed, we heavily use the explicit and simple expression of the preimage of a point in Γ_α given by Eq. (18) obtained under the assumption $\alpha_3 = 1$. Note that, since $w = \alpha_1 v_1 + \alpha_2 v_2 + \alpha_3$, $\alpha_3, \alpha_3 = 1$ is equivalent to $w \in v_1 \mathbb{Z} + v_2 \mathbb{Z} + v_3$, i.e., $w - v_3 \in \gcd(v_1, v_2) \mathbb{Z}$ in the rational case.

Also note that there does not always exist a functional vector α with $\alpha_3 = 1$. Consider for instance the case $\mathbf{v} = (6, 10, 15)$ with $w = 20$: it is impossible to express w as $\alpha_1 v_1 + \alpha_2 v_2 + \alpha_3 v_3$ with one of the α_i 's equal to 1.

Let $\mathbf{m} \in (\mathbb{N}^*)^2$ be given. By **m**-cube we mean a local configuration in the discrete plane that can be observed thanks to π_α through an **m**-window in the functional lattice $\Gamma_\alpha = \mathbb{Z}\mathbf{e}_1 + \mathbb{Z}\mathbf{e}_2$ (see Fig. 2). More precisely,

Definition 15. Let $\mathbf{m} \in (\mathbb{N}^*)^2$ and $\mathbf{y} \in \Gamma_\alpha$. The **m**-cube $\mathcal{C}(\mathbf{y}, \mathbf{m})$ of $\mathcal{P}$ is defined as the following subset of $\mathcal{P}$:

$$\mathcal{C}(\mathbf{y}, \mathbf{m}) = \pi_\alpha^{-1}(\mathbf{y} + \mathbf{z}), \quad \mathbf{z} \in \llbracket 0, m_1 - 1 \rrbracket \mathbf{e}_1 + \llbracket 0, m_2 - 1 \rrbracket \mathbf{e}_2$$

Two **m**-cubes $\mathcal{C}$ and $\mathcal{C}'$ are called *translation equivalent* if there exists a vector $\mathbf{z} \in \mathbb{Z}^3$ such that $\mathcal{C}' = \mathcal{C} + \mathbf{z}$.

In order to enumerate the different types of **m**-cubes that occur in $\mathcal{P}$, that is, the different equivalence classes for the translation equivalence, we represent them as local configurations as follows. Recall that $H_{\mathcal{P},\alpha}$ is defined in Eq. (17).

Definition 16. Let $\mathbf{m} = (m_1, m_2) \in (\mathbb{N}^*)^2$. A $m_1 \times m_2$ -rectangular word $L = [L_{i_1, i_2}]_{(i_1, i_2) \in \llbracket 0, m_1 - 1 \rrbracket \times \llbracket 0, m_2 - 1 \rrbracket}$ over the infinite alphabet $\mathbb{Z}$ is called an **m**-local configuration of $\mathcal{P}$ if there exists $\mathbf{y} \in \mathbb{Z}^2$ such that

$$L = H_{\mathcal{P},\alpha}(\mathbf{z}) - H_{\mathcal{P},\alpha}(\mathbf{y})_{\mathbf{z}} \in \llbracket 0, m_1 - 1 \rrbracket \mathbf{e}_1 + \llbracket 0, m_2 - 1 \rrbracket \mathbf{e}_2. \quad (19)$$

Such a local configuration Eq. (19) is denoted by $LC(\mathbf{y}, \mathbf{m})$.

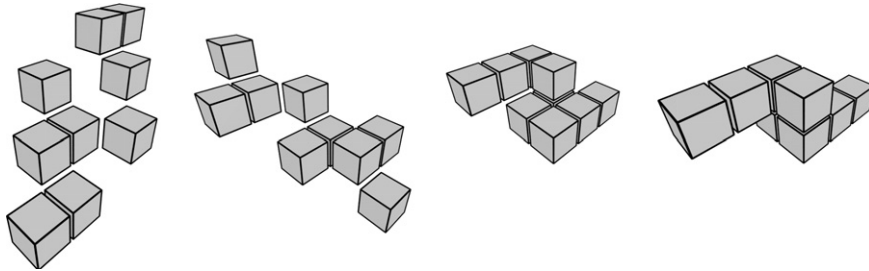


Fig. 2. From left to right: the $(3,3)$ -cube of $\mathfrak{P}(\mathbf{v}, 0, 9)$ (resp. $\mathfrak{P}(\mathbf{v}, 0, 11)$, $\mathfrak{P}(\mathbf{v}, 0, 21)$, $\mathfrak{P}(\mathbf{v}, 0, 37)$) centered on $(0,0,0)$, where $\mathbf{v} = 6\mathbf{e}_1 + 10\mathbf{e}_2 + 15\mathbf{e}_3$, and projected along the vector $-\mathbf{e}_1 + \mathbf{e}_3$ (resp. $\mathbf{e}_1 - \mathbf{e}_2 + \mathbf{e}_3$, $\mathbf{e}_1 + \mathbf{e}_3$, $2\mathbf{e}_1 + \mathbf{e}_2 + \mathbf{e}_3$).

Let us note that a local configuration is a plane partition. Indeed a *plane partition* of $N \in \mathbb{N}$ is a rectangular word $w = [w_{i_1, i_2}]_{(i_1, i_2) \in [0, m_1-1] \times [0, m_2-1]}$ over the infinite alphabet $\mathbb{N}$ satisfying $N = \sum_{i,j} w_{i,j}$ and, for all $i_1 \in [0, m_1-1]$ and $i_2 \in [0, m_2-1]$, $\max\{w_{i_1+1, i_2}, w_{i_1, i_2+1}\} \leq w_{i_1, i_2}$.

Notation 17. Let $L = [L_{i_1, i_2}]_{(i_1, i_2) \in [0, m_1-1] \times [0, m_2-1]}$ be a local configuration of size $m_1 \times m_2$. In all that follows, the notation $L \bmod 2$ stands for the $m_1 \times m_2$ rectangular word $[L_{i_1, i_2} \bmod 2]_{(i_1, i_2) \in [0, m_1-1] \times [0, m_2-1]}$.

Example 18. Let us consider the arithmetic discrete plane $\mathcal{P} = \mathfrak{P}(\mathbf{v}, \mu, w)$ with $\mathbf{v} = 6\mathbf{e}_1 + 10\mathbf{e}_2 + 15\mathbf{e}_3$, $\mu = 0$, and $w = 21$. Then $\alpha = \mathbf{e}_1 + \mathbf{e}_3$ is a functional vector of $\mathcal{P}$. The local configuration $LC(\mathbf{e}_1 + \mathbf{e}_2, 3\mathbf{e}_1 + 3\mathbf{e}_2)$ of $\mathcal{P}$ and its preimage by π_α^{-1} are illustrated in Fig. 3.

Remark 19. A local configuration is not necessarily (simply-)connected. For instance, let us consider $\mathcal{P} = \mathfrak{P}(\mathbf{v}, \mu, w)$ with $\mathbf{v} = \mathbf{e}_1 + 2\mathbf{e}_2 + 7\mathbf{e}_3$ and $w = 4$. Then $\alpha = \mathbf{e}_1 - 2\mathbf{e}_2 + \mathbf{e}_3$ is a functional vector of $\mathcal{P}$, and $LC(-\mathbf{e}_1 - \mathbf{e}_2, \mathbf{e}_1 + \mathbf{e}_2)$ is not connected (see Fig. 4).

5.2. A coding as a two-dimensional word

According to [34], we introduce a two-dimensional word coding in a natural way the parity of the heights $H_{\mathcal{P}, \alpha}(\mathbf{y})$, for $\mathbf{y}$ in the lattice $\Gamma_\alpha = \mathbb{Z}\mathbf{e}_1 + \mathbb{Z}\mathbf{e}_2$. Indeed, for a naive discrete plane $\mathcal{P}$, it is well known that, given two points $\mathbf{x}$ and $\mathbf{x}'$ of $\mathcal{P}$ such that their projections by π_α are 4-connected in the functional plane, then $|x_3 - x'_3| \leq 1$. In other words, the difference between the heights of $\mathbf{x}$ and $\mathbf{x}'$ is at most 1. A quite unexpected fact is that this property holds for any arithmetic discrete plane with $\alpha_3 = 1$. More precisely, it is easy to see that, for all $\mathbf{y} \in \Gamma_\alpha$ and $i = 1, 2$,

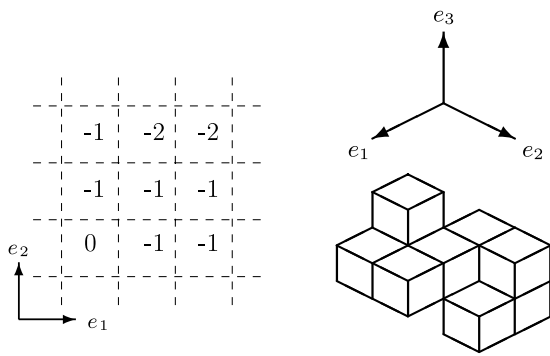


Fig. 3. From left to right: a local configuration of the discrete plane $\mathfrak{P}((6, 10, 15), 0, 21)$ and its corresponding preimage by $\pi_{\mathbf{e}_1 + \mathbf{e}_3}^{-1}$.

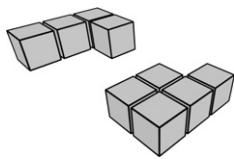


Fig. 4. The $(3,3)$ -cube of $\mathfrak{P}(\mathbf{v}, 0, 4)$ centered on $(0,0,0)$ with $\mathbf{v} = \mathbf{e}_1 + 2\mathbf{e}_2 + 7\mathbf{e}_3$.

$H_{\mathcal{P}, \alpha}(\mathbf{y} + \mathbf{e}_i) - H_{\mathcal{P}, \alpha}(\mathbf{y})$ takes only two values, namely $-\lfloor v_i/w \rfloor$ and $-\lfloor v_i/w \rfloor - 1$. In each case, one of these values is odd, whereas the other one is even; we define E_1 and O_1 to be respectively the even and the odd value taken by $-\lfloor v_i/w \rfloor$ and $-\lfloor v_i/w \rfloor - 1$; we similarly define E_2 and O_2 . It is now natural to introduce the following two-dimensional word of parity of heights by identifying Γ_α to $\mathbb{Z}^2$:

$$U = (U_{i_1, i_2})_{(i_1, i_2) \in \mathbb{Z}^2} = (H_{\mathcal{P}, \alpha}(\mathbf{y}) \bmod 2)_{\mathbf{y} \in \mathbb{Z}^2} \in \{0, 1\}^{\mathbb{Z}^2}. \quad (20)$$

Lemma 20. The two-dimensional word U satisfies, for each $(i_1, i_2) \in \mathbb{Z}^2$

$$U_{i_1, i_2} = 0 \text{ if, and only if, } v_1 i_1 + v_2 i_2 + \mu \bmod 2w \in [0, w[.$$

Proof. According to Eq. (17), $U_{i_1, i_2} = 0$ if, and only if, $\lfloor \frac{v_1 i_1 + v_2 i_2 + \mu}{w} \rfloor$ is even, that is, $v_1 i_1 + v_2 i_2 + \mu \bmod 2w \in [0, w[$. $\square$

The word U is a two-dimensional Rote word; one-dimensional Rote words have been introduced in [29]; they are defined as the infinite words over the alphabet $\{0, 1\}$ that have exactly $2n$ factors of length n for every positive integer n , and whose set of factors is closed under complementation, i.e., every word obtained by interchanging zeros and ones in a factor of the infinite word u is still a factor of u ; two-dimensional Rote words have been studied for instance in [34, 6].

Definition 21. Let $W = [w_{i_1, i_2}]_{(i_1, i_2) \in [0, m_1-1] \times [0, m_2-1]}$ be a rectangular word of size $m_1 \times m_2$ over $\{0, 1\}$. We define the *complement* $\overline{W}$ of W as follows:

$$\overline{W} = [\overline{w_{i_1, i_2}}]_{(i_1, i_2) \in [0, m_1-1] \times [0, m_2-1]}, \quad \text{where } \overline{1} = 0 \text{ and } \overline{0} = 1.$$

We introduce the following equivalence relation defined on the set of rectangular factors of U of a given size:

$$V \sim W \text{ if, and only if, } V \in \{W, \overline{W}\}.$$

We have the following theorem, inspired by [34] where it is stated under the assumption $\dim_{\mathbb{Q}} v = 3$:

Theorem 22. Let $\mathcal{P} = \mathfrak{P}(\mathbf{v}, \mu, w)$ be a discrete plane that admits a functional vector α satisfying $\alpha_3 = 1$. There is a natural bijection between the equivalence classes of the relation $\sim$ on the rectangular factors of the two-dimensional word U of size $\mathbf{m} = (m_1, m_2)$ and the $\mathbf{m}$ -local configurations of $\mathfrak{P}$. Furthermore, the $\mathbf{m}$ -local configurations of $\mathfrak{P}$ are in one-to-one correspondence with the translation equivalence classes of $\mathbf{m}$ -cubes of $\mathcal{P}$.

Proof. Let $\mathbf{m} = (m_1, m_2) \in (\mathbb{N}^*)^2$. Consider the local configuration $L = LC(\mathbf{y}, \mathbf{m})$ with the notation of Definition 16; one has either $L \bmod 2$ or $\overline{L} \bmod 2$ factor of the two-dimensional word U . Indeed if $H_{\mathcal{P}, \alpha}(\mathbf{y})$ is even, then $L \bmod 2$ is a factor of the two-dimensional word U ; otherwise, $H_{\mathcal{P}, \alpha}(\mathbf{y})$ is odd and $\overline{L} \bmod 2$ is a factor of U .

Conversely, let us show how to reconstruct an $\mathbf{m}$ -local configuration L from a given $m_1 \times m_2$ -factor W of U such that $L \bmod 2$ is either W or $\overline{W}$.

Let us first assume that $w_{0,0} = 0$. We define a plane partition H by induction: we set $H(0) = 0$; let $(i_1, i_2) \in \llbracket 0, m_1 - 1 \rrbracket \times \llbracket 0, m_2 - 1 \rrbracket$. If $w_{i_1+1, i_2} = w_{i_1, i_2}$, then we set $H_{i_1+1, i_2} = H_{i_1, i_2} + E_1$. Otherwise, we set $H_{i_1+1, i_2} = H_{i_1, i_2} + O_1$. Similarly, if $w_{i_1, i_2+1} = w_{i_1, i_2}$, we set $H_{i_1, i_2+1} = H_{i_1, i_2} + E_2$. Otherwise, we set $H_{i_1, i_2+1} = H_{i_1, i_2} + O_2$.

The plane partition H is a local configuration of $\mathcal{P}$; indeed, if W occurs at index (k_1, k_2) in U , then $H = LC(\mathbf{y}, \mathbf{m})$ with $\mathbf{y} = k_1 \mathbf{e}_1 + k_2 \mathbf{e}_2$, and $W = (H \bmod 2)$ since $H(\mathbf{y})$ is even (we have $w_{0,0} = 0$).

Now, if $w_{0,0} = 1$ we apply the same reconstruction process to $\overline{W}$. We recover again a local configuration $LC(\mathbf{y}, \mathbf{m})$ such that $\overline{W} = (LC(\mathbf{y}, \mathbf{m}) \bmod 2)$, which ends the proof of the existence of a one-to-one correspondence between equivalence classes of $\sim$ and $\mathbf{m}$ -local configurations.

It is immediate to associate an $\mathbf{m}$ -local configuration to an $\mathbf{m}$ -cube by considering the third coordinate of the points of the $\mathbf{m}$ -cube. Conversely, given a local configuration $L = [L_{i_1, i_2}]$ of $\mathcal{P}$ of size $m_1 \times m_2$, we can associate to it the following subset of $\mathbb{Z}^3$: $\{i_1 \mathbf{e}_1 + i_2 \mathbf{e}_2 + L_{i_1, i_2} \mathbf{e}_3, (i_1, i_2) \in \llbracket 0, m_1 - 1 \rrbracket \times \llbracket 0, m_2 - 1 \rrbracket\}$. It remains to note that there exists $\mathbf{z} \in \mathbb{Z}^3$ such that $\mathbf{z} + \{i_1 \mathbf{e}_1 + i_2 \mathbf{e}_2 + L_{i_1, i_2} \mathbf{e}_3, (i_1, i_2) \in \llbracket 0, m_1 - 1 \rrbracket \times \llbracket 0, m_2 - 1 \rrbracket\} \subset \mathcal{P}$, to conclude the proof. $\square$

5.3. Basic properties of the two-dimensional word U

This section is devoted to the study of combinatorial properties of the two-dimensional word U from which we will deduce geometric properties of the local configurations and $\mathbf{m}$ -cubes.

The two-dimensional word U is called *periodic* if there exists a non-zero vector $\mathbf{z} \in \mathbb{Z}^2$, called *period*, such that $U_{\mathbf{y}+\mathbf{z}} = U_{\mathbf{y}}$ for every $\mathbf{y} \in \Gamma_{\alpha}$. The set of its periods is a lattice whose rank is determined by the dimension over $\mathbb{Q}$ of the vector space generated by the coordinates of the vector $\mathbf{v}$:

Proposition 23. *The rank of the lattice of periods of the two-dimensional word U is $3 - \dim_{\mathbb{Q}} \mathbf{v}$.*

Proof. The proposition can easily be deduced from the following observation: for a non-zero element $(\ell_1, \ell_2) \in \mathbb{Z}^2$, $\frac{v_1 \ell_1 + v_2 \ell_2}{w} \in \mathbb{Q}$ if, and only if, there exists a non-zero integer k such that $(k v_1, k v_2)$ is a period of U . Let us prove this latter statement. If $\frac{v_1 \ell_1 + v_2 \ell_2}{w} \in \mathbb{Q}$, then there exists a non-zero integer k such that $v_1(k \ell_1) + v_2(k \ell_2) \in 2w\mathbb{Z}$, hence $(k \ell_1, k \ell_2)$ is a period. Otherwise, if $\frac{v_1 \ell_1 + v_2 \ell_2}{w} \notin \mathbb{Q}$, then the density of $(v_1(k \ell_1) + v_2(k \ell_2) \bmod 2w)_{k \in \mathbb{Z}}$ yields the desired result. $\square$

A key ingredient in the combinatorial study of the two-dimensional word U is the following lemma. This is a standard approach in symbolic dynamics for the study of sturmian words [26], and more generally, of infinite words coding rotations as well as double rotations [5,6]

in the torus $\mathbb{R}/\mathbb{Z}$. This lemma will allow us in Sections 5.4 and 5.5 to derive enumeration as well as statistical properties for the factors of U , and thus for $\mathbf{m}$ -cubes of $\mathcal{P}$.

In all that follows intervals are considered as intervals of the torus $\mathbb{R}/2w\mathbb{Z}$.

Lemma 24. *Let $W = [w_{i_1, i_2}]_{(i_1, i_2) \in \llbracket 0, m_1 - 1 \rrbracket \times \llbracket 0, m_2 - 1 \rrbracket}$ be a rectangular word of size $m_1 \times m_2$ over $\{0, 1\}$. Let $I_0 = [0, w[$ and $I_1 = [w, 2w[$. Let*

$$I_W = \bigcap_{i_1=0}^{m_1-1} \bigcap_{i_2=0}^{m_2-1} (I_{w_{i_1, i_2}} - (v_1 i_1 + v_2 i_2) \bmod 2w).$$

The set I_W is a left-closed right-open interval of $[0, 2w[$.

Let $\mathcal{P} = \mathfrak{P}(\mathbf{v}, \mu, w)$ be a discrete plane with $w - v_3 \in v_1 \mathbb{Z} + v_2 \mathbb{Z}$. If $\dim_{\mathbb{Q}} \mathbf{v} > 1$ or $\mathfrak{P}$ is rational and $\gcd(v_1, v_2, 2w) = 1$, then a rectangular word W over $\{0, 1\}$ is a factor of U if, and only if, $I_W \neq \emptyset$. Otherwise, if $\mathfrak{P}$ is rational and $\gcd(v_1, v_2, 2w) = 2$, then a rectangular word W over $\{0, 1\}$ is a factor of U if, and only if, I_W contains an integer with the same parity as μ .

Proof. The proof is inspired by [5] where more details are given. It is easily shown that the sets I_W consist of finite unions of left-closed right-open intervals. We then show by induction that each set I_W is an interval. Indeed this can be deduced from the following remark: if I and J are two left-closed right-open intervals of $\mathbb{R}/\mathbb{Z}$ whose intersection is non-connected, then the sum of their lengths is strictly larger than 1.

Let $W = [w_{i_1, i_2}]_{(i_1, i_2) \in \llbracket 0, m_1 - 1 \rrbracket \times \llbracket 0, m_2 - 1 \rrbracket}$ be a rectangular word of size $m_1 \times m_2$ over $\{0, 1\}$. One first checks that W occurs in the sequence U at index (k_1, k_2) if, and only if, $v_1 k_1 + v_2 k_2 + \mu \in I_W$. Indeed, W occurs in the sequence U at index (k_1, k_2) if, and only if, $\forall (i_1, i_2) \in \llbracket 0, m_1 - 1 \rrbracket \times \llbracket 0, m_2 - 1 \rrbracket$, $w_{i_1, i_2} = U_{k_1+i_1, k_2+i_2}$, that is, $v_1(k_1 + i_1) + v_2(k_2 + i_2) + \mu \bmod 2w \in I_{w_{i_1, i_2}}$, or else, $v_1 k_1 + v_2 k_2 + \mu \bmod 2w \in I_{w_{i_1, i_2}} - (v_1 i_1 + v_2 i_2)$.

Conversely, let W be a rectangular word over $\{0, 1\}$ such that $I_W \neq \emptyset$.

If $\dim_{\mathbb{Q}} \mathbf{v} > 1$, then the density of $(v_1 k_1 + v_2 k_2 + \mu \bmod 2w)_{(k_1, k_2) \in \mathbb{Z}^2}$ in $[0, 2w[$ and the fact that I_W has non-empty interior imply that there exists (k_1, k_2) such that $v_1 k_1 + v_2 k_2 + \mu \in I_W$, hence W occurs in U at index (k_1, k_2) .

Next assume $\mathcal{P}$ is rational, that is, $\dim_{\mathbb{Q}} \mathbf{v} = 1$. One has $\gcd\{v_1, v_2, 2w\} \in \{1, 2\}$ since $w = \alpha_1 v_1 + \alpha_2 v_2 + v_3$ and $\gcd(v) = 1$. Let us note that if $\gcd\{v_1, v_2, 2w\} = 2$, then w is odd since $\gcd(v_1, v_2, w) = 1$.

Since the sets I_W are semi-open intervals of integer lengths, then they are non-empty as soon as they contain an integer point. Let $k \in I_W$. If $\gcd\{v_1, v_2, 2v_3\} = 1$, then there exists (k_1, k_2) such that $v_1 k_1 + v_2 k_2 + \mu \equiv k \bmod 2w$. Assume $\gcd\{v_1, v_2, 2v_3\} = 2$. Then for every (k_1, k_2) , $v_1 k_1 + v_2 k_2 + \mu \bmod 2w$ has the same parity as μ . Hence there exists (k_1, k_2) such that $v_1 k_1 + v_2 k_2 + \mu \equiv k \bmod 2w$ if k has the same parity as μ . $\square$

Let us recall [34] that the set of factors of the two-dimensional word U is closed under complementation (see Definition 21) under the assumption $\dim_{\mathbb{Q}} v = 3$. This still holds true if $\dim_{\mathbb{Q}} v > 1$. But in the rational case one has to be more cautious:

Proposition 25. *Let W be a rectangular factor of U . One has $I_{\overline{W}} = I_W + w$. Let $\mathcal{P} = \mathfrak{P}(v, \mu, w)$ be a discrete plane with $w - v_3 \in v_1\mathbb{Z} + v_2\mathbb{Z}$.*

If either $\dim_{\mathbb{Q}}(v_1, v_2, w) \geq 2$ or if $\mathcal{P}$ is rational and $\gcd(v_1, v_2, 2w) = 1$, then the language of U is closed under complementation.

Otherwise, assume that $\mathcal{P}$ is rational and $\gcd(v_1, v_2, 2w) = 2$. Let W be a rectangular factor of the two-dimensional word U . Then $\overline{W}$ is a factor of U if, and only if, I_W contains both an even and an odd integer.

Proof. One easily checks from the definition of I_W that $I_{\overline{W}} = I_W + w$. Hence $I_W \neq \emptyset$ if, and only if, $I_{\overline{W}} \neq \emptyset$. Furthermore, in the rational case, when both intervals are non-empty, then I_W and $I_{\overline{W}}$ contain points with different parities, since w is odd. We thus conclude thanks to Lemma 24. $\square$

Corollary 26. *A rectangular word L over the infinite alphabet $\mathbb{Z}$ occurs as a local configuration of $\mathcal{P}$ if, and only if, $I_{L \bmod 2} \neq \emptyset$.*

Proof. If the rectangular factor L occurs in $\mathcal{P}$, then $I_{L \bmod 2} \neq \emptyset$, according to Lemma 24. Conversely, let L be rectangular factor such that $I_{L \bmod 2} \neq \emptyset$. We set $W = L \bmod 2$. According to Theorem 22, it is sufficient to prove that either W or $\overline{W}$ is a factor of U . In the irrational case or in the rational case under the assumption $\gcd(v_1, v_2, 2w) = 1$, then $I_W \neq \emptyset$ implies W is factor of U , by Lemma 24. In the rational case with $\gcd(v_1, v_2, 2w) = 1$, one concludes by using $I_{\overline{W}} = I_W + w$ and by noticing that w is odd. $\square$

5.4. Enumeration of local configurations

Let us now investigate the enumeration of $\mathbf{m}$ -cubes ($\mathbf{m} = (m_1, m_2)$) occurring in a given arithmetic plane. The number of $(3, 3)$ -cubes included in a given rational naive arithmetic discrete plane has been proved to be at most 9 in [31]. More generally, in [28,20], the authors proved that, given a rational naive arithmetic discrete plane $\mathcal{P}$, $\mathcal{P}$ contains at most $m_1 m_2$ $\mathbf{m}$ -cubes (to be more precise, translation equivalence classes of $\mathbf{m}$ -cubes). In [20] local configurations which are non-necessarily rectangular are also considered. In the following theorem, we show that this property also holds in our framework. For the sake of simplicity, we omit to mention that we consider translation equivalence classes of $\mathbf{m}$ -cubes:

Theorem 27. *Let $\mathcal{P} = \mathfrak{P}(v, \mu, w)$ be a discrete plane with $w - v_3 \in v_1\mathbb{Z} + v_2\mathbb{Z}$. Let $\mathbf{m} = (m_1, m_2) \in (\mathbb{N}^*)^2$. Then, $\mathcal{P}$ contains at most $m_1 m_2$ $\mathbf{m}$ -cubes. More precisely, one has:*

1. *If $\dim_{\mathbb{Q}} v = 1$, $v \in \mathbb{Z}^3$, $\mu \in \mathbb{Z}$, $w \in \mathbb{Z}$ and $\gcd(v) = 1$, then $\mathcal{P}$ contains at most w $\mathbf{m}$ -cubes for every $\mathbf{m} = (m_1, m_2) \in (\mathbb{N}^*)^2$. Moreover, for m_1 and m_2 large enough, $\mathcal{P}$ contains exactly w $\mathbf{m}$ -cubes.*
2. *Let us assume $\dim_{\mathbb{Q}} v = 2$. Let $(p_1, p_2) \in \mathbb{Z}^2$ be a generator of the lattice of periods of the two-dimensional word U . Then $\mathcal{P}$ contains at most $m_1|p_2| + m_2|p_1| - \min\{m_1, |p_1|\} - |p_2| + m_2|p_1| - \min\{m_1, |p_1|\} \min\{m_2, |p_2|\}$ $\mathbf{m}$ -cubes for $(m_1, m_2) \in \mathbb{N}^2$.*
3. *If $\dim_{\mathbb{Q}} v = 3$, then $\mathcal{P}$ contains exactly $m_1 m_2$ $\mathbf{m}$ -cubes for every $\mathbf{m} = (m_1, m_2) \in (\mathbb{N}^*)^2$.*

Let us note that the bounds upon which the previous results hold for m_1 and m_2 can be explicitly computed in terms of v and w .

Proof. According to Lemma 24 and Corollary 26, it amounts to count the number of intervals I_W . The extremal points of the intervals I_W for W factor of size $m_1 \times m_2$ of the sequence U belong to $-(i_1 v_1 + i_2 v_2)$ and $-(i_1 v_1 + i_2 v_2) + w$, for $0 \leq i_1 \leq m_1 - 1$ and $0 \leq i_2 \leq m_2 - 1$. There are at most $2m_1 m_2$ such points. The upper bound $m_1 m_2$ thus follows from Theorem 22.

- (1) Assume $\dim_{\mathbb{Q}} v = 1$. Then the extremal points of the intervals I_W are among the integers $\{0, 1, \dots, 2w - 1\}$ and the bound w follows from Theorem 22. To state the second assertion, it remains to show that each point of the set $\{0, 1, \dots, 2w - 1\}$ can be expressed as $-(i_1 v_1 + i_2 v_2) \bmod 2w$ or $-(i_1 v_1 + i_2 v_2) + w \bmod 2w$, with $(i_1, i_2) \in (\mathbb{N}^*)^2$. If $\gcd(v_1, v_2, 2w) = 1$, the statement follows from Bezout's Lemma. Let us assume now that $\gcd(v_1, v_2, 2w) = 2$. Then w is odd and the integers $-(i_1 v_1 + i_2 v_2) \bmod 2w$, for all $(i_1, i_2) \in (\mathbb{N}^*)^2$, are the even elements of $\{0, 1, \dots, 2w - 1\}$ while the integers $-(i_1 v_1 + i_2 v_2) + w \bmod 2w$, for all $(i_1, i_2) \in (\mathbb{N}^*)^2$, are the odd ones. We then apply Proposition 25.
- (2) Assume that $\dim_{\mathbb{Q}} v = 2$. According to Proposition 23, the lattice of periods of U has dimension 1. Let (p_1, p_2) be a generator of this lattice. One checks that there are at most $2m_1|p_2| + 2m_2|p_1| - 2 \min\{m_1, |p_1|\} - \min\{m_2, |p_2|\}$ factors of U of size $m_1 \times m_2$. We then apply Proposition 25.
- (3) Assume now $\dim_{\mathbb{Q}} v = 3$. Then there are exactly $2m_1 m_2$ points of the form $-(i_1 v_1 + i_2 v_2)$ and $-(i_1 v_1 + i_2 v_2) + w$, hence $2m_1 m_2$ rectangular factors of U of size $m_1 \times m_2$. We then apply Proposition 25. $\square$

5.5. Statistical properties

The frequency of occurrence of a rectangular word W in U is defined as the limit, if it exists, of the number of occurrences of W in the central square factor of U

$$f(W) = \lim_{n \rightarrow \infty} \frac{\text{Card}\{(k_1, k_2) \in \llbracket -n, n \rrbracket^2, W \text{ occurs at index } (k_1, k_2) \text{ in } U\}}{(2n+1)^2}.$$

The frequency of occurrence of a translation equivalence class of an $\mathbf{m}$ -cube $\mathcal{C}$ in $\mathcal{P}$ is defined as the limit, if it exists, of the number of occurrences of integer translates of C in the central square pattern of $\mathcal{P}$:

$$f(\mathcal{C}) = \lim_{n \rightarrow \infty} \frac{1}{(2n+1)^2} \cdot \text{Card}\{\mathbf{y} \in \llbracket -n, n \rrbracket \mathbf{e}_1 + \llbracket -n, n \rrbracket \mathbf{e}_2, \pi_{\alpha}^{-1}(\mathbf{y} + \mathbf{z}) = \mathcal{C}, \text{ for } \mathbf{z} \in \llbracket 0, m_1 - 1 \rrbracket \mathbf{e}_1 + \llbracket 0, m_2 - 1 \rrbracket \mathbf{e}_2\}. \quad (21)$$

We prove below that the frequencies of occurrence of rectangular factors in U do exist, and that they have a simple expression in terms of the lengths of the intervals I_W . We thus recover the corresponding result for $\mathbf{m}$ -cubes. This gives us a simple algorithmic way of computing these frequencies.

Theorem 28. *Let $\mathfrak{P}(\mathbf{v}, \mu, w)$ be a discrete plane such that $w - v_3 \in \gcd(v_1, v_2)\mathbb{Z}$. Let $\mathcal{C}$ be an $\mathbf{m}$ -cube. Let L be the corresponding local configuration. We set $W = L \bmod 2$. Then the frequency $f(\mathcal{C})$ exists and satisfies*

$$f(\mathcal{C}) = \frac{|I_W|}{w}.$$

Proof. Let $\mathcal{C}$ be an $\mathbf{m}$ -cube of $\mathcal{P}$. Let L be the corresponding local configuration. We set $W = L \bmod 2$. According to Theorem 22 and Lemma 24, the cardinality in Eq. (21) also reads

$$\text{Card}\{v_1 k_1 + v_2 k_2 + \mu \in I_W \cup I_{\overline{W}}, (k_1, k_2) \in \llbracket -n, n \rrbracket^2\}.$$

Indeed $I_W = I_{\overline{W}} + w$, by Proposition 25, hence $I_W \cap I_{\overline{W}} = \emptyset$.

Let us first assume that $\dim_{\mathbb{Q}} \mathbf{v} \geq 2$. The sequence $(n\alpha)_{n \in \mathbb{Z}}$ for α irrational is well-distributed, that is, given an interval I of $\mathbb{R}/\mathbb{Z}$,

$$\lim_{n \rightarrow \infty} \frac{\text{Card}\{-n \leq i \leq n, i\alpha + \gamma \in I\}}{2n+1} = |I|$$

uniformly in γ [23]; hence $f(W) = \frac{|I_W|}{2w} = f(\overline{W})$. We deduce that $f(\mathcal{C}) = f(W) + f(\overline{W}) = 2 \frac{|I_W|}{2w} = \frac{|I_W|}{w}$.

Let us next assume that $\dim_{\mathbb{Q}} \mathbf{v} = 1$.

If $\gcd\{v_1, v_2, 2w\} = 1$, then one checks that for every $k \in [0, 2w[$,

$$\lim_{n \rightarrow \infty} \frac{\text{Card}\{(k_1, k_2) \in \llbracket -n, n \rrbracket^2, v_1 k_1 + v_2 k_2 + \mu \equiv k \bmod 2w\}}{(2n+1)^2} = \frac{1}{2w}.$$

Hence $f(W) = \frac{|I_W|}{2w} = f(\overline{W})$, and the result follows.

If $\gcd\{v_1, v_2, 2w\} = 2$, let us recall that this implies that w is odd. Then one checks that for every integer $k \in [0, 2w[$ with the same parity as μ ,

$$\lim_{n \rightarrow \infty} \frac{\text{Card}\{(k_1, k_2) \in \llbracket -n, n \rrbracket^2, v_1 k_1 + v_2 k_2 + \mu \equiv k \bmod 2w\}}{(2n+1)^2} = \frac{1}{w},$$

otherwise, $\text{Card}\{(k_1, k_2) \in \llbracket -n, n \rrbracket^2, v_1 k_1 + v_2 k_2 + \mu \equiv k \bmod 2w\} = 0$ for every n . Hence $f(W)$ (resp. $f(\overline{W})$) is equal to the number of integers in I_W with the same parity (resp. with a different parity) as μ divided by w . Furthermore, an integer $k \in \mathbb{Z}$ with the same parity as μ belongs to I_W if, and only if, the integer $k + w$ (which has a different parity) belongs to $I_{\overline{W}}$. One has $f(C) = f(W) + f(\overline{W})$. This implies that $f(C)$ is equal to the total number of integers of I_W (or equivalently of $I_{\overline{W}}$) divided by w , which ends the proof. We thus get $f(\mathcal{C}) = \frac{|I_W|}{w}$. $\square$

Remark 29. The same approach can be used to study stability by centrosymmetry of local configurations (see Fig. 5), in the flavour of [6], where the set of factors of U is proved to be closed under centrosymmetry if $\dim_{\mathbb{Q}}(\mathbf{v}) = 3$. We can similarly prove that if $\mathfrak{P}(\mathbf{v}, \mu, w)$ is a discrete plane such that $w - v_3 \in \gcd(v_1, v_2)$, then the set of translation equivalence classes of $\mathbf{m}$ -cubes is closed under centrosymmetry. Furthermore, centrosymmetric $\mathbf{m}$ -cubes have the same frequency.

6. From functional arithmetic discrete planes to naive ones

Let $\mathcal{P} = \mathfrak{P}(\mathbf{v}, \mu, w)$ be a rational arithmetic discrete plane with $\mathbf{v} \in \mathbb{N}^3$, $\mu \in \mathbb{Z}$ and $w \in \mathbb{N}^*$. Recall that such an assumption is not restrictive since the isometry group of the unit cube $[-0.5, 0.5]^3$ acts on the set of arithmetic discrete planes.

We assume furthermore that there exists a functional vector $\alpha \in \mathbb{Z}^3$ of $\mathcal{P}$ satisfying $\alpha_3 = 1$ and that $w \geq \max\{v_1, v_2\}$. The aim of this section is to show that, under these hypotheses, $\mathcal{P}$ can be considered as a naive plane up to a change of basis in $\mathbb{Z}^3$. We then discuss two applications.

Lemma 30. *Let $\mathcal{P} = \mathfrak{P}(\mathbf{v}, \mu, w)$ be a rational arithmetic discrete plane with $\mathbf{v} \in \mathbb{N}^3$, $\mu \in \mathbb{Z}$ and $w \in \mathbb{N}^*$ and such that $w \geq \max\{v_1, v_2\}$. Let $\alpha \in \mathbb{Z}^3$ be a functional vector of $\mathcal{P}$. If $\alpha_3 = 1$ then $\mathcal{P}$ is a naive discrete plane in $\mathbb{Z}^3$ endowed with the basis $\{\mathbf{e}_1, \mathbf{e}_2, \alpha\}$.*

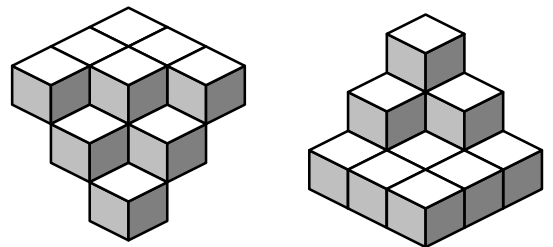


Fig. 5. From left to right: a (3,3)-cube and its centrosymmetric image.

Proof. Let $\mathbf{x} = (x_1, x_2, x_3) \in \mathbb{Z}^3$ and let $\mathbf{y} = \pi_\alpha(\mathbf{x}) = -y_1\mathbf{e}_1 + y_2\mathbf{e}_2 \in \Gamma_\alpha$. According to Corollary 13, one has $x_1 = y_1 + H_{\mathcal{P},\alpha}(\mathbf{y})\alpha_1$, $x_2 = y_2 + H_{\mathcal{P},\alpha}(\mathbf{y})\alpha_2$ and $x_3 = H_{\mathcal{P},\alpha}(\mathbf{y})$ with $H_{\mathcal{P},\alpha}(\mathbf{y}) = -\lfloor \frac{v_1y_1 + v_2y_2 + \mu}{w} \rfloor$. One then checks that $0 \leq v_1x_1 + v_2x_2 + v_3x_3 + \mu < w$ if, and only if, $0 \leq v_1y_1 + v_2y_2 + wx_3 + \mu < w$. Hence,

$$\mathbf{x} \in \mathcal{P} \iff 0 \leq v_1y_1 + v_2y_2 + wx_3 + \mu < w,$$

which yields the result since $x_1\mathbf{e}_1 + x_2\mathbf{e}_2 + x_3\mathbf{e}_3 = y_1\mathbf{e}_1 + y_2\mathbf{e}_2 + x_3\alpha$. $\square$

6.1. Construction of non-naive arithmetic discrete planes

I. Debled-Rennesson has given several algorithms [15] computing the geometric representation of rational naive discrete planes, which rely on functionality of such planes. Generalized functionality enables us to adapt these algorithms to any rational arithmetic discrete plane having a functional vector $\alpha \in \mathbb{Z}^3$ satisfying $\alpha_3 = 1$. For the sake of clarity, we have chosen to work out in full details only one algorithm. The other ones can be extended similarly.

Lemma 30 is the key point of the generation algorithm given below (see Algorithm 1). Indeed, instead of constructing the arithmetic discrete plane $\mathcal{P}$ in $\mathbb{Z}^3$ endowed with the basis $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$, we compute the points of $\mathcal{P}$ as the ones of a naive arithmetic discrete plane related to the basis $\{\mathbf{e}_1, \mathbf{e}_2, \alpha\}$ (see Lemma 30). We finally reconstruct $\mathcal{P}$ by to the change of basis $\{\mathbf{e}_1, \mathbf{e}_2, \alpha\}$ to $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$.

A first idea for constructing the geometric representation of $\mathcal{P}$ consists in computing the value $H_{\mathcal{P},\alpha}(\mathbf{y})$ for each $\mathbf{y} \in \Gamma_\alpha$. Nevertheless, the integer division makes this method relatively slow. We thus introduce the map

$$R_{\mathcal{P},\alpha} : \Gamma_\alpha \rightarrow \mathbb{Z}, \mathbf{y} \\ \mapsto y_1\mathbf{e}_1 + y_2\mathbf{e}_2 \mapsto v_1y_1 + v_2y_2 + wH_{\mathcal{P},\alpha}(\mathbf{y}) + \mu.$$

An easy computation gives:

Lemma 31. Let $\mathcal{P} = \mathfrak{P}(\mathbf{v}, \mu, w)$ be a rational arithmetic discrete plane. Let $\alpha \in \mathbb{Z}^3$ be a functional vector of $\mathcal{P}$ with $\alpha_3 = 1$ and let $H_{\mathcal{P},\alpha} : \mathbb{Z}^2 \rightarrow \mathbb{Z}$ be the height function of $\mathfrak{P}$ related to the functional vector α . Then:

$$H_{\mathcal{P},\alpha}(\mathbf{y} + \mathbf{e}_i) - H_{\mathcal{P},\alpha}(\mathbf{y}) = \begin{cases} -\lfloor v_i/w \rfloor & \text{if } R_{\mathcal{P},\alpha}(\mathbf{y}) + \bar{v}_i < w \\ -\lfloor v_i/w \rfloor - 1 & \text{otherwise} \end{cases}$$

$$R_{\mathcal{P},\alpha}(\mathbf{y} + \mathbf{e}_i) - R_{\mathcal{P},\alpha}(\mathbf{y}) = \begin{cases} \bar{v}_i & \text{if } R_{\mathcal{P},\alpha}(\mathbf{y}) + \bar{v}_i < w \\ \bar{v}_i - w & \text{otherwise} \end{cases}$$

where $\bar{v}_i = v_i \bmod w$.

Hence, given the rectangle $\llbracket m_1, m'_1 \rrbracket \times \llbracket m_2, m'_2 \rrbracket$, one constructs the pre-images of the points $\mathbf{y} = y_1\mathbf{e}_1 + y_2\mathbf{e}_2 \in \Gamma_\alpha$, with $(y_1, y_2) \in \llbracket m_1, m'_1 \rrbracket \times \llbracket m_2, m'_2 \rrbracket$ by first computing the height $H(\mathbf{y}_0)$ of the point $\mathbf{y}_0 = m_1\mathbf{e}_1 + m_2\mathbf{e}_2 \in \Gamma_\alpha$, and, then, step by step, and thanks to Lemma 31, by calculating the height $H(\mathbf{y})$ of each point $\mathbf{y} = y_1\mathbf{e}_1 + y_2\mathbf{e}_2 \in \Gamma_\alpha$, where $(y_1, y_2) \in \llbracket m_1, m'_1 \rrbracket \times \llbracket m_2, m'_2 \rrbracket$ (see Algorithm 1).

Algorithm 1: An algorithm constructing the arithmetic plane $\mathfrak{P}(\mathbf{v}, \mu, w)$ satisfying $\mathbf{v} \in \mathbb{N}^3$, $\mu \in \mathbb{Z}$ and $w = \alpha_1v_1 + \alpha_2v_2 + v_3$ with $(\alpha_1, \alpha_2) \in \mathbb{Z}^2$.

Input: a rational arithmetic discrete plane $\mathcal{P} = \mathfrak{P}(\mathbf{v}, \mu, w)$ with $w \geq \{v_1, v_2\}$;
a functional vector $\alpha \in \mathbb{Z}^3$ $\mathcal{P}$ with $\alpha_3 = 1$;
the ranges $[m_0, m_1] \times [n_0, n_1]$;

Output: the points of $\mathcal{P} = \mathfrak{P}(\mathbf{v}, \mu, w)$ with x -coordinates in $[m_0, m_1]$ and y -coordinates in $[n_0, n_1]$ in the basis $\{\mathbf{e}_1, \mathbf{e}_2, \alpha\}$ of $\mathbb{Z}^3$.

```

 $\mathbf{z}_{ini} = -\left\lfloor \frac{v_1m_0 + v_2n_0 + \mu}{w} \right\rfloor; \quad \mathbf{z} = \mathbf{z}_{ini};$ 
 $\mathbf{x}_{ini} = m_0 + \alpha_1\mathbf{z}; \quad \mathbf{x} = \mathbf{x}_{ini};$ 
 $\mathbf{y}_{ini} = n_0 + \alpha_2\mathbf{z}; \quad \mathbf{y} = \mathbf{y}_{ini};$ 
 $\mathbf{r}_{ini} = v_1m_0 + v_2n_0 + w\mathbf{z}_{ini} + \mu; \quad \mathbf{r} = \mathbf{r}_{ini};$ 
begin
  for  $n \in \{n_0, \dots, n_1\}$  do
    for  $m \in \{m_0, \dots, m_1\}$  do
      Build the voxel  $(x, y, z)$ ;
       $\mathbf{r} = \mathbf{r} + \bar{v}_1$ ;  $\mathbf{x} = \mathbf{x} + 1$ ;
      if  $\mathbf{r} < w$  then
         $\mathbf{z} = \mathbf{z} - \lfloor \frac{v_1}{w} \rfloor$ ;  $\mathbf{x} = \mathbf{x} - \alpha_1 \lfloor \frac{v_1}{w} \rfloor$ ;  $\mathbf{y} = \mathbf{y} - \alpha_2 \lfloor \frac{v_1}{w} \rfloor$ ;
      else
         $\mathbf{z} = \mathbf{z} - (\lfloor \frac{v_1}{w} \rfloor + 1)$ ;  $\mathbf{x} = \mathbf{x} - \alpha_1 (\lfloor \frac{v_1}{w} \rfloor + 1)$ ;
         $\mathbf{y} = \mathbf{y} - \alpha_2 (\lfloor \frac{v_1}{w} \rfloor + 1)$ ;
      end
    end
     $\mathbf{r}_{ini} = \mathbf{r}_{ini} + \bar{v}_2$ ;  $\mathbf{y}_{ini} = \mathbf{y}_{ini} + 1$ ;
    if  $\mathbf{r}_{ini} < w$  then
       $\mathbf{z}_{ini} = \mathbf{z}_{ini} - \lfloor \frac{v_2}{w} \rfloor$ ;  $\mathbf{x}_{ini} = \mathbf{x}_{ini} - \alpha_1 \lfloor \frac{v_2}{w} \rfloor$ ;  $\mathbf{y}_{ini} = \mathbf{y}_{ini} - \alpha_2 \lfloor \frac{v_2}{w} \rfloor$ ;
    else
       $\mathbf{z}_{ini} = \mathbf{z}_{ini} - (\lfloor \frac{v_2}{w} \rfloor + 1)$ ;
       $\mathbf{x}_{ini} = \mathbf{x}_{ini} - \alpha_1 (\lfloor \frac{v_2}{w} \rfloor + 1)$ ;
       $\mathbf{y}_{ini} = \mathbf{y}_{ini} - \alpha_2 (\lfloor \frac{v_2}{w} \rfloor + 1)$ ;
    end
    end
     $\mathbf{x} = \mathbf{x}_{ini}$ ;  $\mathbf{y} = \mathbf{y}_{ini}$ ;  $\mathbf{z} = \mathbf{z}_{ini}$ ;  $\mathbf{r} = \mathbf{r}_{ini}$ ;
  end
end
```

6.2. Arithmetic discrete planes as combinatorial manifolds

Due to the surface nature of Euclidean planes, it becomes natural to try to endow rational arithmetic discrete planes with a relevant notion of discrete surface. The notion of discrete surface has been widely studied during the last 25 years and several approaches have been chosen for the definition of such objects. For instance, in [25], D.G. Morgenthaler and A. Rosenfeld define a discrete surface as a 0-connected subset of $\mathbb{Z}^3$ satisfying some local conditions of 0-adjacency and 2-separatingness. Nevertheless, this definition is not relevant for rational arithmetic discrete planes. For instance, among the 40 tricubes occurring in the naive arithmetic discrete planes with positive normal vector $\mathbf{v} \in \mathbb{Z}^3$, such that $0 \leq v_1 \leq v_2 \leq v_3$, only 7 of them occur in a discrete surface in the sense of D.G. Morgenthaler and A. Rosenfeld [24].

In [17,21,22], the authors have shown that an appropriate way to provide rational arithmetic discrete planes with a discrete surface structure is to consider two-dimensional combinatorial manifolds. For instance, J. Françon showed in [17] that the 2-adjacency graph of a rational standard arithmetic discrete plane has a natural underlying structure of two-dimensional combinatorial manifold. In [21], Y. Kenmochi

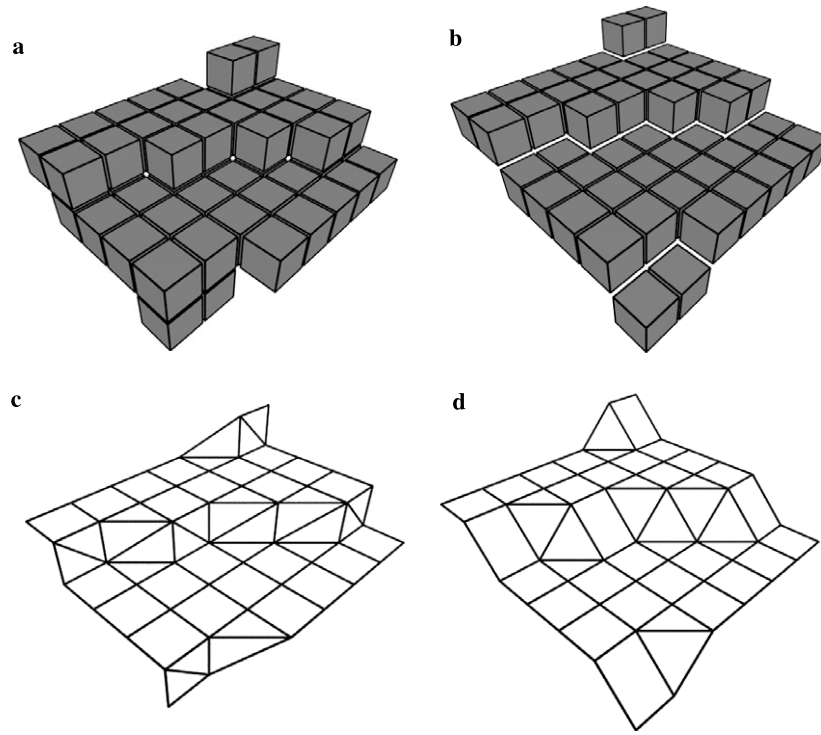


Fig. 6. From a thick arithmetic discrete plane to a two-dimensional combinatorial manifold. (a) The arithmetic discrete plane $0 \leq 6x_1 + 11x_2 + 31x_3 < 42$ represented by voxels. (b) The arithmetic discrete plane $0 \leq 6x_1 + 11x_2 + 42x_3 < 42$ represented by voxels. (c) The two-dimensional combinatorial manifold $\mathcal{M}((6, 11, 31), 0, 42)$. The two-dimensional combinatorial manifold $\mathcal{M}((6, 11, 42), 0, 42)$.

and A. Imiya, thanks to a similar approach, proved that one can provide any rational naive arithmetic discrete plane $\mathcal{P}$ with two different structures of two-dimensional combinatorial manifolds, depending on the 0 -adjacency and the 1 -adjacency graph of $\mathcal{P}$, respectively (see Fig. 6).

In the present section, we show that these latter approaches can be extended in a quite natural way to any rational arithmetic discrete plane $\mathcal{P}$ with normal vector $\mathbf{v} \in \mathbb{Z}^3$ under the assumption $w - v_3 \in \gcd(v_1, v_2)\mathbb{Z}$ and $w \geq \max\{v_1, v_2\}$. We have chosen to consider 1 -adjacency only, 0 -adjacency being handled in a similar way.

Let us first recall basic notions concerning combinatorial manifolds. Let G be a graph and let F be a set of elementary cycles of G , that is, circular permutations $(V_1, V_2, \dots, V_k)$, with $k > 2$, of vertices of G such that $V_i \neq V_j$ if $i \neq j$ and, for $i \in \mathbb{N}$, any pair $\{V_i, V_{i+1}\}$ is an edge of G (the indices are considered modulo k). Such a cycle is called a *face*. Two faces are called *adjacent* if they share an edge. An *umbrella* at a vertex V of G is a circular permutation $(F_1, F_2, \dots, F_k)$ of faces of F , with $k > 1$, all sharing a vertex V , and such that F_i and F_{i+1} are adjacent for $i \in \mathbb{N}$ (indices are considered modulo k).

Definition 32. Let G be a graph and let F be a set of faces of G . The pair (G, F) is called a *two-dimensional combinatorial manifold (without boundary)* if the following holds:

- (i) any edge of G belongs to exactly two faces of G ,
- (ii) any vertex of G belongs to exactly one umbrella of G .

Since rational naive arithmetic discrete planes can be provided with a two-dimensional combinatorial manifold structure [21], then, thanks to Lemma 30, there is a quite natural way to provide any rational arithmetic discrete plane $\mathfrak{P}(\mathbf{v}, \mu, w)$ having a functional vector $\alpha \in \mathbb{Z}^3$ satisfying $\alpha_3 = 1$. According to Lemma 30, the arithmetic discrete plane $\mathfrak{P}(\mathbf{v}, \mu, w)$ is naive relatively to the basis $\{\text{vect}e_1, \mathbf{e}_2, \alpha\}$ and can be provided with a two-dimensional combinatorial manifold structure. Let us introduce the bijection map

$$\begin{aligned} \phi : \mathfrak{P}(\mathbf{v}, \mu, w) &\rightarrow \mathfrak{P}((v_1, v_2, w), 0, w) \\ (x_1, x_2, x_3) &\mapsto (x_1 - \alpha_1 x_3, x_2 - \alpha_2 x_3, x_3). \end{aligned}$$

By ϕ , the arithmetic discrete planes $\mathfrak{P}(\mathbf{v}, \mu, w)$ and $\mathfrak{P}((v_1, v_2, w), 0, w)$ are identified. We thus define a two-dimensional combinatorial manifold $\mathfrak{M}(\mathbf{v}, \mu, w)$ over $\mathfrak{P}(\mathbf{v}, \mu, w)$ as follows: the vertices of $\mathfrak{M}(\mathbf{v}, \mu, w)$ are the points of $\mathfrak{P}(\mathbf{v}, \mu, w)$ and for all $(\mathbf{x}, \mathbf{x}') \in \mathfrak{P}(\mathbf{v}, \mu, w)^2$, $\{\mathbf{x}, \mathbf{y}\}$ is an edge if, and only if, $\phi(\mathbf{x})$ and $\phi(\mathbf{y})$ are 1 -adjacent in $\mathfrak{P}((v_1, v_2, w), \mu, w)$. Finally, the faces of $\mathfrak{M}(\mathbf{v}, \mu, w)$ are the image by ϕ of the faces of the naive arithmetic discrete plane $\mathfrak{P}((v_1, v_2, w), \mu, w)$ (see Fig. 6).

7. Conclusion

The aim of the present work was to introduce suitable tools in order to generalize well-known properties of naive arithmetic discrete planes. We have exhibited a *generalized*

functionality for arithmetic discrete planes $\mathcal{P} = \mathfrak{P}(\nu, \mu, \omega)$ by introducing a suitable projection direction α . We have proved that, as soon as $|\langle \alpha, \nu \rangle| = w$ and $\gcd(\alpha_1, \alpha_2, \alpha_3) = 1$, there is a one-to-one correspondence between $\mathcal{P}$ and a two-dimensional lattice Γ_α ; we also have described the latter lattice. We then have focused on the class of arithmetic discrete planes for which there exists α with $\alpha_3 = 1$: we first have investigated plane partitions and local configurations; indeed we have extended the well-known result on the number of (m, n) -configurations in a naive plane (there are at most mn such configurations), and we have considered their statistical behaviour; we finally have proved that arithmetic discrete planes with $\alpha_3 = 1$ and $w \geq \max\{v_1, v_2\}$ can be considered as naive in a suitable basis of $\mathbb{R}^3$; we have deduced a generation algorithm and shown how to endow them with a structure of two-dimensional manifold.

The approach developed in Section 5 is classical in symbolic dynamics and, in particular, in the study of sturmian and multidimensional sturmian words [26]. As an analogous example of application of symbolic dynamics in discrete geometry see also [4].

The results of the present paper offer new perspectives for further investigation of general properties of arithmetic discrete planes of any thickness. In particular, we plan to use them to generate arbitrarily large parts of discrete planes via symbolic substitutions following [2], to recover the corresponding Farey tessellation as well as the symmetry properties of (m, n) -cubes of a discrete plane [32], and finally as a new approach to the recognition problem of discrete planes [19,18,33].

References

- [1] E. Andres, R. Acharya, C. Sibata, The discrete analytical hyperplanes, *Graph. Models Image Process.* 59 (5) (1997) 302–309.
- [2] P. Arnoux, V. Berthé, A. Siegel, Two-dimensional iterated morphisms and discrete planes, *Theoret. Comput. Sci.* 319 (2004) 145–176.
- [3] R.P. Barneva, V.E. Brimkov, P. Nehlig, Thin discrete triangular meshes, *Theoret. Comput. Sci.* 246 (1–2) (2000) 73–105.
- [4] V. Berthé, B. Nouvel, Density of symbols in discretized rotation configurations, in: *Words 2005 – 5th International Conference on Words*, vol. 36 of Publications du LaCIM, Université du Québec à Montréal, Université du Québec à Montréal, September 2005, pp. 163–173.
- [5] V. Berthé, L. Vuillon, Tilings and rotations on the torus: a two-dimensional generalization of sturmian sequences, *Discrete Math.* 223 (2000) 27–53.
- [6] V. Berthé, L. Vuillon, Palindromes and two-dimensional sturmian sequences, *J. Autom. Lang. Comb.* 6 (2) (2001) 121–138.
- [BFJ05] V. Berthé, C. Fiorio, D. Jamet, Generalized functionality for arithmetic discrete planes, in: *DGCI 2005, 12th International Conference, Lecture Notes in Computer Science*, vol. 3429, Springer, 2005, pp. 276–286.
- [8] V.E. Brimkov, E. Andres, R.P. Barneva, Object discretizations in higher dimensions, *Pattern Recogn. Lett.* 23 (6) (2002) 623–636.
- [9] V.E. Brimkov, R.P. Barneva, Graceful planes and thin tunnel-free meshes, in: *DGCI, 8th International Conference, Lecture Notes in Computer Science*, vol. 1568, 1999, pp. 53–64.
- [10] V.E. Brimkov, R.P. Barneva, P. Nehlig, Minimally thin discrete triangulations, in: A. Kaufman, R. Yagel, M. Chen (Eds.), *Volume Graphics*, Springer, 2000, pp. 51–70 (Chapter 3).
- [11] V.E. Brimkov, R.P. Barneva, Graceful planes and lines, *Theoret. Comput. Sci.* 283 (2002) 151–170.
- [12] V.E. Brimkov, R. Klette, Curves, hypersurfaces, and good pairs of adjacency relations, in: *Lecture Notes in Computer Science*, vol. 3322, Springer, 2004, pp. 270–284.
- [13] M.A. Jacob-Da Col, About local configurations in arithmetic planes, *Theoret. Comput. Sci.* 283 (2002) 183–201.
- [14] I. Debled-Renesson, J.-L. Rémy, J. Rouyer-Degli, Segmentation of discrete curves into fuzzy segments, in: *9th International Workshop on Combinatorial Image Analysis – Electronic Notes in Discrete Mathematics*, vol. 12, Palermo, Italy, Elsevier, May 14–16 2003, p. 12.
- [15] I. Debled-Renesson, Reconnaissance des Droites et Plans Discrets. Thèse de doctorat, Université Louis Pasteur, Strasbourg, France, 1995.
- [16] I. Debled-Renesson, J.-P. Reveillès, A New Approach to Digital Planes, in: *Vision geometry III, Proc. SPIE*, vol. 2356, Boston, USA, 1994.
- [17] J. Françon, Sur la topologie d'un plan arithmétique, *Theoret. Comput. Sci.* 156 (1–2) (1996) 159–176.
- [18] J. Françon, L. Papier, Polyhedrization of the boundary of a voxel object, in: *DGCI, 8th International Conference, Lecture Notes in Computer Science*, vol. 1568, 1999, pp. 425–434.
- [19] J. Françon, J.-M. Schramm, M. Tajine, Recognizing arithmetic straight lines and planes, in: *DGCI, 6th International Workshop, Lecture Notes Computer Science*, 1996, pp. 141–150.
- [20] Y. Gérard, Local Configurations of digital hyperplanes, in: *DGCI, 8th International Conference*, vol. 1568, 1999, pp. 65–75.
- [21] Y. Kenmochi, A. Imiya, Naive planes and discrete combinatorial surfaces, in: *Proceedings of the 9th International Conference on Discrete Geometry for Computer Imagery, Springer-Verlag*, 2000, pp. 249–261.
- [22] Y. Kenmochi and A. Imiya, Combinatorial topologies for discrete planes, in: *DGCI, 11th International Conference, DGCI 2003, Lecture Notes in Computer Science*, vol. 2886, Springer-Verlag, 2003, pp. 144–153.
- [23] L. Kuipers, H. Niederreiter, Uniform distribution of sequences. (Ravnomernoe raspredelenie posledovatel'nostej), S.M. Ermakov (Ed), Moskva: Nauka. Glavnaya Redaktsiya Fiziko-Matematicheskoy Literatury (Transl. from the English. Transl.), 1985, 408 p.
- [24] R. Malgouyres, G. Bertrand, Complete local characterization of strong 26-surfaces: continuous analogs for strong 26-surfaces, *International Journal on Pattern Recognition and Artificial Intelligence (IJPRAI)* 13 (4) (1999) 465–484.
- [25] D.G. Morgenthaler, A. Rosenfeld, Surfaces in three-dimensional digital images., *Informat. Control* 51 (1981) 227–247.
- [26] N. Pytheas Fogg, Substitutions in dynamics, arithmetics and combinatorics, in: V. Berthé, S. Ferenczi, C. Mauduit, A. Siegel (Eds.), *Lecture Notes in Mathematics*, vol. 1794, Springer Verlag, 2002.
- [27] J.-P. Reveillès, Calcul en Nombres Entiers et Algorithmique. Thèse d'état, Université Louis Pasteur, Strasbourg, France, 1991.
- [28] J.-P. Reveillès, Combinatorial pieces in digital lines and planes, in: *Vision geometry IV, Proc. SPIE*, 2573, vol. 2573, San Diego, CA, 1995, pp. 23–24.
- [29] G. Rote, Sequences with subword complexity $2n$, *J. Number Theory* 46 (1994) 196–213.
- [30] J.-M. Schramm, Coplanar tricubes, in: *DGCI, 7th International Workshop, Lecture Notes in Computer Science*, vol. 1347, 1997, pp. 87–98.

- [31] J. Vittone, J.-M. Chassery, Coexistence of tricubes in digital naive plane, in: DGCI, 7th International Workshop, Lecture Notes in Computer Science, vol. 1347, 1997, pp. 99–110.
- [32] J. Vittone, J.-M. Chassery, (n,m) -cubes and Farey Nets for Naive Planes Understanding, in: DGCI, 8th International Conference, Lecture Notes in Computer Science, vol. 1568, Springer-Verlag, 1999, pp. 76–87.
- [33] J. Vittone, J.-M. Chassery, Recognition of digital naive planes and polyhedrization, in: DGCI, 9th International Conference, Lecture Notes in Computer Science, vol. 1953, Springer-Verlag, 2000, pp. 296–307.
- [34] L. Vuillon, Local configurations in a discrete plane, *Bull. Belgian Math. Soc.* 6 (1999) 625–636.