



HAL
open science

Pyramides irrégulières descendantes pour la segmentation de grandes images histologiques

Romain Goffe

► **To cite this version:**

Romain Goffe. Pyramides irrégulières descendantes pour la segmentation de grandes images histologiques. Algorithme et structure de données [cs.DS]. Université de Poitiers, 2011. Français. NNT : . tel-00657621

HAL Id: tel-00657621

<https://theses.hal.science/tel-00657621>

Submitted on 7 Jan 2012

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.

Thèse

pour l'obtention du grade de
DOCTEUR DE L'UNIVERSITÉ DE POITIERS
Faculté des Sciences Fondamentales et Appliquées
(Diplôme National – Arrêté du 7 Août 2006)

École Doctorale : **Sciences et Ingénierie pour l'Information**
Secteur de Recherche : **Informatique et Applications**

présentée par :
Romain GOFFE

Pyramides irrégulières descendantes pour la segmentation de grandes images histologiques

Directeurs de thèse :
Luc BRUN
Guillaume DAMIAND

Soutenue le 14 Septembre 2011 devant le jury composé de :

Pascal LIENHARDT, Professeur, Université de Poitiers, XLim/SIC Président du Jury
Achille BRAQUELAIRE, Professeur, Université de Bordeaux I, LaBRI Rapporteur
Christophe FIORIO, Professeur, Université Montpellier 2, LIRMM Rapporteur
Luc BRUN, Professeur, Université de Caen, GREYC Directeur de Thèse
Guillaume DAMIAND, Chargé de Recherche, CNRS, LIRIS Directeur de Thèse

Remerciements

Je tiens avant-tout à remercier Luc Brun et Guillaume Damiand pour leur patience, leur soutien constant, leur aide précieuse et l'encadrement irréprochable durant ces années de thèse. Encore aujourd'hui, j'ai peine à réaliser la chance et la valeur de ce qu'ils m'ont transmis. Avec un peu de recul, les cartes combinatoires m'auront clairement demandé un certain nombre d'arrachages de cheveux, de nuits blanches et de litres de café mais le bilan que j'en tire est plus que largement positif. Le sujet s'est révélé exceptionnellement formateur, enrichissant et *fun*.

Merci à mes rapporteurs Achille Braquelaire et Christophe Fiorio, aux membres du jury et mes autres relecteurs d'avoir pris le temps de lire, corriger et évaluer mon travail, réalisé au sein de différents laboratoires. J'ai en effet eu l'opportunité de travailler avec l'équipe IG du XLIM-SIC à Poitiers, l'équipe M2Disco du LIRIS à Lyon et l'équipe Image du GREYC à Caen. En dehors des multiples déménagements nécessaires, cela aura été l'occasion de rencontrer de nombreuses personnes qui m'ont toutes beaucoup apporté. C'est donc très sincèrement que je remercie tous les membres de ces équipes, pour leur accueil chaleureux et leurs conseils avisés.

Merci à tous mes collègues thésards, post-doc et ater pour tous ces extraordinaires souvenirs de soirées, de musique, de discussions aussi techniques que passionnées, de parties de belote, de poker, de go ou d'échecs. Prenant le risque d'en oublier certains, merci aux Poitevins Alexandre, Sylvain, Marc, Guillaume, Olivier, Boris, Thomas, Romuald ; aux Lyonnais Guillaume, Romain, Stéphane, Camille, Alain ; et aux Caennais Vincent, Guillaume, Pierre, Xavier, Moncef, Benoît, Maher, Matthieu, François, Vinh, François-Xavier, Haz-Hédine, Gaëlle. Je vous dois à tous d'avoir apprécié vos villes respectives, malgré ma difficulté notoire d'adaptation à leurs climats.

Mention particulière à Alex, collègue, colocataire et ami qui m'aura supporté, soutenu et apporté dans tant de domaines, allant des cartes combinatoires aux conseils de cuisine. Merci également à Natacha qui m'aura accompagné courageusement sur une jolie partie du trajet.

Rapide aparté pour ma fidèle guitare qui m'a permis de concilier musique et open-source de façon inattendue mais incroyablement enrichissante à travers le *songbook* ; projet aussi chronophage que complémentaire par l'apprentissage et l'utilisation de nouveaux outils, composants et langages informatiques.

Je n'oublie pas non plus tous les amis d'école et d'avant, *On Fire & Co*, dont la valse des mariages, étrennée depuis peu, ne fait que commencer. La liste étant longue, je la limite à ceux avec qui tout a commencé et continue encore aujourd'hui : Antonin, Simon, Touf, Stitch, Roni, Ratz, Shark, Stéfoune, Loubidou mais aussi Fifouille, Pampa, Krakoy, Antoine et Ludo. Sous ces étranges dénominations se reconnaîtront les personnalités qui m'ont construit comme je suis.

Enfin, un merci inestimable à Florentine, Godefroy, Dorian, Grégoire et mes parents qui ont su traverser bien des épreuves, chacun à leur manière, mais toujours en allant de l'avant.

Romain

Table des matières

Introduction	9
1 Représentation d'images	17
1.1 Préliminaires	18
1.1.1 Image	18
1.1.2 Notions de base de topologie	19
1.1.3 Complexes cellulaires abstraits	21
1.1.4 Régions et partitions	25
1.2 Structures géométriques	26
1.2.1 Tableaux d'étiquettes	27
1.2.2 Chemins frontières	28
1.2.3 Frontières interpixel	30
1.3 Structures relationnelles	31
1.3.1 Graphes d'adjacence de régions	32
1.3.2 Graphes duaux	34
1.3.3 Cartes combinatoires	36
1.3.4 Cartes topologiques	42
1.4 Structures hiérarchiques	47
1.4.1 Pyramides régulières	49
1.4.2 Pyramides de graphes simples	52
1.4.3 Processus de décimation	53
1.4.4 Pyramides de graphes duaux	56
1.4.5 Pyramides combinatoires	58
1.4.6 Pyramides généralisées	61
1.5 Conclusion	63

2	Cartes topologiques tuilées	65
2.1	Introduction	66
2.2	Tuiles topologiques	67
2.2.1	Définition du modèle de tuile topologique	68
2.2.2	Équivalences au bord des tuiles	69
2.2.3	Relations d'adjacence entre tuiles	71
2.3	Cartes tuilées	75
2.3.1	Abstraction des bords fictifs	77
2.3.2	Définition d'opérateurs	79
2.3.3	Définition du modèle de cartes tuilées	82
2.4	Construction d'une première partition	83
2.4.1	Extraction incrémentale de la carte topologique d'une tuile	84
2.4.2	Connexion de tuiles adjacentes	86
2.4.3	Itération sur toutes les tuiles	89
2.5	Raffinement d'une région par un processus de décomposi- tion/fusion	90
2.5.1	Première étape : décomposition	91
2.5.2	Deuxième étape : fusion	94
2.5.3	Extension du processus de décomposition/fusion aux cartes tuilées	96
2.6	Structures de données et implantation	100
2.6.1	Structures de données	100
2.6.2	Déchargement d'une tuile topologique	102
2.6.3	Chargement d'une tuile topologique	104
2.7	Conclusion	104
 3	 Pyramides descendantes tuilées	 107
3.1	Introduction	108
3.2	Définition du modèle de pyramide descendante tuilée	109
3.2.1	Pyramides d'images	110
3.2.2	Pyramides de tuiles	116
3.2.3	Pyramides descendantes tuilées	120
3.3	Représentation des relations hiérarchiques	122
3.3.1	Projection des (0,1)-chemins	123
3.3.2	Relations <i>prédécesseur/successeur</i> dans une carte tuilée .	127
3.4	Utilisation pratique des relations hiérarchiques	130

3.4.1	Identification des arêtes filles	131
3.4.2	Identification des régions filles	133
3.5	Construction d'une pyramide descendante tuilée	136
3.5.1	Projection des frontières	136
3.5.2	Extraction des successeurs d'une tuile	138
3.5.3	Extraction d'une pyramide	142
3.6	Conclusion	143
4	Segmentation et expérimentations	145
4.1	Introduction	146
4.2	Segmentation de grandes images	146
4.2.1	Préliminaires	147
4.2.2	Approches par blocs	148
4.2.3	Approches hiérarchiques	150
4.2.4	Approches par architecture matérielle dédiée	152
4.2.5	Bilan	153
4.3	Expérimentations sur le modèle	153
4.3.1	Préliminaires	154
4.3.2	Segmentation par blocs avec le modèle de carte tuilée	157
4.3.3	Segmentation hiérarchique avec le modèle de pyramide descendante	163
4.4	Utilisation du modèle pour la segmentation d'images histologiques	165
4.4.1	Processus de segmentation	167
4.4.2	Extraction des mitoses	170
4.5	Conclusion	171
	Conclusion et Perspectives	173
	Références Bibliographiques	177
	Liste des figures	187
	Liste des algorithmes	191
	Index	193

Introduction

Le diagnostic précoce du cancer du sein est établi à partir d'un prélèvement effectué sur le patient puis déposé sur une lame de verre. Une étape de coloration des cellules (par immuno-histochimie ou histochimie) permet de mettre en évidence les phases de mitoses représentatives d'une tumeur. Le prélèvement est ensuite observé au microscope par le pathologiste qui va effectuer une quantification des cellules colorées pour établir le diagnostic.

Les récentes améliorations techniques des scanners permettent d'obtenir une précision de l'ordre du dixième de micromètre ($54\,000\text{dpi}$ ¹) sur l'ensemble de la lame. En contrepartie, la masse de données que représente l'image scannée représente un véritable défi technologique pour le domaine du traitement d'images. Cette problématique a donné lieu à la mise en place du projet FoGrImMi (FOuille de GRandes IMages Microscopiques), financé par l'Agence Nationale de la Recherche (ANR).

Cette thèse s'inscrit dans le cadre du traitement d'images de lames entières en microscopie afin d'automatiser et de proposer une aide au diagnostic précoce du cancer du sein.

Au cours de cette introduction, nous commençons par présenter le contexte clinique de l'acquisition des images en retraçant brièvement l'historique qui conduit aujourd'hui à l'utilisation des scanners de lames entières. Nous nous intéressons ensuite aux grandes images produites. Nous détaillons leurs caractéristiques techniques, la nomenclature des objets qu'elles contiennent ainsi que le processus d'analyse des pathologistes. Cette étude permet de mettre en avant différents points clés que nous approfondissons dans la suite de ce manuscrit.

1. Le point par pouce, noté **ppp** (*dpi* en anglais) définit la précision d'un scanner et correspond au nombre de points d'information sur une ligne d'un pouce (2,54 cm). Une précision de $54\,000\text{dpi}$ correspond donc à une taille de point de $25\,400\ \mu\text{m}/54\,000 = 0,47\ \mu\text{m}$.

Contexte médical

L'utilisation d'un microscope comme outil d'analyse visuelle des pathologistes remonte au XVII^e siècle. Malgré les avancées effectuées en optique, l'utilisation d'un microscope entraîne plusieurs inconvénients lors de l'analyse d'une image comme par exemple les ruptures liées aux changements d'objectifs.

Depuis l'arrivée du numérique, l'analyse d'un prélèvement ne s'effectue plus directement sur la lame de verre à travers l'objectif mais à partir d'une image numérique acquise par le microscope et appelée *lame virtuelle*.

Les avantages apportés par ce contexte de microscopie virtuelle sont multiples. En effet, par rapport à une lame de verre physique, sa représentation numérique sous forme d'image facilite grandement sa réplique, son stockage et son partage. Le format numérique apporte d'autres avantages comme par exemple, la conservation des couleurs qui n'est plus soumise aux altérations du temps ou encore l'exploration assistée par logiciel dont les fonctions de grossissement permettent d'éviter les changements brusques de résolution lors d'un changement d'objectif précédemment évoqués.

L'analyse des pathologistes demande un grossissement très important permettant l'identification de structures au niveau cellulaire. D'un point de vue technique, la résolution de l'image doit être suffisamment importante pour distinguer les cellules. Seule la dernière génération de scanners de lames entières permet d'obtenir ce niveau de détails. En contrepartie, l'acquisition de ces images produit un volume de données numériques considérable. Cette contrainte technique a restreint la numérisation des lames de verre à des zones partielles de la lame pour garder un volume de données exploitable.

Or, la répartition des structures à observer étant généralement hétérogène, la sélection arbitraire d'une zone partielle d'observation peut fausser le diagnostic puisque certaines zones, potentiellement révélatrices d'une tumeur, sont ignorées. Afin de pallier ce problème, différentes entreprises ont mis au point des microscopes capables de numériser entièrement la lame. Récemment, les derniers modèles de scanner à haute résolution ont permis d'obtenir ces images de lame entière (ILE) (*whole slide imaging (WSI)* en anglais). Deux exemples de ces scanners haute résolution sont illustrés dans la Figure 1.

L'analyse des images produites par les scanners ILE implique généralement différentes disciplines de la médecine. Par exemple, la *cytologie* consiste à étudier les cellules à leur niveau de détail le plus fin tandis que l'*histologie* s'intéresse à l'agencement de ces cellules en tissus. Nous nous intéressons dans cette thèse uniquement aux images histologiques qui représentent les plus grands défis pour l'analyse d'images.



FIGURE 1 – Exemples de scanners de lames entières (54 000 dpi). (a) Aperio Technologies : ScanScope CS ; (b) Hamamatsu : NanoZoomer.

Grandes images

Les scanners ILE ont une résolution suffisante pour fournir une image permettant l'observation des structures au niveau cellulaire pour l'ensemble de la lame de verre.

Le facteur d'échelle entre la taille réelle du prélèvement et l'observabilité des cellules qui le composent est d'environ 40 000 et un prélèvement moyen correspond à 4 cm^2 de tissu. Par conséquent, hors compression, les scanners ILE produisent des images atteignant 4,5Go (grossissement $\times 20\,000$), voire 18Go (grossissement $\times 40\,000$). Le volume de données numérisées est ensuite stocké sous la forme d'un unique fichier contenant l'image à pleine résolution ainsi qu'une séquence d'images codant différentes résolutions de l'image de base. Étant donné que le volume de données associé aux images produites est très important, nous qualifions ces images de *grandes images*.

Composition d'une image histologique

Dans le cadre du diagnostic du cancer du sein, une image histologique est composée d'un grand nombre d'objets différents. Un schéma récapitulatif présentant les différents éléments et leurs positionnements les uns par rapport aux autres est proposé par la Figure 2 (voir également Figure 3). Ce schéma classe les objets suivant cinq tranches de résolutions auxquelles ils peuvent être observés. Par exemple, avec un grossissement $\times 10$ (résolution 1), il est uniquement possible de distinguer le tissu du fond de l'image tandis qu'avec un grossissement $\times 40\,000$ (résolution 5), il est possible de différencier les différentes phases de la division des cellules (mitoses).

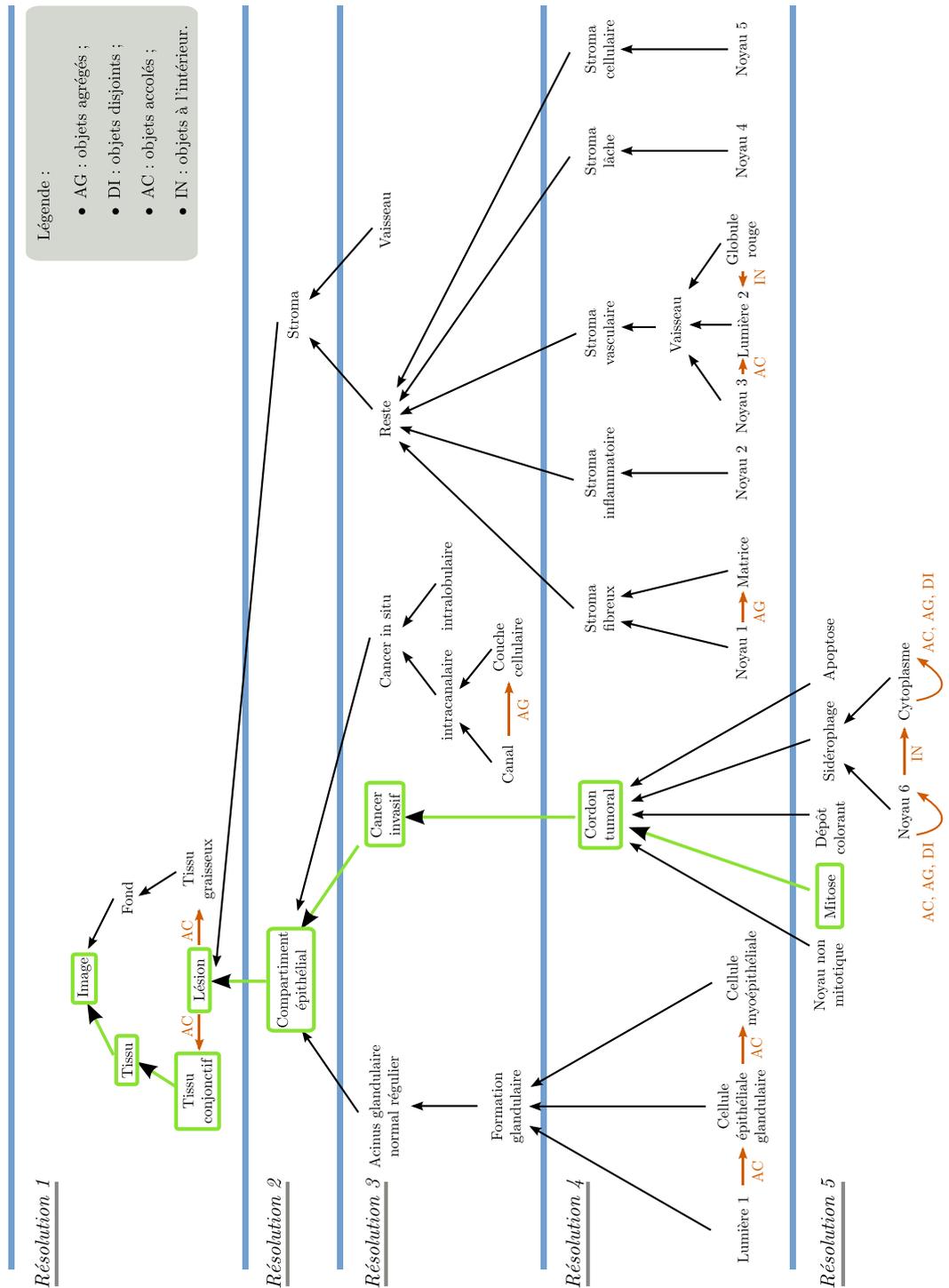


FIGURE 2 – L'arbre des objets du domaine d'analyse histologique.

Analyse

Tous les éléments qui composent une image histologique ne sont pas forcément pertinents pour établir le diagnostic. L'objectif est donc de restreindre l'analyse de l'image à des zones d'intérêt. Dans la Figure 2, nous avons mis en évidence les zones d'intérêt pour les pathologistes en encadrant pour chaque résolution les objets du domaine servant à établir le diagnostic. De façon simplifiée, le processus de sélection s'effectue de façon descendante (de la résolution la plus faible à la plus élevée) suivant les étapes décrites ci-dessous :

- résolution 1 : différencier le tissu du fond de l'image ;
- résolution 2 : extraire les limites de la lésion ;
- résolution 3 : éliminer les acini glandulaires réguliers ;
- résolution 4 : extraire les figures de mitoses ;
- résolution 5 : différencier les figures de mitoses.

La Figure 3 propose un exemple concret d'une image d'histologie, chaque vignette mettant en évidence les objets observables à une résolution donnée. Dans cet exemple, nous illustrons les objets du domaine correspondant au schéma simplifié de l'analyse des pathologistes présenté dans la Figure 2. Ainsi, chaque ligne correspond à une résolution plus élevée que la ligne précédente. Une ligne contient deux vignettes illustrant une zone d'intérêt (colonne de droite) à explorer plus en détail et une zone non pertinente (colonne de gauche).

Notre démarche de recherche

Axes de recherche

Nous pouvons dégager trois aspects fondamentaux du processus d'analyse des grandes images histologiques effectué par les pathologistes.

Premièrement, la caractérisation des objets du domaine repose sur des propriétés topologiques comme les relations d'adjacence et d'imbrication. Ces relations sont respectivement représentées par les sigles AC et IN dans la Figure 2.

Deuxièmement, les deux étapes de quantification et de dénombrement des mitoses s'appuient sur des critères colorimétriques et géométriques. En effet, les cellules en phase de division cellulaire sont colorées à l'aide de marqueurs histochimiques qui permettent de les mettre en évidence. La différenciation des différentes phases de mitoses s'effectue par une analyse géométrique de leur forme.

Troisièmement, l'analyse est réalisée suivant une approche hiérarchique descendante allant de la plus faible résolution à la plus élevée. À chaque

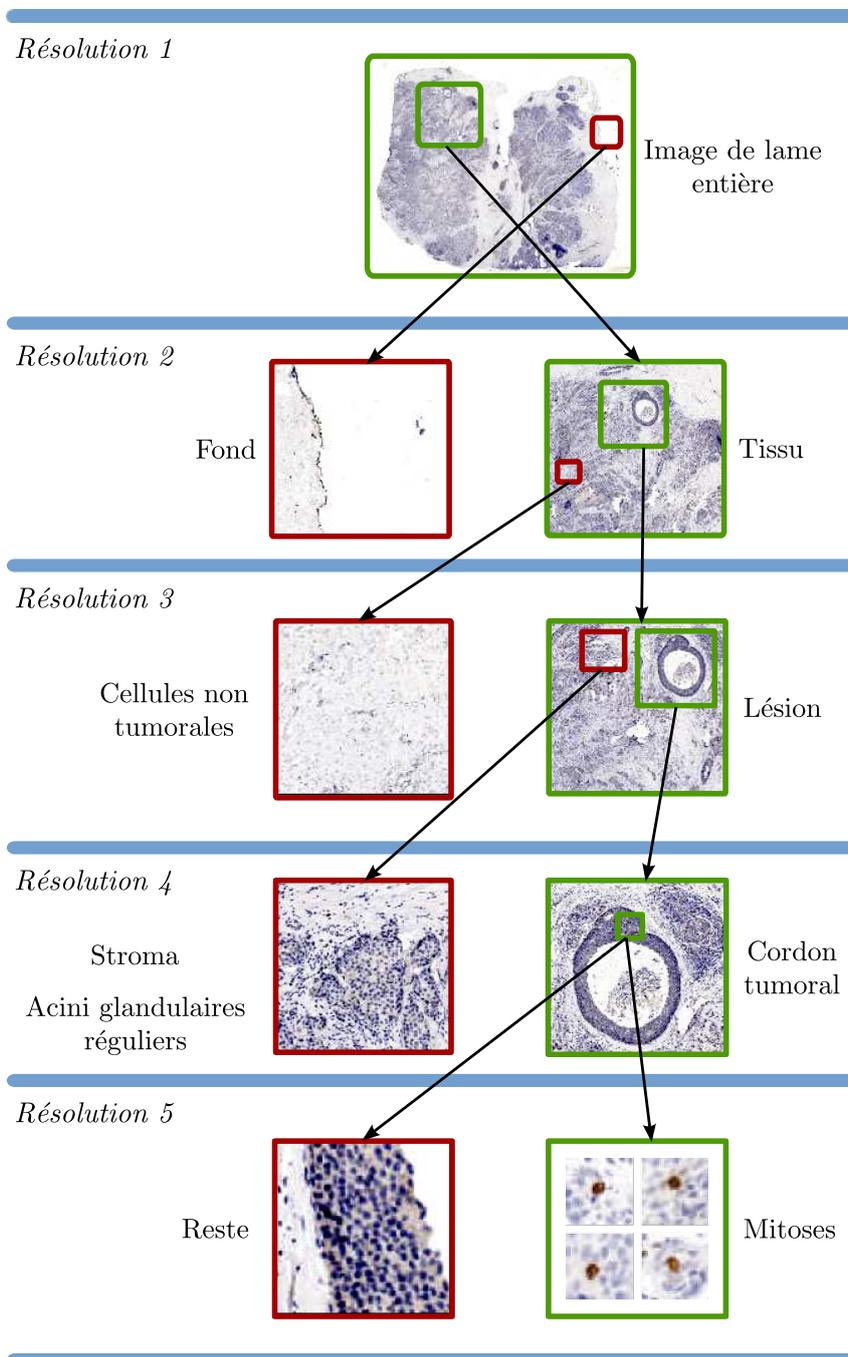


FIGURE 3 – Processus hiérarchique descendant pour l'analyse des lames histologiques.

niveau, le pathologiste détermine des zones d'intérêt comportant les éléments pertinents à l'établissement du diagnostic et raffine son observation de ces zones à la résolution supérieure jusqu'au niveau cellulaire.

Par conséquent, de façon à assister le processus d'analyse des pathologistes, il est nécessaire de disposer d'une solution logicielle adaptée à ces trois aspects fondamentaux des images histologiques.

Dans un premier temps, nous avons recherché une solution parmi les modèles de représentation d'images capables de coder les relations topologiques entre les objets d'une image. Il s'agit en effet de disposer d'une structure de données qui soit à la fois capable de décrire les objets du domaine et de représenter leurs relations d'adjacence et d'imbrication.

Dans un second temps, parmi les modèles topologiques nous avons sélectionné ceux qui ont la possibilité d'associer aux objets une information géométrique de façon à pouvoir caractériser les objets suivant leur forme.

Enfin, les modèles dits *hiérarchiques* permettent de représenter les images avec de multiples niveaux de détail. L'objet hiérarchique obtenu est appelé une *pyramide*. Deux types de pyramides doivent être distingués :

- Les pyramides *multi-résolution* représentent une pile d'images de résolutions décroissantes. La résolution d'une image désigne la surface réelle de l'image codée par un pixel.
- Les pyramides *multi-échelle* représentent une suite de partitions imbriquées. Une échelle désigne un niveau de raffinement d'une partition dans le modèle.

Les images histologiques se caractérisent par la présence de différents niveaux de détail à différentes résolutions. Ces images sont donc intrinsèquement multi-résolution et multi-échelle. Nous avons donc étudié les différents modèles hiérarchiques permettant de combiner efficacement les propriétés topologiques et géométriques d'une image multi-résolution.

Organisation du manuscrit

Ce manuscrit est organisé en quatre chapitres qui illustrent les étapes principales de la construction de notre modèle.

Dans le Chapitre 1, nous proposons un état de l'art des différents modèles existants pour la représentation d'images. Nous étudions en particulier les modèles relationnels capables de représenter les relations d'adjacence et d'imbrication entre les objets d'une image et leurs extensions hiérarchiques pour traiter les images multi-résolution. Cette recherche bibliographique nous a amené à la conclusion qu'il n'existait pas de modèle hiérarchique descendant adapté à la représentation d'information topologique et géométrique de grandes

images multi-résolution. En revanche, le modèle de pyramide combinatoire offrant plusieurs des propriétés recherchées, nos travaux se sont portés sur la définition d'un nouveau modèle fondé sur les pyramides combinatoires.

Notre première contribution est une extension au modèle de carte combinatoire pour la représentation de très grandes images. Il s'agit en effet de la contrainte technique majeure à laquelle sont confrontés les pathologistes lors de l'analyse des images histologiques : l'image à pleine résolution est trop volumineuse pour être entièrement chargée en mémoire ce qui exclut *de facto* l'utilisation de nombreux modèles de représentation. Nous avons adopté une approche dite *par blocs* permettant de décomposer en tuiles une carte combinatoire. Ces travaux font l'objet du Chapitre 2 et conduisent à la définition d'un nouveau modèle que nous appelons *carte tuilée* qui permet d'utiliser le modèle de carte combinatoire pour la représentation de grandes images 2D.

Notre seconde contribution est l'intégration de notre modèle de carte tuilée dans un processus de construction hiérarchique descendant. En effet, dans le cadre des pyramides combinatoires, les méthodes de construction existantes pour la représentation d'images sont uniquement ascendantes. Ces méthodes sont ainsi fondées sur l'opération de fusion de régions en partant de la partition la plus détaillée codant la résolution la plus importante. L'avantage de proposer un processus de construction descendant, fondé sur le raffinement des partitions depuis la résolution la plus faible, est double. Tout d'abord, cela permet de reproduire le comportement des pathologistes (analyse descendante). De plus, cela permet de réduire le coût mémoire de la structure de données et d'optimiser le temps de traitement en ne raffinant que les zones d'intérêt. Nous présentons ces travaux dans le Chapitre 3 à travers la définition du modèle de *pyramide descendante tuilée*.

Enfin, dans le Chapitre 4, nous nous intéressons au cas pratique de l'utilisation des pyramides descendantes tuilées pour la segmentation de grandes images. Nous conduisons différentes expérimentations à partir de l'implantation du modèle que nous avons réalisée et nous illustrons les possibilités offertes par celui-ci dans le cadre de la segmentation d'images histologiques multi-résolution.

Chapitre 1

Représentation d'images

Sommaire

1.1	Préliminaires	18
1.1.1	Image	18
1.1.2	Notions de base de topologie	19
1.1.3	Complexes cellulaires abstraits	21
1.1.4	Régions et partitions	25
1.2	Structures géométriques	26
1.2.1	Tableaux d'étiquettes	27
1.2.2	Chemins frontières	28
1.2.3	Frontières interpixel	30
1.3	Structures relationnelles	31
1.3.1	Graphes d'adjacence de régions	32
1.3.2	Graphes duaux	34
1.3.3	Cartes combinatoires	36
1.3.4	Cartes topologiques	42
1.4	Structures hiérarchiques	47
1.4.1	Pyramides régulières	49
1.4.2	Pyramides de graphes simples	52
1.4.3	Processus de décimation	53
1.4.4	Pyramides de graphes duaux	56
1.4.5	Pyramides combinatoires	58
1.4.6	Pyramides généralisées	61
1.5	Conclusion	63

1.1 Préliminaires

Le traitement d'images consiste à utiliser différents outils mathématiques pour exploiter l'information contenue dans une image. En particulier, la segmentation permet de regrouper certaines composantes en régions pour donner un sens aux données. Nous commençons par préciser la notion d'image numérique dans la Section 1.1.1. Dans la Section 1.1.2, nous introduisons les définitions fondamentales de topologie qui constituent la base de nos travaux. Dans la Section 1.1.3, nous étudions la topologie des complexes cellulaires abstraits qui consiste à considérer une image comme un ensemble de cellules interconnectées. Cette introduction nous amène à la notion de partition d'une image en régions dans la Section 1.1.4.

1.1.1 Image

Le terme *image* est employé pour qualifier la représentation numérique d'une information analogique, généralement associée à une scène naturelle. L'analyse de ces données permet d'exploiter l'information contenue dans cette représentation numérique et définit le *traitement d'image*. Cette étape de traitement implique alors une transformation de l'espace de mesure continu à un espace discret par une opération appelée la *discrétisation*. Cette discrétisation s'effectue par un pavage de l'espace initial. Le pavage utilise un motif pour réaliser un remplissage du plan sans trous et sans chevauchements. Dans le cas de la discrétisation de \mathbb{R}^2 , le pavage est dit régulier lorsqu'il respecte les contraintes suivantes :

- le motif est un polygone convexe ;
- les sommets sont en correspondance ;
- les cellules sont identiques et ont des côtés de même longueur.

Il existe trois différents pavages réguliers d'un espace de dimension 2 : les pavages carrés, triangulaires et hexagonaux (Figure 1.1).

En 2D, le pavage habituellement utilisé est le pavage carré. Une image est ainsi définie comme un partitionnement de \mathbb{R}^2 en pixels (*picture elements* en anglais) où chaque pixel est un élément du pavage de coordonnées entières (i, j) . Ce partitionnement est naturellement représenté par un sous-ensemble de \mathbb{Z}^2 constituant un tableau où l'indice (i, j) correspond aux coordonnées d'un pixel. Nous nous plaçons dans un repère dont l'origine est le pixel en haut à gauche de l'image. Une image I est ainsi définie par un couple (D, f) où :

- D est le *domaine* de l'image désignant un sous-ensemble fini du plan $P = \{(i, j) \in \mathbb{Z}^2\}$;
- f est une fonction de D vers l'ensemble des couleurs ou des intensités de l'image.

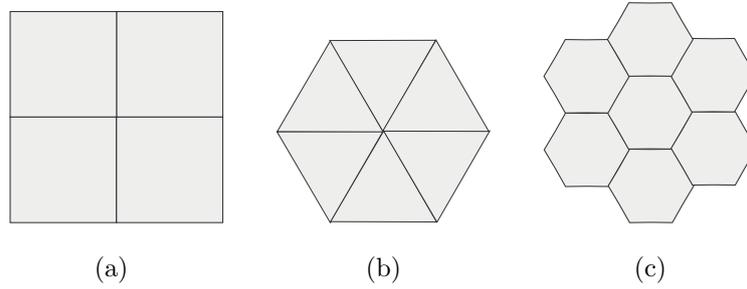


FIGURE 1.1 – Les trois différents pavages en dimension 2. (a) Pavage carré; (b) Pavage triangulaire; (c) Pavage hexagonal.

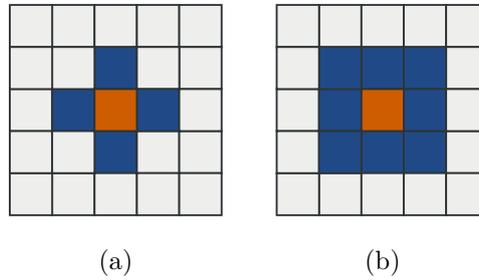


FIGURE 1.2 – Notion de connexité entre les pixels d'une image. (a) 4-connectivité : les quatre pixels 4-connexes au pixel central sont représentés en couleur sombre; (b) 8-connectivité : les huit pixels 8-connexes au pixel central sont représentés en couleur sombre.

Nous introduisons la notion de *connexité* entre les pixels composant une image I . Deux pixels $p(i, j)$ et $p'(i', j')$ sont dits :

- 4-connexes si et seulement si $|i - i'| + |j - j'| = 1$ (Figure 1.2(a));
- 8-connexes si et seulement si $|i - i'| + |j - j'| \leq 2$ et $\max(|i - i'|, |j - j'|) = 1$ (Figure 1.2(b)).

1.1.2 Notions de base de topologie

Un espace topologique est un couple (E, τ) où E est un ensemble et τ sa topologie, définie comme l'ensemble des ouverts de E .

Définition 1 (Espace topologique). *Soit un espace E . τ est une topologie de E si τ est une famille de parties, appelées ouverts vérifiant :*

- toute réunion d'ouverts est un ouvert;
- toute intersection finie d'ouverts est un ouvert;
- E et \emptyset sont des ouverts.

Un sous-ensemble de E est dit *fermé* si son complémentaire dans E appartient à τ .

Définition 2 (Voisinage). Soit (E, τ) un espace topologique et e un point de E . Un voisinage de e est une partie V de E telle qu'il existe un ouvert O de τ vérifiant : $e \in O \subset V$. L'ensemble des voisinages de e se note $V(e)$.

Définition 3 (Adhérence). Soit X une partie d'un espace topologique E . L'adhérence de X dans E , noté \overline{X} , est le plus petit fermé de E contenant X :

$$e \in \overline{X} \Leftrightarrow \forall V \in V(e), V \cap X \neq \emptyset.$$

Définition 4 (Intérieur). Soit X une partie d'un espace topologique E . L'intérieur de X dans E , noté $\overset{\circ}{X}$, est le plus grand ouvert de E contenu dans X :

$$e \in \overset{\circ}{X} \Leftrightarrow \exists V \in V(e) \mid V \subset X.$$

Nous pouvons remarquer que $\overset{\circ}{X} \subset X \subset \overline{X}$.

Définition 5 (Bord). Soit X une partie d'un espace topologique E . Le bord de X , noté $\partial(X)$ est la différence entre l'adhérence et l'intérieur de X :

$$\partial(X) = \overline{X} - \overset{\circ}{X}.$$

Définition 6 (Homéomorphisme). Soit X et Y deux espaces topologiques. $f : X \rightarrow Y$ est un homéomorphisme si f est une bijection de X vers Y et f et f^{-1} sont continues¹. Lorsqu'il existe un homéomorphisme entre X et Y , les deux espaces sont dits homéomorphes.

Ces définitions de base nous permettent d'aborder la notion de variété que nous utilisons dans la suite de ce manuscrit.

Une *variété topologique* de dimension n est un espace séparé² dans lequel chaque point dispose d'un voisinage homéomorphe à une boule de dimension n ou à une demi-boule si la variété topologique est ouverte.

Une *quasi-variété* en dimension n est un objet nD obtenu par l'assemblage de n -cellules le long de $(n - 1)$ -cellules de façon à ce que chaque $(n - 1)$ -cellule

1. Une fonction f entre deux espaces topologiques X et Y est continue si l'image réciproque de chaque ouvert de Y est un ouvert de X .

2. Un espace topologique séparé, ou espace de *Hausdorff*, est un espace topologique dans lequel pour chaque couple de points x et y distincts, il existe un voisinage de x et un voisinage de y distincts.

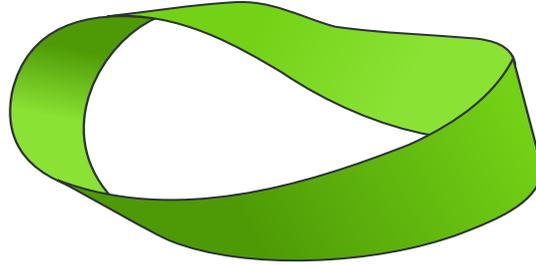


FIGURE 1.3 – Le ruban de Möbius : une 2-variété non-orientable.

soit incidente à au plus deux n -cellules. Notons qu’une quasi-variété est une variété en dimension 2 mais que cette propriété n’est pas forcément vérifiée en dimension supérieure [Lie94].

Une variété ou quasi-variété en dimension n est dite *orientable* s’il est possible de définir de façon globale une direction à *gauche* et à *droite* en tout point. Dans le cas contraire, elle est dite *non-orientable*. Le ruban de Möbius (Figure 1.3) est un exemple célèbre de 2-variété non-orientable.

1.1.3 Complexes cellulaires abstraits

Les espaces d’*Alexandroff* sont des espaces topologiques dont chaque élément dispose d’un voisinage minimal [Ale37]. Ces espaces sont très utilisés en topologie digitale. Dans notre cas, un complexe cellulaire abstrait est un espace d’Alexandroff localement fini dont les éléments sont appelés k -cellules où k désigne la dimension de la cellule. Intuitivement, un complexe cellulaire abstrait [Whi49] est un pavage du plan dont tous les éléments sont pris en compte. La dimension d’un complexe cellulaire abstrait correspond à la plus grande dimension de ses cellules. Dans le cadre de la représentation d’images 2D, nous nous intéressons donc uniquement aux complexes cellulaires de dimension 2 (Figure 1.4) tels que définis par [Kov89, KKM90, KKM91, Fio95]. Les k -cellules d’un complexe cellulaire sont qualifiées d’abstraites car elles ne sont pas associées à un plongement particulier.

Définition 7 (Complexe cellulaire abstrait). *Un complexe cellulaire abstrait C est un triplet $C = (E, B, \dim)$ vérifiant :*

- E : un ensemble de k -cellules (éléments abstraits) ;
- $B \subset E \times E$: une relation binaire, non réflexive, anti-symétrique et transitive appelée relation de bord ;
- $\dim : E \rightarrow \mathbb{N}$: une fonction de dimension telle que :

$$\forall (c, c') \in B, c \neq c' \Rightarrow \dim(c) < \dim(c').$$

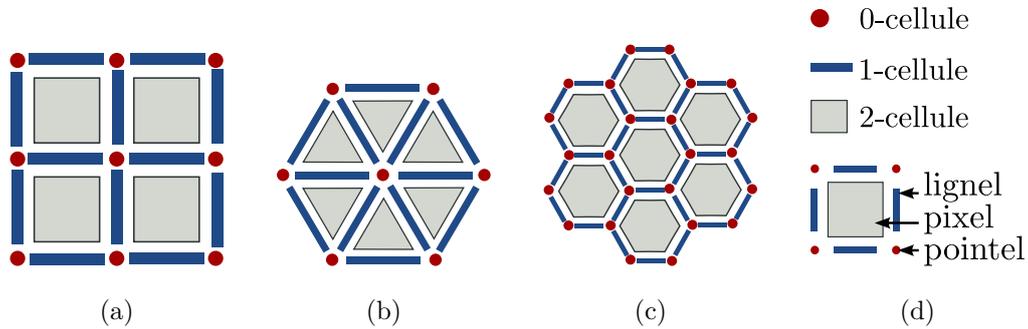


FIGURE 1.4 – Complexes cellulaires en dimension 2. (a) Pavage carré; (b) Pavage triangulaire; (c) Pavage hexagonal; (d) Notations.

Françon définit le terme d'*éléments interpixel* [Fra91] pour désigner les k -cellules : les termes *pointel*, *lignel* et *pixel* désignent respectivement une k -cellule de dimension 0, 1 et 2.

Définition 8 (Sous-complexe). Soit $C = (E, B, \dim)$ un complexe cellulaire abstrait. Le triplet $S = (E', B', \dim')$ est un sous-complexe de C si et seulement si

- E' est un sous-ensemble de E ;
- B' est l'intersection de B avec $E' \times E'$;
- \dim' est la restriction de la fonction \dim à E' .

La dimension d'un sous-complexe cellulaire est donnée par la plus grande dimension de ses cellules. Dans la mesure où un sous-complexe est défini par son ensemble de cellules, les opérations booléennes ensemblistes comme l'union, l'intersection et le complémentaire s'étendent directement aux sous-complexes.

Définition 9 (Sous-complexe ouvert). Un sous-complexe S d'un complexe C est dit ouvert dans C si, pour chaque cellule s de S , il contient toutes les cellules de C bordées par s .

À l'inverse, un sous-complexe est dit *fermé* dans un complexe C si son complémentaire dans C est ouvert. Nous introduisons plusieurs définitions permettant d'appliquer les notions d'adhérence, d'intérieur et de bord (Section 1.1.2) aux complexes cellulaires abstraits.

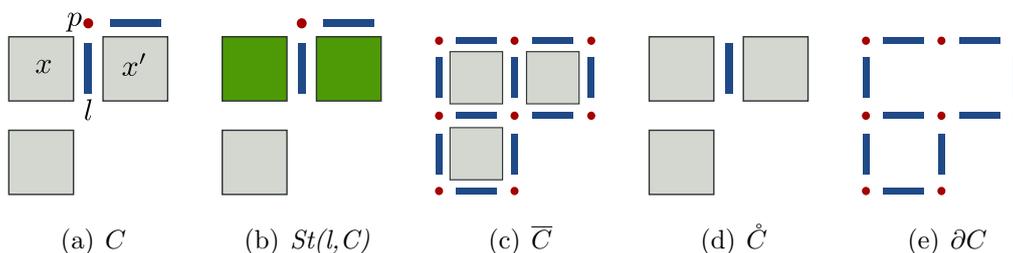


FIGURE 1.5 – Topologie d'un complexe cellulaire. (a) Un complexe C ; (b) Étoile de la 1-cellule l ; (c) Adhérence de C ; (d) Intérieur de C ; (e) Bord de C .

Définition 10 (Étoile). *L'étoile d'une k -cellule c appartenant à un complexe C , notée $St(c,C)$ est composée de c et de l'ensemble des éléments de C bordés par c .*

L'étoile d'une cellule correspond au plus petit voisinage contenant cette cellule. Ainsi, pour tout couple de k -cellules $(c, c') \in C \times C$, avec $C = (E, B, dim)$ et $c \neq c'$, nous avons $c' \in St(c,C) \Leftrightarrow (c, c') \in B$.

Définition 11 (Adhérence). *L'adhérence d'un sous-complexe S d'un complexe C , notée \bar{S} , est l'ensemble des k -cellules $c \in C$ dont l'étoile intersecte S :*

$$c \in \bar{S} \Leftrightarrow St(c,C) \cap S \neq \emptyset.$$

Définition 12 (Intérieur). *L'intérieur d'un sous-complexe S d'un complexe C , noté $\overset{\circ}{S}$, est l'ensemble des k -cellules $c \in C$ dont l'étoile est incluse dans S :*

$$c \in \overset{\circ}{S} \Leftrightarrow St(c,C) \subset S.$$

Définition 13 (Bord). *Le bord d'un sous-complexe S de dimension n d'un complexe C , noté ∂S , est la différence entre l'adhérence et l'intérieur de S :*

$$\partial S = \bar{S} - \overset{\circ}{S}.$$

La Figure 1.5 illustre les définitions d'étoile, d'adhérence, d'intérieur et de bord sur l'exemple d'un complexe C de dimension 2.

Définition 14 (Cardinal). *Le nombre total de cellules de toutes dimensions d'un complexe C est appelé cardinal de C et est noté $|C|$.*

Kovalevsky définit les notions d'incidence (Définition 15) et d'adjacence (Définition 16) entre les k -cellules [Kov00].

Définition 15 (Incidence). *Une cellule c est incidente à une cellule c' si et seulement si c est égale à c' , c borde c' ou c' borde c .*

Définition 16 ((k, l) -adjacence). *Deux l -cellules ($l > k$) sont dites k -adjacentes si elles sont toutes deux incidentes à une même k -cellule.*

Définition 17 (Degré). *Le degré d'une l -cellule correspond au nombre de $(l + 1)$ -cellules qu'elle borde.*

Dans le cas du complexe illustré Figure 1.5(a), les deux pixels x et x' sont $(1, 2)$ -adjacents, le pointel p est incident au lignel l . Les deux cellules p et l sont de degré 2.

Définition 18 ((k, l) -chemin). *Une séquence (c_0, c_1, \dots, c_n) définit un (k, l) -chemin de l -cellules ($l > k$) de c_0 à c_n si et seulement si $\forall p, 0 \leq p < n/2$,*

- $\dim(c_{2p}) = k$;
- $\dim(c_{2p+1}) = l$;
- $(c_{2p}, c_{2p+1}) \in B$.

Nous appelons (k, l) -chemin simple un (k, l) -chemin vérifiant : $\forall(i, j), i \neq j, c_i \neq c_j$. Considérons le cas d'un pavage carré. La Figure 1.6 illustre les trois cas de chemins simples envisageables :

1. $k = 0$ et $l = 1$ (Figure 1.6(a)) : le chemin considéré est une *ligne topologique*³ composée d'une séquence alternée de pointels et de lignels ;
2. $k = 1$ et $l = 2$ (Figure 1.6(b)) : chaque couple de pixels consécutifs (c_{2p+1}, c_{2p+3}) sont incidents à un même lignel (4-connexité) ;
3. $k = 0$ et $l = 2$ (Figure 1.6(c)) : les deux pixels (c_{2p+1}, c_{2p+3}) sont incidents à un même pointel (8-connexité).

Grâce à la notion de (k, l) -chemin (Définition 18), nous pouvons introduire dans la Définition 19 la notion de (k, l) -composante connexe qui permet de caractériser un ensemble de cellules.

Définition 19 ((k, l) -composante connexe). *Soit S un sous-complexe de C . Le sous-complexe S est une (k, l) -composante connexe de C si pour tout couple (c_1, c_2) de l -cellules contenues dans S , il existe un (k, l) -chemin inclus dans S reliant c_1 à c_2 . L'ensemble des (k, l) -composantes connexes de S est noté (k, l) -cc(S).*

Dans la suite de ce manuscrit, nous nous plaçons dans le cadre d'un pavage carré et nous travaillons en 4-connexité. En général, nous considérons $(k, l) = (1, 2)$.

3. Une ligne topologique est un sous-complexe connexe de dimension 1 dont chaque cellule, excepté deux, est incidente à exactement deux autres cellules.

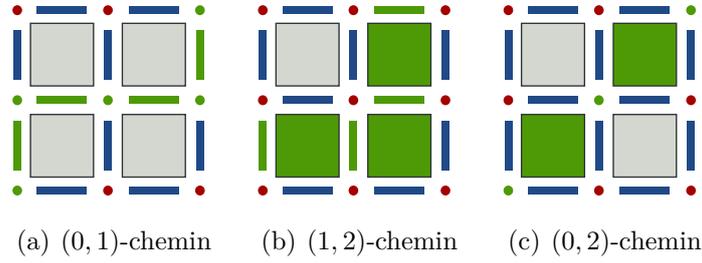


FIGURE 1.6 – (k, l) -chemins dans un complexe cellulaire. (a) $(0, 1)$ -chemin composé de pointels et de lignels; (b) $(1, 2)$ -chemin composé de lignels et de pixels; (c) $(0, 2)$ -chemin composé de pointels et de pixels.

1.1.4 Régions et partitions

En traitement d'images, nous appelons *région* un ensemble homogène de pixels au sens d'un *critère*. Les critères d'homogénéité définissant les régions d'une image peuvent être de natures différentes, les plus utilisés étant les critères colorimétriques fondés sur l'intensité des pixels. Néanmoins, d'autres critères peuvent être envisagés. Par exemple, des critères géométriques comme un coefficient d'élongation ou des critères topologiques utilisant les caractéristiques d'imbrication et d'adjacence peuvent être envisagés.

Définition 20 (Région). *Soit C un complexe cellulaire abstrait de dimension 2. Une région r est une $(1, 2)$ -composante connexe ouverte dans C .*

Par conséquent, une région désigne un ensemble de pixels 4-connexes. Nous pouvons remarquer que toute 1 -cellule d'une région est incidente à au moins une 2 -cellule de la même région. L'ensemble des pixels d'une région r est noté $pixels(r)$. Nous parlons de *partition* d'une image lorsqu'une image est partitionnée en régions (Définition 21).

Définition 21 (Partition). *Soit I une image considérée comme un complexe cellulaire de dimension 2. L'ensemble $P = \{r_0, \dots, r_n\}$ est une partition de I si et seulement si $\forall i \in \{0, \dots, n\}$:*

- r_i est une région (Définition 20);
- $\forall j \in \{0, \dots, n\}, j \neq i, \overline{r_i} \cap \overline{r_j}$ est un complexe cellulaire de dimension au plus 1, éventuellement vide;
- $\bigcup_{k=0}^n \overline{r_k} = I$.

Nous introduisons également une *région infinie*, notée r_∞ correspondant au complémentaire de l'image dans \mathbb{Z}^2 . À partir de la notion de partition, nous nous intéressons alors aux relations d'adjacence entre les régions.

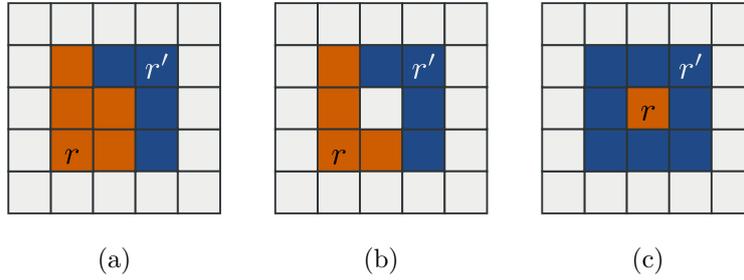


FIGURE 1.7 – Différents types d'adjacence entre deux régions r et r' . (a) Adjacence simple ; (b) Adjacence multiple ; (c) Imbrication.

Définition 22 (Adjacence de régions). *Deux régions r et r' sont dites adjacentes si et seulement si $\partial(r) \cap \partial(r') \neq \emptyset$ et $\dim(\partial(r) \cap \partial(r')) = 1$.*

Notons que d'après la Définition 21, pour tout couple de régions (r, r') , l'intersection de leurs adhérences est un sous-complexe de dimension au plus 1 donc nous avons $\dim(\partial(r) \cap \partial(r')) \leq 1$. Dans la Définition 22, le fait d'imposer l'égalité $\dim(\partial(r) \cap \partial(r')) = 1$ permet d'interdire la 8-adjacence de régions.

Trois types d'adjacence entre les régions peuvent être distingués et sont illustrés dans la Figure 1.7 : l'adjacence simple (Figure 1.7(a)), la multi-adjacence (Figure 1.7(b)) et l'imbrication (Figure 1.7(c)).

Définition 23 (Multi-adjacence de régions). *Deux régions adjacentes r et r' sont dites multi-adjacentes si et seulement si $|cc(\partial(r) \cap \partial(r'))| > 1$.*

Définition 24 (Imbrication de régions). *Une région r adjacente à une autre région r' est dite imbriquée dans r' si et seulement si pour toute 2-cellule c de r , tout $(1, 2)$ -chemin de c à r_∞ contient au moins une 2-cellule appartenant à r' .*

Notons que la relation d'imbrication est transitive : si r_1 est imbriquée dans r_2 et que r_2 est imbriquée dans r_3 alors r_1 est imbriquée dans r_3 .

Les partitions sont des structures particulièrement intéressantes qui ont donné lieu à de nombreux travaux. Dans la suite de ce manuscrit, nous étudions différentes structures permettant de représenter ces partitions en les regroupant en trois catégories : les structures géométriques (Section 1.2), les structures topologiques (Section 1.3) et les structures hiérarchiques (Section 1.4).

1.2 Structures géométriques

Les structures géométriques sont des structures codant une partition de l'image en privilégiant les caractéristiques géométriques des régions plutôt que

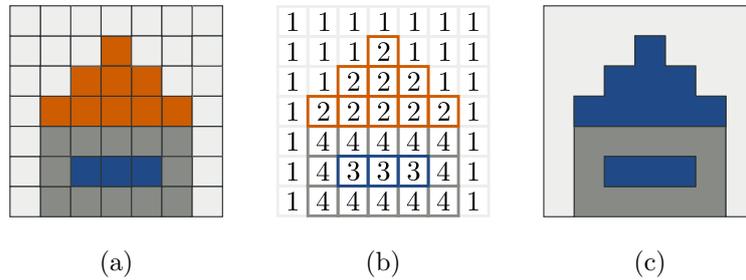


FIGURE 1.8 – Tableaux d’étiquettes. (a) Partition; (b) Image fortement étiquetée; (c) Image faiblement étiquetée.

leurs relations d’adjacence. Elles permettent d’évaluer des critères comme la forme ou la longueur du bord des régions d’une partition. Nous présentons ainsi différentes approches permettant de représenter la géométrie des régions. Nous abordons les notions d’image étiquetée (Section 1.2.1), de contour en tant que chemin de pixels connexes (Section 1.2.2) puis de frontière interpixel (Section 1.2.3).

1.2.1 Tableaux d’étiquettes

Les tableaux d’étiquettes [FV92, SC83, Nic95] associent une *étiquette* (ou *label*) à tous les pixels d’une partition pour indiquer leur appartenance à une région donnée. Deux cas peuvent être distingués :

- l’étiquetage *fort* : deux pixels ont la même étiquette si et seulement s’ils appartiennent à la même région ;
- l’étiquetage *faible* : deux régions non connexes peuvent avoir la même étiquette.

La représentation par tableau d’étiquettes est illustrée en Figure 1.8. Dans cet exemple, nous considérons une partition composée de quatre régions (Figure 1.8(a)). La Figure 1.8(b) donne un exemple d’étiquetage fort : chacun des pixels de ces quatre régions est associé à un identifiant entier unique. La Figure 1.8(c) présente l’image étiquetée où une couleur a été associée à chaque étiquette : l’étiquetage est faible car deux régions sont identifiées par la même couleur.

Avantages et inconvénients

Cette représentation a l’avantage d’être facile à mettre en œuvre et est adaptée aux traitements locaux impliquant peu de re-étiquetages. En revanche, le modèle est peu adapté pour des opérations de découpe et de fusion de régions

en raison du grand nombre d'étiquettes à mettre à jour. Un autre inconvénient des tableaux d'étiquettes provient de sa non compacité : l'information est très redondante, donc coûteuse en mémoire. Enfin, les tableaux d'étiquettes ne définissent pas explicitement les frontières des régions. La structure n'est pas adaptée à une analyse des relations topologiques comme par exemple, déterminer de manière efficace si deux régions sont adjacentes ou imbriquées l'une dans l'autre.

Extensions

Différentes extensions et améliorations ont été proposées pour pallier les inconvénients des tableaux d'étiquettes. Par exemple, un *codage par plages* (*Run-length Encoding* en anglais) répond en partie au problème de compacité du modèle en compressant les étiquettes redondantes. Pour chaque ordonnée, nous indiquons quels sont les intervalles de pixels appartenant à une région donnée. En Figure 1.9(b), le codage par plage des régions 1 et 2 est donné à titre d'exemple. Cette approche permet ainsi une meilleure compression des données et facilite le balayage des pixels mais ne répond pas au problème d'accès aux frontières des régions.

Les *forêts d'ensembles disjoints* sont intéressantes en tant que structure complémentaire. Une *forêt* est un ensemble fini d'*arbres*. Un arbre est un graphe sans cycle composé d'un ensemble fini de *nœuds* où chaque nœud a une relation parent vers un nœud différent sauf pour un nœud particulier appelé *racine* dont le parent est lui-même. Pour une forêt d'ensembles disjoints, le principe consiste à représenter chaque composante connexe par un arbre [Tar75, CLRS90]. Les deux opérations permises par la structure sont l'*union* (*union* en anglais) et la *recherche* (*find* en anglais). La recherche utilise le lien parent jusqu'à trouver la racine de la hiérarchie. L'union de deux régions se résume à fusionner leurs racines ce qui permet de mettre à jour efficacement le modèle après une opération de fusion entre deux régions. Une illustration est donnée par la Figure 1.9(c).

1.2.2 Chemins frontières

Un inconvénient majeur des structures dérivées des tableaux d'étiquettes vient de la représentation des frontières des régions. En effet, l'information photométrique et/ou géométrique des frontières peut être utilisée lors des opérations de fusion ou de découpe des régions. Il est donc souvent nécessaire d'avoir accès aux frontières des régions sans devoir parcourir l'ensemble de ses pixels.

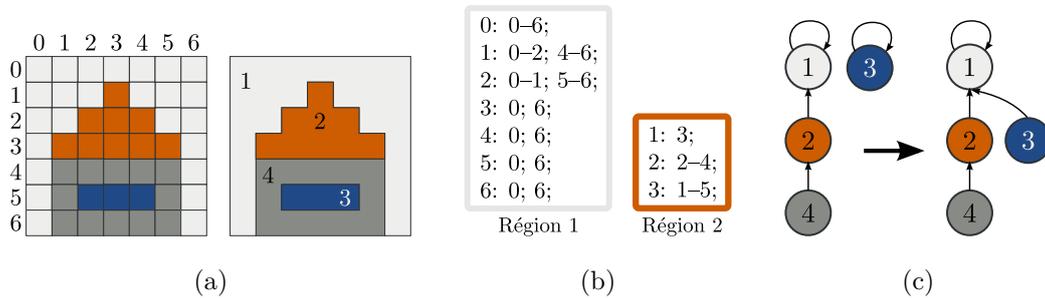


FIGURE 1.9 – Extensions des tableaux d'étiquettes. (a) Partition et image étiquetée; (b) Codage des régions par plage; (c) Forêts d'ensembles disjoints : la fusion des régions 3 et 4 consiste à fusionner les racines des deux ensembles.

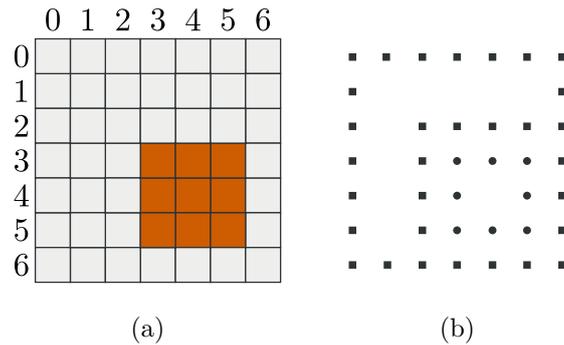


FIGURE 1.10 – Frontières des régions. (a) Partition; (b) Frontières des régions : les marques carrées et circulaires désignent respectivement les pixels correspondant aux frontières de la région claire et de la région foncée.

Une première approche pour la représentation des frontières consiste à identifier tous les couples (p, p') de pixels adjacents d'étiquette différente lors d'un balayage de l'image. Cette approche définit les frontières de la partition. Cependant, il est difficile de déterminer si une frontière est un bord intérieur ou extérieur d'une région, ce qui est nécessaire pour détecter les imbrications. Dans la Figure 1.10(b), l'adjacence simple entre la région claire et la région foncée de la Figure 1.10(a) est définie par une séquence de couples de pixels appartenant à la région foncée (marques circulaires) et à la région claire (marques carrées).

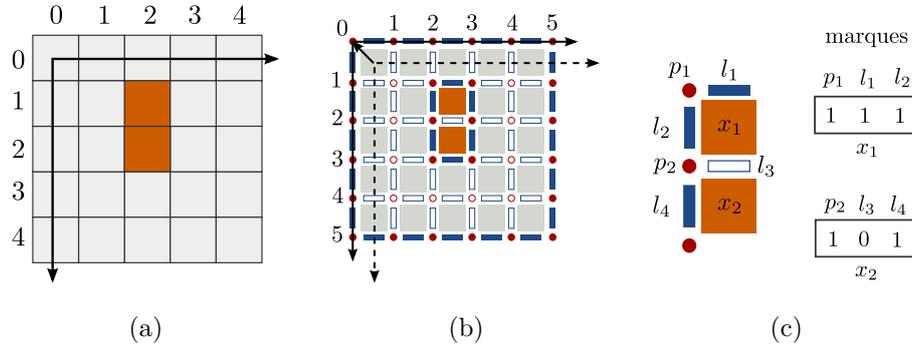


FIGURE 1.11 – Matrice de Khalimsky. (a) Axes de coordonnées de l'image; (b) Repère interpixel obtenu par translation de $(-\frac{1}{2}, -\frac{1}{2})$; (c) Matrice de Khalimsky : la région foncée est composée de deux pixels x_1 et x_2 de coordonnées respectives $(2, 1)$ et $(2, 2)$. Les indices $(2, 1)$ et $(2, 2)$ de la matrice de Khalimsky désignent 2×3 bits correspondant respectivement aux éléments (p_1, l_1, l_2) pour x_1 et (p_2, l_3, l_4) pour x_2 . La valeur correspond à la marque déterminant si l'élément est frontière (1) ou non (0).

1.2.3 Frontières interpixel

Matrice de Khalimsky

Dans le cadre d'un complexe cellulaire abstrait de dimension 2 avec pavage carré, les frontières des régions sont uniquement composées de pointels et de lignels. Une *matrice de Khalimsky* [Kov89, KKM90] permet de décrire toutes les cellules du complexe à l'aide d'un tableau de dimension $2n + 1$, n étant la taille de l'image. Chaque case du tableau contient un booléen indiquant si la cellule correspondante appartient au complexe.

En termes d'implantation, cette structure peut être optimisée en codant un pointel et deux lignels dans une même cellule. Nous utilisons une *marque* pour les pointels et les lignels appartenant aux bords des régions, appelés *éléments frontières*. Plus précisément, considérons un tableau de taille $(l+1) \times (h+1)$ où l et h désignent respectivement la largeur et la hauteur de l'image. L'indice (i, j) d'un élément du tableau correspond au centre du pixel $p = (i, j)$ de l'image. Par une translation du plan discret par $(-\frac{1}{2}, -\frac{1}{2})$ (Figure 1.11(b)), (i, j) désigne les coordonnées du pointel haut-gauche appartenant au bord de p . Un bit permettant d'indiquer si un pointel/lignel est un élément frontière ou non, trois bits sont alors associés à chaque pixel p : un bit pour le pointel (i, j) et deux bits pour les deux lignels incidents au pointel (i, j) appartenant au bord de p .

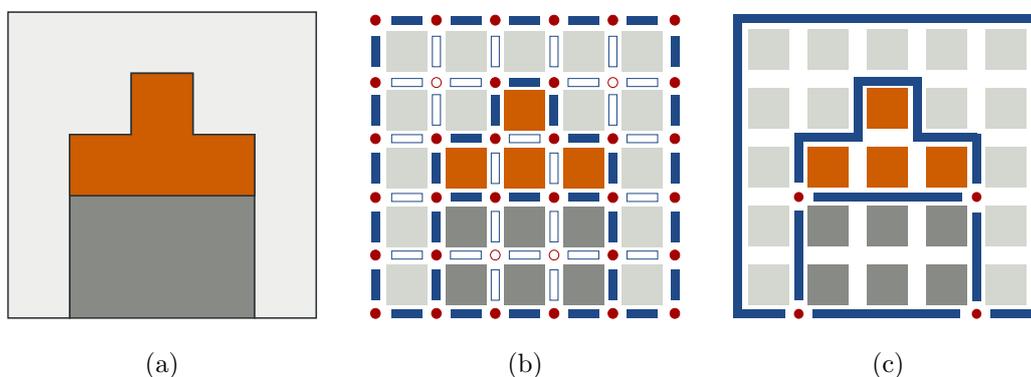


FIGURE 1.12 – Frontières interpixel. (a) Image; (b) Éléments interpixel : les pointels et lignels frontières sont représentés en cercles et rectangles pleins; (c) Représentation minimale des frontières par des segments et des nœuds.

Représentation minimale des frontières

Nous utilisons les éléments frontières de la représentation interpixel pour définir de manière minimale les frontières des régions d'une partition en termes de *segments* et de *nœuds* [Dom92], comme illustré dans la Figure 1.12(c).

Définition 25 (Segment). *Un segment est un chemin frontière maximal⁴ entre deux régions.*

Définition 26 (Nœud). *Un nœud résulte de l'intersection d'au moins trois segments (un segment connecte deux nœuds).*

Puisqu'un segment est une séquence ordonnée et alternée de pointels et de lignels, deux orientations sont envisageables et chaque segment définit deux *segments orientés*.

1.3 Structures relationnelles

Les structures relationnelles permettent d'exploiter des propriétés topologiques comme l'adjacence ou l'imbrication entre les régions d'une partition. Dans cette section, nous présentons l'évolution des travaux sur ces modèles. Nous abordons ainsi les premiers travaux sur les graphes d'adjacence de régions dans la Section 1.3.1, l'extension apportée par les graphes duaux dans la Section 1.3.2 et les modèles de cartes combinatoires et topologiques dans la Section 1.3.3 et la Section 1.3.4.

4. Un chemin frontière est dit *maximal* s'il ne peut être prolongé sans que cela modifie une de ses deux régions incidentes.

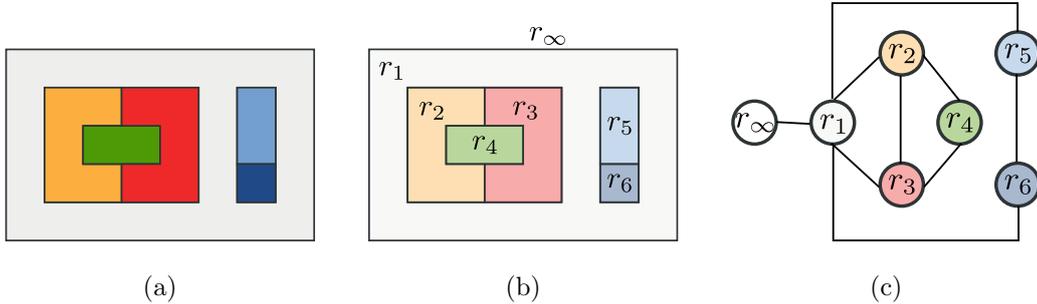


FIGURE 1.13 – Graphes d’adjacence (RAG). (a) Image; (b) Image étiquetée en six régions; (c) Graphe d’adjacence de régions : les sommets et les arêtes du graphe représentent respectivement les régions et les adjacences de l’image.

1.3.1 Graphes d’adjacence de régions

Les graphes d’adjacence de régions [Ros74] (*Region Adjacency Graph* ou *RAG* en anglais) font partie des premiers modèles relationnels visant à représenter les relations d’adjacence entre les régions d’une image (Figure 1.13(b)) : chaque région (y compris le fond) est associée à un sommet d’un graphe, les arêtes représentent les relations d’adjacence entre les régions. Ainsi, une arête entre deux sommets du graphe indique l’existence d’au moins un bord commun entre deux régions de l’image (Figure 1.13(c)). Notons qu’une variante du modèle des graphes d’adjacence, appelée *multi-graphe d’adjacence*, représente les adjacences multiples par des arêtes multiples entre deux sommets. Nous pouvons remarquer que la relation d’adjacence est symétrique, le graphe n’est donc pas orienté.

Les RAG définissent ainsi une structure de données simple et adaptée aux opérations de fusion de régions. En effet, fusionner deux régions adjacentes de la partition revient à fusionner deux sommets adjacents du graphe. Cette opération de *fusion de sommets* se décompose en trois étapes :

- identification des sommets à fusionner ;
- suppression de l’arête (Figure 1.14(b)) ;
- suppression de toutes les arêtes doubles et des boucles éventuellement créées par l’identification des sommets (Figure 1.14(c)).

Notons que les deux premières étapes correspondent à la contraction de l’arête entre les sommets fusionnés. Comme indiqué dans la Section 1.1.4, ce processus de fusion induit la création d’une nouvelle région connexe, union des régions codées par les deux sommets fusionnés. La Figure 1.14 illustre le processus de fusion appliqué aux régions r_2 et r_3 . Ainsi, la première étape identifie les deux sommets du graphe correspondant aux régions à fusionner

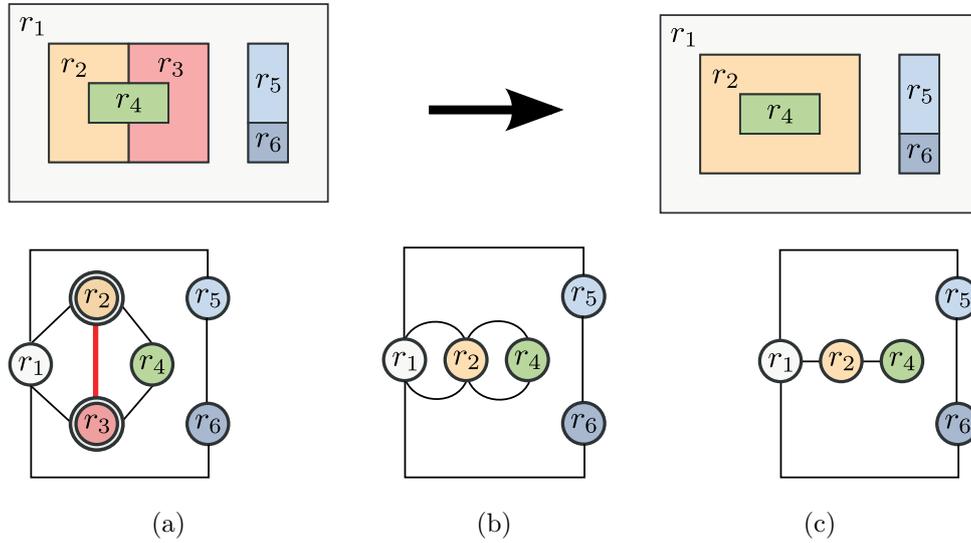


FIGURE 1.14 – Graphes d’adjacence : fusion de sommets. (a) Identification des sommets à fusionner (double cercles); (b) Suppression de l’arête; (c) Suppression des arêtes doubles.

(marqués par un double cercle en Figure 1.14(a)). La deuxième étape consiste à supprimer la relation d’adjacence entre r_2 et r_3 par suppression de l’arête qui les relie (en gras). Ces deux étapes correspondent à la contraction de l’arête entre r_2 et r_3 . Cette contraction entraîne la création de deux arêtes doubles entre r_1 et r_2 et entre r_2 et r_4 (Figure 1.14(b)) qui sont supprimées lors de la troisième étape pour obtenir le graphe final mis à jour (Figure 1.14(c)).

Il est possible d’associer certaines propriétés ou paramètres aux régions d’un graphe d’adjacence. Ces paramètres sont mis à jour lors des opérations de fusion et sont utilisés pour pondérer les arêtes du graphe. La moyenne des niveaux de gris des pixels d’une région ou sa variance sont deux exemples de paramètres dont la mise à jour se fait de manière incrémentale du fait de leur propriété d’additivité. En revanche, certains paramètres non additifs entraînent une mise à jour coûteuse dans le cadre d’un graphe d’adjacence. Par exemple, considérons la fusion de régions d’après une fonction booléenne *Oracle* indiquant si deux régions partagent un contour de faible gradient :

$$Oracle(r_1, r_2) = \min_{s \in \partial(r_1) \cap \partial(r_2)} \overline{\nabla I}^s < \epsilon,$$

avec ϵ un seuil quelconque et $\overline{\nabla I}^s = \frac{1}{|s|} \int_s \nabla I(t) dt$ représentant le gradient moyen le long d’une frontière s séparant r_1 et r_2 . Dans ce cas, le critère n’est pas additif et est basé sur des valeurs associées à chaque contour. Or, il n’est pas

possible de mettre efficacement à jour le minimum de l'ensemble de ces valeurs avec un graphe d'adjacence codant tous les contours entre deux régions par une seule arête. En effet, il faut dans ce cas parcourir à nouveau les frontières de chaque région après chaque fusion afin de mettre à jour correctement le poids de l'arête.

De plus, le modèle de graphe d'adjacence est bien adapté aux opérations de fusion mais ne l'est pas pour les opérations de découpe des régions du fait des nouvelles relations d'adjacence potentiellement créées. En effet, découper une région r en un ensemble de régions $\{r_1, \dots, r_n\}$ revient à supprimer r du graphe puis à parcourir $\{r_1, \dots, r_n\}$ pour mettre à jour les relations d'adjacences en insérant les nœuds et les arêtes adéquats. Un autre inconvénient des graphes d'adjacence est qu'ils ne permettent pas de différencier les cas de multi-adjacence des adjacences simples, ni de détecter les imbrications. Le modèle ne permet pas non plus de reconstruire la géométrie des frontières des régions. De ce fait, plusieurs extensions et modèles dérivés ont été proposés pour résoudre ces lacunes.

1.3.2 Graphes duaux

Les graphes duaux [WK94, KM95] maintiennent deux structures en parallèle : une extension du RAG, le multi-graphe d'adjacence $G = (\mathcal{S}, \mathcal{A})$ (Figure 1.15(b)) et son dual $\bar{G} = (\bar{\mathcal{S}}, \bar{\mathcal{A}})$, le *graphe des sommets* (Figure 1.15(c)). Dans un graphe des sommets, les sommets codent les intersections de frontières.

Cette propriété se traduit par des arêtes multiples. Dans l'exemple de la Figure 1.15(b), la multi-adjacence entre les deux régions r_2 et r_3 se traduit par deux arêtes entre les sommets correspondant à r_2 et r_3 . Les relations d'imbrication se retrouvent dans le primal sous forme de boucles et dans le dual sous forme de *pont* (ou *arête fictive*) dans la mesure où ces arêtes ne correspondent à aucune frontière réelle de l'image segmentée. Dans la Figure 1.15(b), la relation d'imbrication entre r_1 et la composante connexe $c = \{r_5, r_6\}$ se traduit par une boucle partant du sommet r_1 et englobant les régions r_5 et r_6 . Cette relation se retrouve dans le dual de la Figure 1.15(c) par une arête fictive (en pointillés) reliant le sommet s_0 à s_5 .

Le cas de la région infinie r_∞ est un peu particulier dans la mesure où nous savons qu'elle contient l'image par construction. Cette relation d'imbrication est donc parfois représentée implicitement. Dans notre cas, les deux composantes connexes $c_1 = \{r_2, r_3, r_4\}$ et $c_2 = \{r_5, r_6\}$ sont imbriquées dans la région r_1 ce qui se traduit par deux boucles dans le primal et deux arêtes fictives (en pointillés) dans le dual (Figure 1.15(d)).

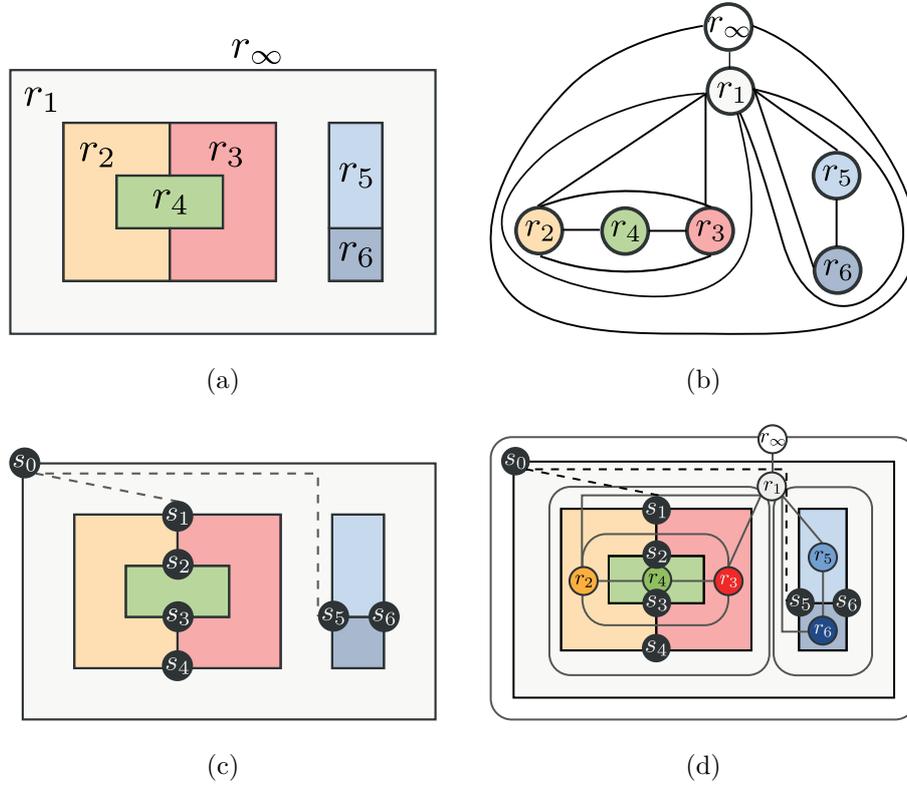


FIGURE 1.15 – Graphes duaux. (a) Image étiquetée composée de six régions ; (b) Multi-graphe d’adjacence représentant les multi-adjacences (arêtes multiples) et les imbrications (boucles) ; (c) Graphe des sommets : les imbrications se traduisent par des arêtes fictives (pointillés) ; (d) Superposition des graphes (b) et (c).

La suppression d’une arête a revient à la retirer de l’ensemble \mathcal{A} . Pour chaque opération effectuée sur le primal (*resp.* dual), il faut mettre à jour le dual (*resp.* primal) de façon adéquate. Par exemple, pour chaque arête supprimée dans le primal, il faut contracter l’arête correspondante dans le dual. Les deux opérations de *contraction* et de *suppression* sont dites duales. Cette dualité des opérations se traduit de la façon suivante. Considérons $G = (\mathcal{S}, \mathcal{A})$ un graphe d’adjacence, $\overline{G} = (\overline{\mathcal{S}}, \overline{\mathcal{A}})$ son graphe dual et une arête $a \in \mathcal{A}$ avec a' son image dans $\overline{\mathcal{A}}$. Nous notons $G \setminus a$ (*resp.* G/a) la contraction (*resp.* suppression) de l’arête a dans le graphe G . L’opération duale d’une contraction dans le primal est une suppression dans le dual : $\overline{G \setminus a} = \overline{G}/a'$. L’opération duale d’une suppression dans le primal est une contraction dans le dual : $\overline{G/a} = \overline{G} \setminus a'$.

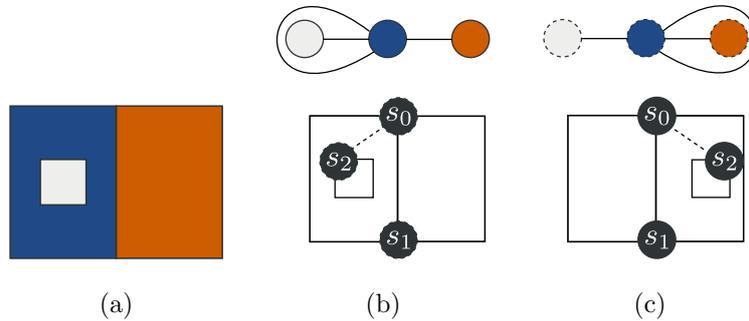


FIGURE 1.16 – Cas limite de caractérisation de la relation d'imbrication pour les graphes duaux. (a) Image; (b) Graphe dual associé; (c) Autre graphe dual équivalent.

Le cas des relations d'imbrication pour les modèles de graphes duaux n'est que partiellement résolu [KM95]. Bien que les ponts (arêtes fictives) et les boucles permettent d'indiquer qu'il existe une imbrication, il n'est pas toujours possible de la caractériser localement. Sur l'exemple proposé Figure 1.16, les deux graphes duaux de la Figure 1.16(b) et de la Figure 1.16(c) sont équivalents : il n'est donc pas possible de déterminer localement la relation d'imbrication.

Les modèles de graphes duaux permettent de représenter les multi-adjacences d'une image segmentée. Cependant, les relations d'imbrication restent partiellement représentées. Il est néanmoins possible de lever l'ambiguïté des relations d'imbrication en utilisant un algorithme global au graphe basé sur une propagation à partir de la région infinie. De plus, un graphe dual impose de maintenir en parallèle deux structures et est difficilement extensible en dimension supérieure.

1.3.3 Cartes combinatoires

Les relations d'adjacence et d'imbrication entre les régions d'une partition constituent la *description topologique*⁵ d'une image dans la mesure où ces relations ne sont pas liées à des contraintes géométriques comme la forme des frontières de ces régions.

Les premiers travaux définissant le concept de carte proviennent d'Edmonds introduisant les premiers modèles relationnels avec les graphes planaires [Edm60]. De nombreux travaux ont ensuite développé cette structure de données comme [Tut63, Jac70, Cor75] pour décrire les relations topologiques

5. Nous parlons également de *relations topologiques*.

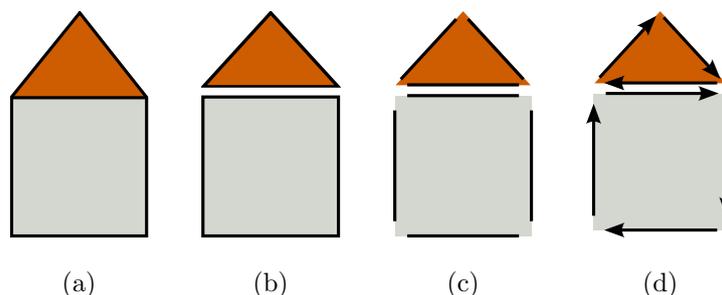


FIGURE 1.17 – Construction de l'ensemble \mathcal{B} des brins d'une carte combinatoire en dimension 2 par décompositions successives. (a) Image ; (b) Décomposition des régions ; (c) Décomposition des arêtes ; (d) Ensemble des brins.

entre les régions. Ces travaux ont ainsi conduit au modèle de carte combinatoire 2D.

En dimension n , la définition du modèle proposée par Lienhardt permet de représenter les quasi-variétés orientables (Section 1.1.2) de dimension n [Lie89]. Une généralisation des cartes combinatoires a également été proposée par Lienhardt afin de représenter les quasi-variétés en dimension n , orientables ou non [Lie91, Lie94].

Contrairement à la représentation interpixel (Section 1.2.3), une carte combinatoire en dimension 2 (ou 2-carte) représente les relations d'adjacence entre les régions. Par rapport à un graphe d'adjacence des régions (Section 1.3.1), une 2-carte permet de représenter les multi-adjacences. En revanche, les relations d'imbrication ne sont pas directement représentées dans une carte combinatoire. Il est néanmoins possible de les retrouver au moyen d'un algorithme global partant de la région infinie ou en utilisant la géométrie.

Ensemble de brins

Les cartes combinatoires introduisent des éléments abstraits appelés *brins* pour représenter les frontières des régions. Intuitivement, en dimension 2, un brin correspond à un segment orienté ou à une demi-arête [Wei85]. À chaque brin est associé un unique nœud, un unique segment et une unique région. L'ensemble \mathcal{B} des brins qui composent une 2-carte peut se concevoir comme l'ensemble des éléments de base résultant d'une décomposition des objets d'une image. La Figure 1.17 illustre les différentes étapes de cette décomposition : l'ensemble \mathcal{B} est obtenu après décomposition des régions et des arêtes d'une image. Chaque brin est représenté par une flèche indiquant l'orientation du segment orienté associé.

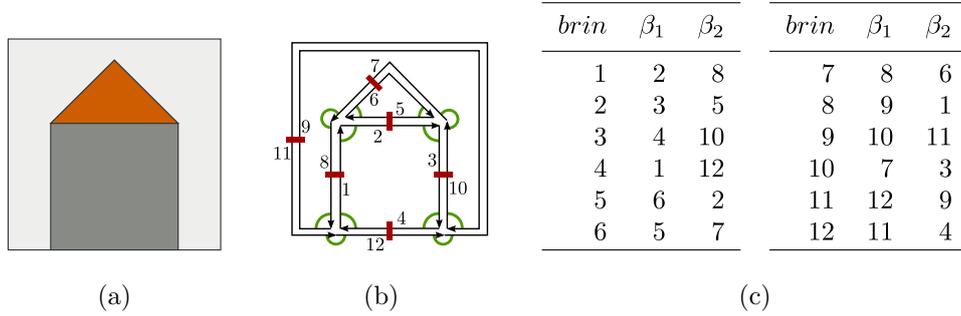


FIGURE 1.18 – Cartes combinatoires : un modèle topologique pour la représentation des multi-adjacences entre les régions d'une image. (a) Image; (b) 2-carte : les flèches représentent les brins (segments orientés), les opérateurs β_1 et β_2 sont respectivement représentés par des arcs de cercle et des segments; (c) Relations explicites des β_1 et β_2 .

Opérateurs β_1 et β_2

Les relations d'adjacence entre les régions sont décrites par deux opérateurs β_1 et β_2 s'appliquant de \mathcal{B} dans \mathcal{B} . Une 2-carte s'apparente à un graphe planaire où les deux opérateurs β_i définissent les relations entre les demi-arêtes. En pratique, β_1 est une permutation qui génère une orientation sur le bord d'une face en imposant un ordre cyclique anti-trigonométrique⁶ sur l'ensemble des segments orientés définissant une face. L'involution β_2 met en relation deux brins appartenant à une même arête, représentant une relation d'adjacence entre deux régions distinctes. Pour des raisons pratiques, nous introduisons l'opérateur β_0 tel que $\beta_0 = \beta_1^{-1}$. Deux brins b et b' vérifiant $b = \beta_i(b')$ sont dits *cousus* par β_i ou *i-cousus*. Enfin, la composition des opérateurs β_i et β_j est notée par $\beta_i \circ \beta_j = \beta_{ji}$.

Définition 27 (Carte combinatoire de dimension 2). *Une carte combinatoire 2D (ou 2-carte) est un triplet $C = (\mathcal{B}, \beta_1, \beta_2)$ où :*

- \mathcal{B} est un ensemble fini de brins ;
- β_1 est une permutation⁷ sur \mathcal{B} ;
- β_2 est une involution⁸ sur \mathcal{B} .

6. Le modèle n'est pas dépendant d'une orientation précise mais nous choisissons arbitrairement cette orientation pour des raisons de cohérence entre les différentes illustrations.

7. Une *permutation* sur un ensemble E est une bijection de E sur E .

8. Une *involution* f sur un ensemble E est une bijection de E sur E telle que $f = f^{-1}$.

Cycles

Les modèles à base de carte représentent une partition par un ensemble fini d'éléments de base (les brins) et un ensemble de permutations sur ces brins. Puisque les éléments de la partition comme les sommets, les arêtes et les faces ne sont pas directement représentés (par exemple, une arête est composée de deux brins), la notion de cycle permet de caractériser des sous-ensembles de brins représentant les objets de la partition comme les sommets, les arêtes et les faces. En 2D, un *cycle* correspond en pratique à une séquence de brins ordonnée et associée à une permutation (Définition 28).

Définition 28 (Cycle). *Soit \mathcal{B} un ensemble fini de brins et π une permutation. Le π -cycle d'un brin b est défini comme la restriction de π à l'ensemble des images de b par les itérations successives de π . Le π -cycle d'un brin b est lui-même une permutation notée $\pi^*(b) = (b_0, \dots, b_{n-1})$ avec :*

- $\forall i \in \{0, \dots, n-1\}, b_i = \pi^i(b)$ et $b_i \neq b_0$;
- $\pi^n(b) = b$.

Pour une carte combinatoire $C = (\mathcal{B}, \beta_1, \beta_2)$, chaque brin appartient à une face et donc à un unique cycle de β_1 caractérisant cette face. De la même façon, un brin appartient à un unique cycle de β_2 caractérisant l'arête correspondante. Enfin, les sommets correspondent aux cycles de $\beta_1 \circ \beta_2$. Par exemple, dans la Figure 1.18(b) :

- le cycle $(\beta_1 \circ \beta_2)^*(2) = (2, 6, 8)$ définit l'ensemble des brins incidents au sommet résultant de l'intersection des arêtes $\{2, 5\}$, $\{6, 7\}$, $\{8, 1\}$;
- le cycle $\beta_2^*(2) = (2, 5)$ décrit l'arête séparant la face carrée et la face triangulaire ;
- le cycle $\beta_1^*(1) = (1, 2, 3, 4)$ décrit la face carrée.

La notion de cycle se généralise à un ensemble de permutations par la notion d'*orbite* [Lie94]. Nous nous en servons pour caractériser certains sous-ensembles de brins à partir des opérateurs β_i (Définition 29).

Définition 29 (Orbite). *Soit $\Phi = \{\pi_1, \dots, \pi_k\}$ un ensemble fini de permutations sur \mathcal{B} . Le groupe de permutations généré par Φ est noté $\langle \Phi \rangle$. Il s'agit de l'ensemble des permutations obtenues par toutes compositions des permutations contenues dans Φ et de leurs inverses. L'orbite d'un brin b par rapport à Φ est définie par $\langle \Phi \rangle(b) = \{\pi(b) \mid \pi \in \langle \Phi \rangle\}$.*

Par exemple, l'orbite composante connexe $\langle \beta_1, \beta_2 \rangle(b)$ d'un brin b désigne l'ensemble des brins appartenant à la composante connexe de b . Notons qu'une orbite restreinte à une seule permutation correspond à l'espace de définition du cycle correspondant. Par définition, un brin b appartient simultanément à un

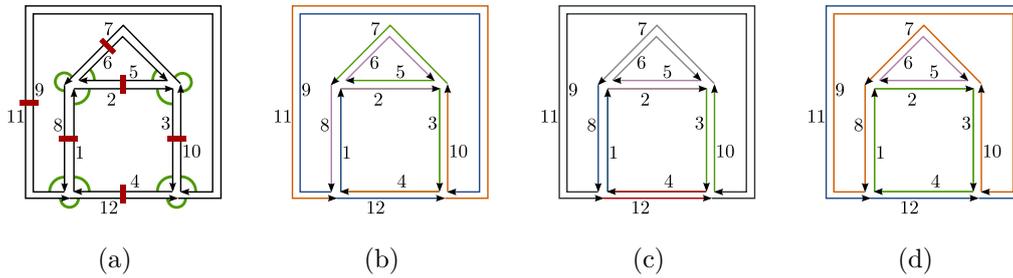


FIGURE 1.19 – Représentation des sommets, arêtes et faces dans une carte combinatoire par des cycles. (a) Image ; (b) Caractérisation des sommets par les cycles $(\beta_1 \circ \beta_2)^*$. Nous pouvons distinguer quatre sommets : $(2, 6, 8)$, $(3, 7, 5)$, $(4, 11, 10)$, $(1, 9, 12)$; (c) Caractérisation des arêtes par les cycles β_2^* . Nous pouvons distinguer six arêtes : $(1, 8)$, $(2, 5)$, $(3, 10)$, $(4, 12)$, $(6, 7)$, $(9, 11)$; (d) Caractérisation des faces par les cycles β_1^* . Nous pouvons distinguer quatre faces : la face triangulaire $(5, 6)$, carrée $(1, 2, 3, 4)$, le fond $(7, 8, 9, 10)$ et la face infinie $(11, 12)$.

seul cycle et une seule orbite définie par β_1 , β_2 et $\beta_1 \circ \beta_2$. De façon équivalente, un brin appartient à un seul sommet (orbite $\langle \beta_1 \circ \beta_2 \rangle$), une seule arête (orbite $\langle \beta_2 \rangle$) et une seule face (orbite $\langle \beta_1 \rangle$). La Figure 1.19 synthétise l'ensemble des cycles de la carte combinatoire proposée Figure 1.19(a). Les cycles correspondant aux sommets, aux arêtes et aux faces sont respectivement représentés dans la Figure 1.19(b), dans la Figure 1.19(c) et dans la Figure 1.19(d).

Carte duale

Il est également intéressant de définir le dual d'une carte combinatoire pour faire l'analogie avec le modèle de graphes duaux. Ainsi, le dual d'une carte correspond à une carte combinatoire composée du même ensemble de brins et dont l'involution et la permutation sont définies à partir du β_1 et β_2 de la carte primale.

Définition 30 (Dual d'une carte combinatoire). Soit $C = (\mathcal{B}, \beta_1, \beta_2)$ une carte combinatoire. Le dual de C , noté \overline{C} correspond au triplet $\overline{C} = (\mathcal{B}, \beta_1 \circ \beta_2, \beta_2)$.

Nous pouvons remarquer que la propriété $\overline{\overline{C}} = C$ est bien respectée. En effet, comme β_2 est une involution, $\beta_1 \circ \beta_2 \circ \beta_2 = \beta_1$.

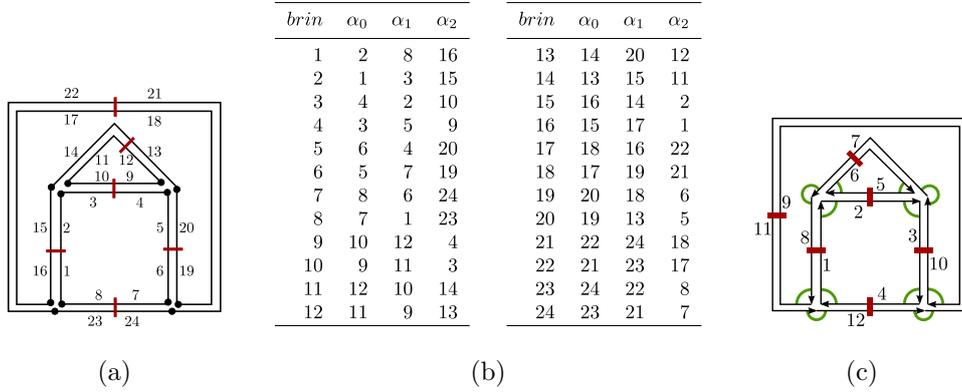


FIGURE 1.20 – (a) Cartes généralisées : deux brins liés par α_0 sont dessinés consécutifs et alignés, deux brins liés par α_1 partagent un même disque, deux brins liés par α_2 sont dessinés parallèles; (b) Relations explicites des trois involutions α_0 , α_1 et α_2 ; (c) Carte combinatoire correspondante.

Cartes généralisées

Le modèle de *cartes généralisées* propose un formalisme homogène en dimension n de façon à représenter les quasi-variétés, orientables ou non, avec ou sans bord [Lie89, Lie91, Lie94]. En dimension 2, une approche intuitive de la construction d’une carte généralisée reprend le principe de construction d’une carte combinatoire par décompositions successives (Section 1.17) et rajoute une étape de décomposition d’une arête en ses deux sommets extrémités. De la sorte, le parcours d’une face ne nécessite plus l’utilisation d’une permutation. Une carte généralisée utilise en effet trois involutions α_0 , α_1 et α_2 telles que :

- α_0 lie deux brins appartenant à la même face et à la même arête;
- α_1 lie deux brins appartenant à la même face et au même sommet;
- α_2 lie deux brins appartenant à la même arête et au même sommet.

La Figure 1.20 réutilise la même image de test utilisée Figure 1.18 pour les cartes combinatoires. Ainsi, nous pouvons comparer sur le même exemple le modèle de carte combinatoire (Figure 1.20(c)) et celui de la carte généralisée correspondante (Figure 1.20(a)). Dans cet exemple, l’involution α_0 relie deux brins partageant un même petit segment perpendiculaire à la direction de l’arête. Ces deux brins sont dessinés de façon consecutive et alignée (par exemple, $\alpha_0(1) = 2$). L’involution α_1 met en correspondance deux brins partageant un même disque (par exemple, $\alpha_1(1) = 8$). L’involution α_2 s’établit entre deux brins parallèles partageant un même petit segment perpendiculaire (par exemple, $\alpha_2(1) = 16$). L’ensemble des relations α_i est donné par le

tableau Figure 1.20(b). Lienhardt propose la définition suivante pour les cartes généralisées en dimension n [Lie91] :

Définition 31 (Carte généralisée). *Soit $n \in \mathbb{N}$. Une n -carte généralisée, ou nG -carte, est une algèbre $G = (\mathcal{B}, \alpha_0, \dots, \alpha_n)$ où :*

- \mathcal{B} est un ensemble fini de brins ;
- $\forall i, 0 \leq i \leq n, \alpha_i$ est une involution sur \mathcal{B} ;
- $\forall i, j, 0 \leq i < i + 2 \leq j \leq n, \alpha_i \circ \alpha_j$ est une involution sur \mathcal{B} .

L'avantage des cartes généralisées vient de son formalisme qui est homogène quelle que soit la dimension considérée. Toutefois, lorsque l'application visée est restreinte à la représentation d'images de dimension 2, le modèle de carte combinatoire est à privilégier du point de vue de l'implantation. En effet, les deux modèles sont équivalents au niveau des relations topologiques qu'ils représentent mais le modèle de carte généralisée requiert deux fois plus de brins et une involution supplémentaire.

1.3.4 Cartes topologiques

Une carte topologique [Fio95, Bru96, Dam01, BDM03, DBF04] combine trois modèles distincts pour représenter complètement une partition :

- une carte combinatoire pour représenter les relations d'adjacence entre les régions ;
- un modèle géométrique pour le codage de la forme des frontières des régions ;
- une structure de données pour les relations d'imbrication entre les régions.

Ces trois modèles sont illustrés dans la Figure 1.21 et détaillés ci-dessous.

Carte combinatoire

Comme illustré dans la Figure 1.21(b), une 2-carte décrit les relations d'adjacence entre les régions grâce aux opérateurs β_1 et β_2 . La carte combinatoire d'une carte topologique est *minimale* en nombre d'éléments afin que chaque arête de la carte corresponde exactement à une frontière dans l'image. En pratique, une carte minimale ne contient aucun sommet supprimable de degré 1 ou 2. Afin de lier les modèles géométriques et topologiques, les brins et les régions sont reliés entre eux : chaque brin connaît sa région d'appartenance et une région connaît un de ses brins, appelé brin *représentant* et noté ($rep(r)$).

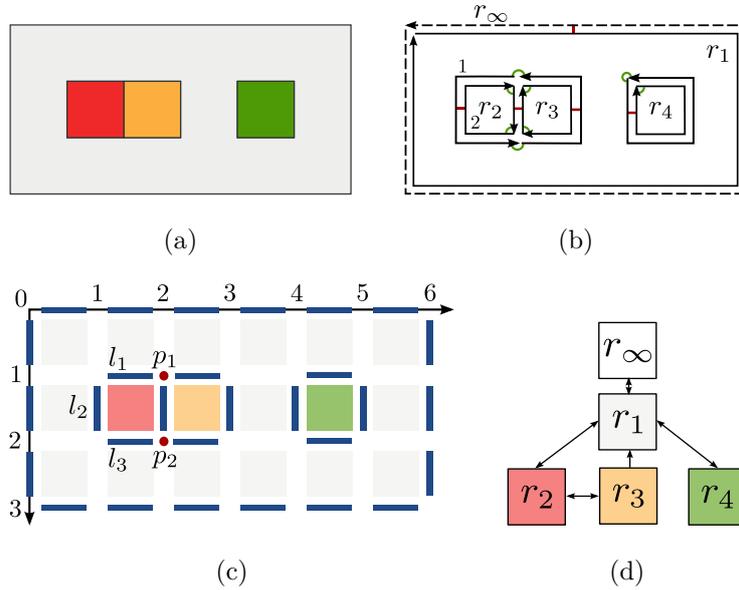


FIGURE 1.21 – Cartes topologiques : trois modèles complémentaires pour la représentation de partitions. (a) Partition; (b) Carte combinatoire pour les relations d’adjacence. Les brins en pointillés sont ceux de la région infinie; (c) Matrice d’éléments interpixel pour la géométrie : pointels et lignels sont représentés par des cercles et des segments en gras; (d) Arbre des régions pour les relations d’imbrication.

Frontières des régions

Toutes les cellules d’une 2-carte sont associées à leurs éléments géométriques correspondants dans la matrice des éléments interpixel. La notion de *plongement* est employée lorsqu’une information géométrique est associée à un élément topologique. Considérons un brin b appartenant à un sommet s , une arête a et une face f :

- le cycle $(\beta_1 \circ \beta_2)^*(b)$ correspond au sommet s . Puisque le plongement de chaque nœud correspond à un pointel, nous définissons cette association par une fonction *pointel* de \mathcal{B} dans l’ensemble des labels des pointels. Nous appelons p le premier pointel d’un brin b si $pointel(b) = p$;
- le cycle $\beta_2^*(b)$ correspond à l’arête a associée à un segment non-orienté et chacun des deux brins du cycle β_2^* représente une des deux orientations. Le plongement d’un brin est donc un segment orienté dont le premier lignel (*resp.* pointel) est noté *lignel*(b) (*resp.* *pointel*(b)).

- le cycle $\beta_1^*(b)$ correspond à la face f . Cette association est décrite par une fonction *région* de \mathcal{B} dans l'ensemble des labels de régions. Nous appelons r la région d'un brin b si $région(b) = r$.

Par exemple, dans la Figure 1.21, la région infinie r_∞ contient un unique brin représenté en pointillés. Le plongement du brin 1 est le segment orienté représenté par la séquence (p_1, l_1, l_2, l_3) ; $lignél(1) = l_1$, $pointél(1) = p_1$; $degré(p_1) = degré(p_2) = 3$. Dans le cadre d'une représentation 4-connexe de l'image, chaque pointel est incident à quatre lignels que nous différencions par quatre valeurs $\{x, -x, y, -y\}$ indiquant la position du lignal par rapport au pointel. En pratique, la mise en relation entre un brin et son plongement est effectuée par une fonction *doublet* définissant l'orientation initiale du segment orienté associé au brin (Définition 32).

Définition 32 (Doublet d'un brin). *Le doublet d'un brin b , noté $doublet(b)$, est un triplet (i, j, l) où :*

- (i, j) sont les coordonnées entières de $pointél(b)$;
- $l \in \{x, -x, y, -y\}$ désigne la direction de $lignél(b)$ par rapport aux coordonnées du pointel (i, j) .

À titre d'exemple, les doublets des brins 1 et 2 de la Figure 1.21(b) valent respectivement $(6, 4, -x)$ et $(1, 4, -y)$ dans la Figure 1.21(c). Nous appelons *arête élémentaire* une arête dont le plongement est un unique lignal. Une arête élémentaire est ainsi codée par deux *brins élémentaires*. Enfin, une *région élémentaire* est une région composée de 4 brins et encodant un unique pixel de l'image.

Arbre des régions

L'arbre des régions décrit les relations d'imbrication entre les régions. Il est nécessaire d'utiliser une structure externe car ces relations ne sont pas directement accessibles dans le modèle de carte combinatoire (Section 1.3.3) : lorsqu'une région est imbriquée dans une autre, les opérateurs β_1 et β_2 ne permettent pas de passer d'un brin de la région englobante à un brin de la région imbriquée. Notre modèle de carte topologique utilise un arbre des régions (appelé *arbre d'imbrication*) pour représenter ces relations. Chaque nœud de l'arbre est une région et les arêtes indiquent une relation d'imbrication. Notons que la racine de l'arbre correspond à la région infinie (Section 1.1.4). De cette façon, chaque région r connaît sa région englobante (notée $père(r)$). Inversement, les régions imbriquées dans r sont notées $fil(s(r))$.

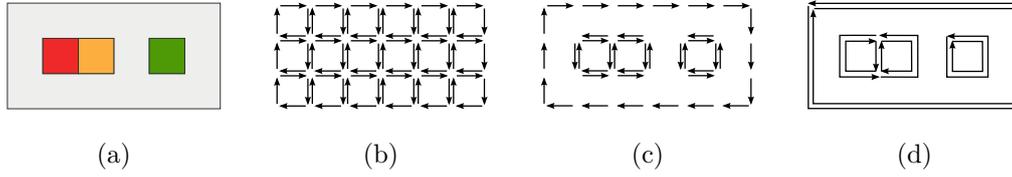


FIGURE 1.22 – Simplifications progressives d’une carte combinatoire. (a) Partition; (b) Niveau 0 : carte complète; (c) Niveau 1 : carte ligned; (d) Niveau 2 : carte topologique.

Terminologie

Un sous-ensemble de brins \mathcal{B}' de \mathcal{B} est dit *symétrique* si $\forall b \in \mathcal{B}', \beta_2(b) \in \mathcal{B}'$. Soit une arête a et b un brin incident à a . Nous qualifions a de *boucle* si $\beta_1(b) = b$ et d’*arête pendante* si $\beta_1(b) = \beta_2(b)$. Nous différencions également les notions de bord externe et de bord interne à une région : un bord interne d’une région r correspond à une frontière entre r et une région imbriquée dans r . Toute frontière de r n’étant pas un bord interne est un bord externe. Par convention, le brin représentant d’une région est toujours choisi sur son bord externe.

Niveaux de simplification

La carte topologique est définie au travers de trois niveaux de représentation, allant du plus complet au plus simple (en termes de nombre de brins et de régions) par simplifications successives :

- le niveau 0 ou *carte complète* (Figure 1.22(b)). Il s’agit du niveau maximal de représentation : à chaque pixel correspond une région composée de quatre brins. En comptant la région infinie, la carte associée à une image de taille $(m \times n)$ contient $(m \times n) + 1$ faces;
- le niveau 1 ou *carte ligned* (Figure 1.22(c)). Le premier niveau de simplification s’obtient par suppression des arêtes séparant deux régions adjacentes dont les pixels ont le même label. Le résultat obtenu représente les frontières par des arêtes composées de brins dont le plongement est un unique ligned;
- le niveau 2 ou *carte topologique* (Figure 1.22(d)) s’obtient après suppression des sommets supprimables de degré 2. La carte obtenue est alors *minimale* en nombre de brins. Cette propriété de minimalité est importante pour la généralisation du modèle en dimension supérieure et pour diminuer la complexité des opérations (garantissant l’efficacité du modèle).

Ces niveaux de représentation présentent une définition constructive du modèle de carte topologique. En pratique, ils sont parfois utilisés temporairement et/ou localement lors de certains traitements ou opérations. Le modèle de carte topologique est un modèle permettant d'implanter efficacement des opérations de nature topologique et géométrique.

Différentes implantations

Différentes implantations des trois modèles composant une carte topologique ont été proposées [Fio95, MK05, MAM04]. Nous pouvons notamment distinguer :

- Le *graphe topologique des frontières* (ou TGF) est un modèle dont la particularité est d'intégrer explicitement la notion de région [Fio95, Fio96]. La structure de données relative aux imbrications est ainsi directement codée dans le modèle de carte combinatoire (Section 1.3.3) : chaque région conserve la liste de ses frontières en différenciant bords extérieurs et bords intérieurs (relations d'imbrication). Un algorithme d'extraction optimal ainsi qu'une opération de fusion de régions [AFG99] ont été proposés pour le modèle.
- Le modèle de GeoMap [MK05] représente par une même structure la topologie et la géométrie à l'aide d'un modèle dérivé de cartes combinatoires, les XMap [Kot02]. Ce modèle est utilisé en segmentation et propose plusieurs outils interactifs de haut niveau basés sur le contour des régions. Une implantation différente du modèle géométrique, basée sur une subdivision sous-pixel permet essentiellement d'obtenir des résultats géométriques plus précis [MK06].
- Une implantation du modèle de carte topologique, proposée par Marchadier *et al.* [MAM04], a la particularité de ne pas utiliser les complexes cellulaires abstraits et les éléments interpixel pour représenter la géométrie. La construction de la carte est en effet basée sur une grille carrée obtenue depuis une transformation par ligne de partage des eaux.
- Une définition équivalente des cartes combinatoires proposées par Cori [Cor75] préfère utiliser les opérateurs $\sigma = \beta_1 \circ \beta_2$ et $\alpha = \beta_2$ pour désigner respectivement la permutation sur les brins d'un même sommet et sur les brins d'une même arête [Dom92, Bru96]. Ce modèle code les relations d'imbrication des régions par des relations père/fils entre régions englobantes et régions imbriquées. Il permet également de mettre à jour efficacement la carte par des opérations incrémentales.

1.4. Structures hiérarchiques

TABLE 1.1 – Tableau récapitulatif des différentes associations et terminologies entre les éléments des trois modèles composant une carte topologique.

frontières	plongement	2-carte	cycle
interpixel	géométrique		
nœud	pointel	sommet	$(\beta_1 \circ \beta_2)^*$
segment	séquence alternée de pointels et lignels	arête	β_2^*
région	ensemble ouvert 4-connexe de pixels	face	β_1^*
segment orienté	chemin frontière orienté	brin	$Id.$

Bilan

Le modèle de carte topologique présente plusieurs avantages. Tout d’abord, la carte combinatoire permet de représenter les relations d’adjacence et de multi-adjacence entre les régions à l’aide d’un ensemble minimal de brins et de deux permutations β_1 et β_2 . Les relations d’imbrication sont également accessibles grâce à l’arbre d’imbrication des régions. Le modèle représente donc toutes les relations topologiques d’une partition. De plus, le plongement des brins dans une structure géométrique comme la matrice des éléments interpixel permet d’associer facilement le modèle topologique (carte combinatoire) à la géométrie de la partition. Étant donné un brin b , il est par exemple possible de parcourir géométriquement le chemin frontière orienté qui lui est associé. Enfin, les cycles permettent de retrouver efficacement les éléments de la partition comme les sommets, les arêtes et les faces. Le modèle de carte topologique offre ainsi une représentation complète de la partition d’une image. Le modèle est de plus généralisable en dimension supérieure (cartes 3D, cartes généralisées).

Un récapitulatif des associations et correspondances entre les représentations géométriques et topologiques est proposé Tableau 1.1 permettant de mettre en évidence les liens entre les éléments d’une partition et leur notation dans les différentes structures.

1.4 Structures hiérarchiques

Les modèles hiérarchiques, également appelés *pyramides*, sont apparus comme une solution pour mimer certains phénomènes psycho-visuels obser-

vés. En effet, il s'avère parfois pertinent de conserver plusieurs niveaux de représentation d'un même objet, le niveau de détail le plus important n'étant pas forcément le plus adapté lors d'un traitement particulier. Un modèle hiérarchique permet ainsi d'optimiser la recherche de détails après avoir localisé au préalable son emplacement potentiel à un niveau de segmentation plus grossier. Il existe de nombreux cas où les modèles hiérarchiques sont adaptés à l'exploitation d'images, en particulier si les données présentent un facteur d'échelle élevé comme dans le cas d'images satellite, astronomiques et microscopiques.

Dans le cadre d'utilisation des modèles hiérarchiques, il convient de différencier les notions de résolution et d'échelle. En effet, ces deux notions sont proches mais distinctes :

- la *résolution* provient de l'observation et se caractérise par une mesure. Dans le cas d'une image naturelle, cette mesure correspond à la surface réelle représentée par un pixel ;
- l'*échelle* représente un niveau de détail dans un modèle. Une échelle faible correspond ainsi à un niveau de détail peu élevé. Par exemple, une analyse multi-échelles comme les approches échelle/espace (*scale-space* en anglais) considère un ensemble d'images dont la perception est de plus en plus grossière bien que la résolution reste constante.

La confusion entre ces deux termes vient du fait que certains modèles hiérarchiques combinent intrinsèquement les deux notions. En effet, considérons par exemple le cas d'une image multi-résolution, le niveau de détail qui va être représenté par le modèle est généralement moindre à faible résolution qu'à forte résolution.

Parmi les procédés de construction des modèles hiérarchiques, nous pouvons distinguer deux catégories : les méthodes ascendantes (*bottom-up* en anglais) et descendantes (*top-down* en anglais). Les deux approches sont intéressantes et dépendent essentiellement de l'application visée et des opérations à effectuer sur la structure de données utilisée. Les modèles dont le processus de construction est ascendant définissent une partition maximale (au sens d'une sur-segmentation) comme premier niveau de la pyramide. Les niveaux suivants se construisent par simplification du niveau précédent jusqu'à obtenir la segmentation souhaitée.

Par opposition aux constructions ascendantes, les méthodes descendantes (*top-down* en anglais) partent d'une segmentation grossière composée de peu de régions. Chaque niveau est un raffinement du précédent par application d'opérations de découpe. Il y a deux avantages à privilégier une approche descendante par rapport à une construction ascendante. Tout d'abord, le coût mémoire du modèle est moindre dans la mesure où pour atteindre une

segmentation donnée, il n'est pas forcément nécessaire de disposer du niveau de partition le plus détaillé (donc contenant le plus de régions). Cela s'avère donc intéressant pour les applications imposant des contraintes mémoire. Il est également possible d'utiliser la notion de *focus d'attention* qui permet de définir un objet à faible échelle avant de le raffiner à des échelles plus élevées. En effet, certaines informations sont plus faciles à extraire à faible échelle (donc dans les premiers niveaux du modèle) comme par exemple, l'aspect général d'une forme ou une zone de couleur prédominante. La même information est plus difficile à détecter à forte résolution car l'information pertinente est masquée par le niveau de détail plus important.

1.4.1 Pyramides régulières

Quadrees

Les quadrees, également appelés arbres quaternaires définissent une partition d'une image 2D en appliquant récursivement une opération de découpe en quatre basée sur un motif géométrique habituellement carré [HP74, FB74, DRS80, Sam80, Sam84]. Ainsi chaque parent de la hiérarchie correspond à quatre fils au niveau suivant. Le processus de construction est descendant : le premier niveau est une unique région représentant la totalité de l'image. Un niveau se construit d'après le précédent en découpant les régions qui ne satisfont pas un critère d'homogénéité donné : chaque niveau représente ainsi l'ensemble des subdivisions des blocs non homogènes du niveau précédent. Le processus est réitéré jusqu'à obtention d'un niveau où tous les blocs sont homogènes.

La Figure 1.23 illustre le modèle en présentant le processus de construction d'un quadtree associé à une image binaire de taille 8×8 . Dans cet exemple, une région est considérée homogène si et seulement si tous ses pixels ont la même valeur. L'arbre est composé de blocs homogènes (blancs ou noirs) et non homogènes (gris). Chaque bloc non homogène est divisé au niveau suivant.

Ce partitionnement d'une image est efficace mais induit plusieurs inconvénients majeurs. Les dimensions de l'image doivent être sous la forme $2^n \times 2^n$ pour ne pas engendrer de problèmes de recouvrement. De plus, le partitionnement ne tient pas compte de la géométrie des régions : une région peut être répartie sur des blocs appartenant à différents niveaux de la hiérarchie. Il faut en conséquence déterminer les adjacences entre les blocs de la hiérarchie pour retrouver une région donnée.

Différentes extensions et variantes des modèles de quadrees ont été proposées. Par exemple, les pyramides d'Hartman et Tanimoto [HT84] utilisent un motif géométrique de base hexagonal. Certains modèles comme les arbres

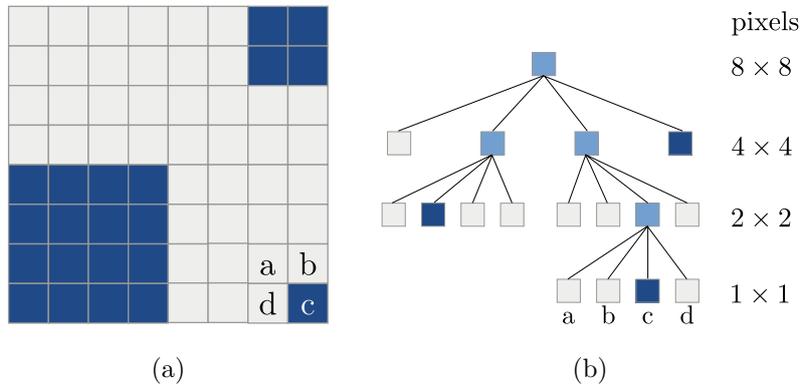


FIGURE 1.23 – Arbres quaternaires (quadrees). (a) Image binaire dont les pixels ne peuvent avoir que deux valeurs (noir ou blanc); (b) Arbre quaternaire associé composé de blocs homogènes (noirs ou blancs) ou inhomogènes (gris). Chaque bloc non homogène est divisé au niveau suivant.

de partitionnement binaire [FKN80] réutilisent le même schéma de subdivision récursive.

Pyramides d'images

Les pyramides régulières réutilisent le principe de construction des quadrees mais en adoptant une approche ascendante et sont définies comme une séquence d'images dont la taille décroît de façon exponentielle, chaque image définissant un *niveau* de la pyramide [TP75, JR94, BCR90, RS98, RS88, BA83, OCD95, CLR95]. La base de la pyramide (ou niveau 0) constitue le premier niveau et correspond à l'image dont la taille est la plus importante. Le sommet de la pyramide est le dernier niveau (niveau n), généralement constitué d'une image contenant un unique pixel. Les relations hiérarchiques sont établies à partir des notions de *fenêtre de réduction* (Définition 33) et de *champ récepteur* (Définition 34).

Définition 33 (Fenêtre de réduction). *Soit une pyramide régulière composée de $n + 1$ niveaux. La fenêtre de réduction de taille $m \times m$ d'un pixel p d'un niveau k ($0 < k \leq n$) est une fenêtre de taille $m \times m$ définie au niveau $k - 1$ et correspondant à l'ensemble des pixels codés par p au niveau k . Cette relation établit une hiérarchie, codée par $\mathcal{FR}(p)$, telle que :*

$$\forall p' \in \mathcal{FR}(p), p \text{ est le parent de } p' \text{ et } p' \text{ est un fils de } p.$$

En d'autres termes, chaque pixel est le parent de tous les pixels appartenant à sa fenêtre de réduction; chaque pixel d'une fenêtre de réduction est le fils

d'au moins un parent. Cette relation parent-enfant peut être itérée sur tous les niveaux jusqu'à la base, définissant la notion de *champ récepteur*.

Définition 34 (Champ récepteur). *Soit une pyramide régulière composée de $n + 1$ niveaux (I^0, \dots, I^n). Le champ récepteur d'un pixel p d'un niveau k ($0 < k \leq n$) est un ensemble de pixels du niveau 0, noté $\mathcal{CR}(p)$, vérifiant :*

$$\forall p' \in \mathcal{CR}(p), \forall i, 0 < i \leq k, \exists (p_0, \dots, p_k) \in I^0 \times \dots \times I^k \text{ avec}$$

- $p_0 = p'$;
- p_i est le parent de p_{i-1} ;
- $p_k = p$.

Par conséquent, le champ récepteur d'un pixel est l'ensemble de ses descendants appartenant à la base de la pyramide. Nous introduisons ensuite r comme étant le ratio (fixe) entre la taille de deux images consécutives de la pyramide. La quantité $Q = (m \times m)/r$ permet de différencier trois sous-catégories de pyramides illustrées dans la Figure 1.24 :

- $Q < 1$: pyramides trouées et non recouvrantes : certains pixels n'ont pas de parent [Kro97] (Figure 1.24(a)) ;
- $Q = 1$: pyramides non trouées et non recouvrantes : chaque pixel a exactement un parent (Figure 1.24(b)) ;
- $Q > 1$: pyramides recouvrantes : la fenêtre étant plus importante que le ratio entre deux niveaux de la pyramide, un chevauchement des fenêtres conduit à ce que certains pixels aient plusieurs parents potentiels (Figure 1.24(c)).

Discussion sur les structures régulières

Les pyramides régulières présentent plusieurs avantages énoncés par Bister *et al.* [BCR90] :

- les niveaux réduits permettent de contenir l'influence du bruit de l'image de base ;
- certaines propriétés globales peuvent être ramenées à des propriétés locales ;
- les temps de calcul pour certaines opérations comme la détection de régions sont moindres quand il est possible de les appliquer sur les niveaux réduits.

De nombreuses applications de ces structures régulières en segmentation, compression ou réduction de bruit sont fondées sur ces modèles hiérarchiques réguliers avec des justifications psycho-visuelles. Cependant, les pyramides régulières sont limitées par la taille constante de la fenêtre de réduction et

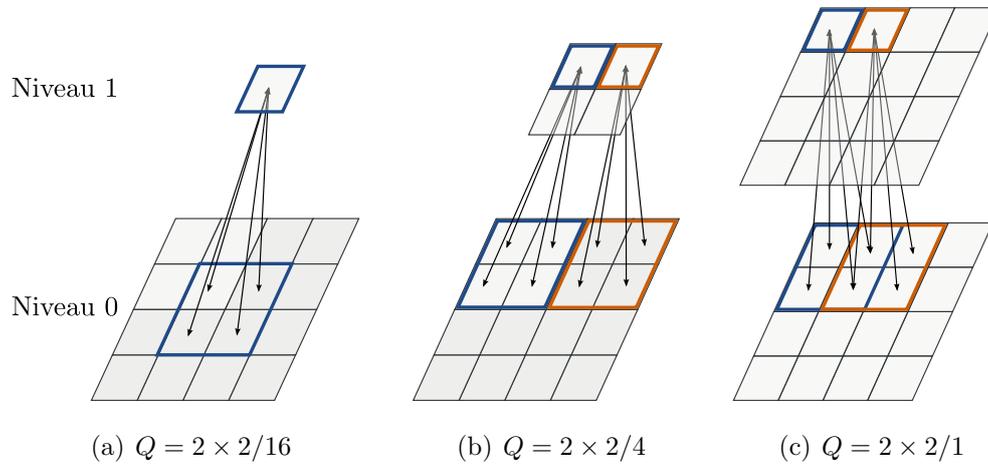


FIGURE 1.24 – Trois exemples de pyramides régulières avec une fenêtre de réduction fixée à 2×2 . Les flèches représentent les relations père-fils. (a) Pyramide trouée et non recouvrante ($Q < 1$) : les pixels du bord n'ont pas de parent ; (b) Pyramide non trouée et non recouvrante ($Q = 1$) : chaque pixel a exactement un parent ; (c) Pyramide recouvrante ($Q > 1$) : certains pixels ont plusieurs parents potentiels.

le ratio fixe imposé entre deux niveaux consécutifs. Le modèle présente donc une certaine rigidité qui peut être mise en évidence par de légères variations des données originales (décalages, rotations). De plus, il est difficile de représenter des régions dites *allongées* privilégiant fortement une direction [BCR90].

1.4.2 Pyramides de graphes simples

Les limitations des pyramides régulières évoquées dans la Section 1.4.1 ont conduit Meer [Mee89] à proposer une structure hiérarchique fondée sur les graphes d'adjacence (Section 1.3.1) de façon à combiner à la fois les avantages d'un modèle hiérarchique et ceux des modèles relationnels.

Par un procédé de construction ascendant, une pyramide de graphes se construit depuis la base vers son sommet par un processus de décimation détaillé dans la Section 1.4.3. Dans une *pyramide de graphes simples*, chaque niveau est représenté par un graphe d'adjacence. La base de la pyramide correspond généralement au codage de chacun des pixels d'une image par un sommet. Chaque graphe est un couple $G^k = (\mathcal{S}^k, \mathcal{A}^k)$ où \mathcal{S}^k est l'ensemble des sommets de G^k et $\mathcal{A}^k \subset \mathcal{S}^k \times \mathcal{S}^k$ représente l'ensemble des arêtes du graphe G . L'exposant k indique le niveau dans la hiérarchie de la pyramide (0 indiquant le premier niveau, soit la base de la pyramide dans le cas d'une construction

TABLE 1.2 – Hiérarchie de la pyramide de graphes simples illustrée dans la Figure 1.25 par partitionnement de l'ensemble des sommets des différents niveaux en deux sous ensembles : les sommets survivants et non survivants.

	sommets (\mathcal{S})	survivants (\mathcal{SS})	non survivants (\mathcal{SN})
G^0	$r_1^0, r_2^0, r_3^0, r_4^0, r_5^0, r_6^0$	$r_1^0, r_2^0, r_3^0, r_6^0$	r_4^0, r_5^0
G^1	$r_1^1, r_2^1, r_3^1, r_6^1$	r_1^1, r_2^1, r_6^1	r_3^1
G^2	r_1^2, r_2^2, r_6^2		

ascendante). La construction d'un niveau $k + 1$ depuis le niveau précédent k s'effectue en sélectionnant un ensemble de *sommets survivants* noté \mathcal{SS}^k . Les sommets restants sont appelés *non survivants* et leur ensemble est noté \mathcal{SN}^k ($\mathcal{S}^k = \mathcal{SS}^k \cup \mathcal{SN}^k$). La hiérarchie s'établit en associant chaque sommet non survivant de \mathcal{SN}^k à un sommet de \mathcal{S}^{k+1} [MMR91].

Un exemple de pyramide de graphes simples est proposé dans la Figure 1.25 avec une pyramide composée de trois niveaux $G^0/G^1/G^2$. Notons r_j^i la région j du niveau i . Les ensembles \mathcal{S} , \mathcal{SS} et \mathcal{SN} pour les trois niveaux sont explicitement donnés par le Tableau 1.2. À titre d'exemple, r_2^2 est un *sommet survivant* du niveau 2. Les *fils* de r_2^2 sont r_2^1 et r_3^1 . Le *champ récepteur* du sommet r_2^2 correspond aux sommets $\{r_2^0, r_3^0, r_4^0\}$ (Figure 1.25(a)).

En conclusion, les pyramides de graphes simples sont une extension hiérarchique des graphes d'adjacence de régions (Section 1.3.1) dont les nœuds sont reliés par des relations père-fils. Par conséquent, le modèle hiérarchique hérite des mêmes inconvénients que les graphes d'adjacence, à savoir, essentiellement l'impossibilité de différencier les différents cas d'adjacence.

1.4.3 Processus de décimation

La première opération de réduction de graphe proposée provient des travaux de Meer [Mee89] et définit la création d'un graphe G^{k+1} à partir d'un niveau G^k comme un processus stochastique en trois étapes détaillées ci-après.

Première étape : sélection des sommets survivants

Deux contraintes ont été proposées par Meer [Mee89] pour que les processus locaux soient parallélisables et que le facteur de décimation soit suffisamment important pour limiter le nombre de niveaux :

- stabilité externe : $\forall s \in \mathcal{S}^k \setminus \mathcal{S}^{k+1}, \exists s' \in \mathcal{S}^{k+1} : (s, s') \in \mathcal{A}^k$;
- stabilité interne : $\forall (s, s') \in \mathcal{S}^{k+1} \times \mathcal{S}^{k+1} : (s, s') \notin \mathcal{A}^k$.

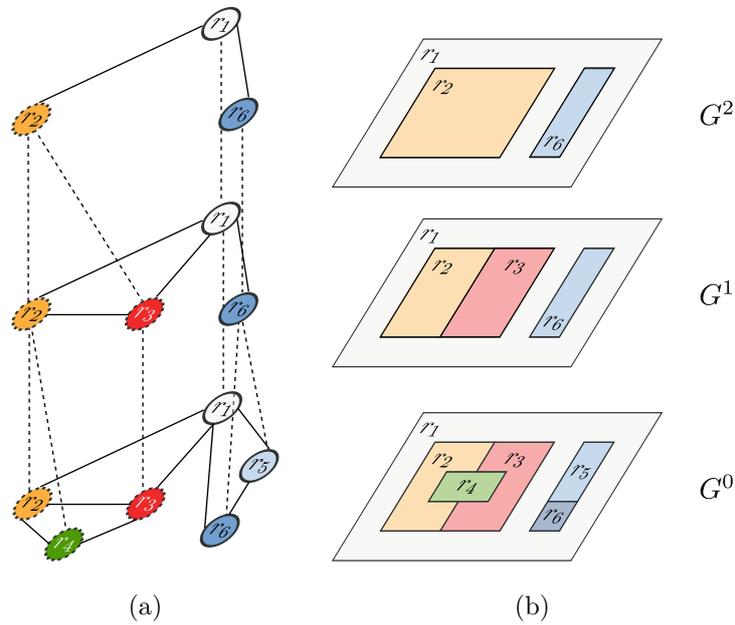


FIGURE 1.25 – Pyramides de graphes simples. (a) Graphes : les trois niveaux G^0 , G^1 , G^2 forment une pile de graphes d'adjacence successivement réduits ; (b) Pyramide des images correspondantes.

La stabilité externe impose à chaque sommet non survivant d'être adjacent à un sommet survivant par une décision locale. La propriété de stabilité interne assure que le nombre de sommets diminue rapidement d'un niveau à l'autre en interdisant à deux sommets adjacents de survivre simultanément et permet également une bonne répartition des sommets survivants.

Le processus de décimation est un processus local et itératif déterminant quels sont les sommets survivants d'un niveau. À chaque sommet est associée une valeur x_i représentant le tirage d'une variable aléatoire du sommet i (noté s_i), ainsi que deux variables booléennes p_i et q_i indiquant respectivement si s_i est un sommet survivant et si s_i peut devenir un sommet survivant lors d'une itération ultérieure. Dans ce cas, le sommet s_i est qualifié de *candidat*. Notons $\mathcal{S}(s_i)$ l'ensemble des sommets voisins de s_i . Nous désignons par \vee et \wedge les opérateurs logique *ou* et *et*. Considérons la construction du niveau $k + 1$ d'une pyramide composée de $n + 1$ niveaux ($0 \leq k < n$). Les prédicats p_i et q_i sont initialisés de façon à ce qu'un sommet soit désigné survivant s'il s'agit d'un maximum local et candidat si aucun de ses voisins n'est désigné survivant :

- $p_i^0 = 1 \Leftrightarrow x_i = \max_{s_j \in V(s_i)} \{x_j\}$;
- $q_i^0 = \bigwedge_{s_j \in V(s_i)} \bar{p}_j$.

Les itérations suivantes permettent de s'assurer qu'un sommet soit élu survivant s'il s'agit d'un maximum local de la variable aléatoire parmi ses voisins candidats. Un sommet reste candidat si aucun de ses voisins ne survit :

- $p_i^{k+1} = 1 \Leftrightarrow p_i^k = 1 \vee (q_i^k \wedge x_i = \max_{s_j \in V(s_i)} \{q_j^k x_j\})$;
- $q_i^{k+1} = \bigwedge_{s_j \in V(s_i)} \overline{p_j^k}$.

Deuxième étape : définition des relations parent-enfant

Cette étape consiste à établir les relations hiérarchiques entre les sommets de deux niveaux consécutifs de la pyramide. Un sommet i non survivant est ainsi relié à un voisin survivant que nous appelons *parent* de i . Nous pouvons remarquer que la contrainte de stabilité externe nous assure qu'il en existe au moins un. Par analogie avec la définition proposée pour les pyramides régulières, une *fenêtre de réduction* correspond à l'ensemble des fils d'un sommet survivant.

Troisième étape : construction des arêtes

Par itérations successives du processus de décimation, nous obtenons $G^{k+1} = (\mathcal{S}^{k+1}, \mathcal{A}^{k+1})$ avec :

1. $\mathcal{S}^{k+1} = \{s_i \in \mathcal{S}^k \mid p_i = 1\}$;
2. $\mathcal{A}^{k+1} = \{(s_i, s_j) \in \mathcal{S}^{k+1} \times \mathcal{S}^{k+1} \mid (s_i \neq s_j) \wedge \Delta^{k+1}(i, j) = 1\}$.

Dans (2), nous avons $\Delta^{k+1}(i, j) = 1$ si et seulement si l'une des deux propositions suivantes est vérifiée :

- $\exists s \in \text{fils}(s_i) \cup \text{fils}(s_j) \mid (s, s_i) \in \mathcal{A}^k \text{ et } (s, s_j) \in \mathcal{A}^k$;
- $\exists (s, s') \in \text{fils}(s_i) \times \text{fils}(s_j) \mid \{(s_i, s), (s, s'), (s', s_j)\} \in \mathcal{A}^k$.

Le processus est itéré jusqu'à ce que le dernier niveau soit composé d'un unique sommet. La méthode proposée par Meer [Mee89] et Montanvert [MMR91] permet ainsi de décider localement si un sommet est survivant (à l'aide d'un algorithme parallélisable) tout en garantissant une décimation suffisante pour limiter le nombre de niveaux de la pyramide. Certaines alternatives comme [Jol01, HGS+02] ont été proposées pour améliorer certains aspects.

Décimation guidée par les données

L'idée de base consiste à privilégier la réduction de certaines régions en introduisant un *opérateur d'intérêt* [Jol01]. Par exemple, un opérateur d'intérêt basé sur la variance des niveaux de gris d'une région va favoriser la réduction des régions de l'image particulièrement homogènes. Pour cela, la valeur x_i associée à un sommet lors du processus de décimation ne correspond plus

à un tirage aléatoire (pyramides *stochastiques* [Mee89]) mais à la valeur de l'opérateur d'intérêt conféré à la région (pyramides *adaptatives* [JM92]). L'aspect asynchrone de la méthode proposée par [Jol01] permet par ailleurs de modéliser des résultats psycho-visuels démontrant que la remontée de certaines informations visuelles jusqu'au cerveau se fait de façon asynchrone [TFM96]. En effet, les régions caractérisées par l'opérateur d'intérêt sont détectées plus rapidement.

Couplage maximal

Cette méthode a été proposée par Haxhimusa *et al.* [HGS⁺02] qui a observé que le degré moyen des sommets de la pyramide tend à augmenter dans les niveaux élevés de la pyramide. Ce phénomène diminue le facteur de décimation et donc augmente la taille de la pyramide. Pour corriger ce problème, la solution consiste à effectuer le processus de décimation sur la base d'un *couplage maximal*. Un couplage est un sous-ensemble d'arêtes où deux arêtes distinctes ne peuvent être adjacentes au même sommet. Le couplage est qualifié de maximal si on ne peut rajouter aucune arête au sous-ensemble sans perdre la propriété de couplage. Le processus de décimation proposé par Haxhimusa *et al.* [HGS⁺02] peut se résumer en trois étapes illustrées dans la Figure 1.26 : la construction d'un couplage maximal (Figure 1.26(a)), l'insertion de nouvelles arêtes de façon à ce que tous les sommets soient incidents à au moins un sommet (Figure 1.26(b)) et une étape de suppression d'arêtes telle que la forêt obtenue soit uniquement composée d'arbres de profondeur 1 (Figure 1.26(c)). Le résultat obtenu permet ainsi de sélectionner de façon naturelle l'ensemble des sommets survivants. En effet, pour chaque arbre obtenu, une première étape désigne une racine (le sommet survivant) telle que tous ses fils soient à une distance de 1. Les relations parent-enfant sont ensuite établies entre les nœuds restants de l'arbre (les fils) et la racine (le parent) comme illustré par la Figure 1.26(d). Cette méthode assure une meilleure stabilité du facteur de décimation qui reste ainsi approximativement égal à 2. Cette méthode a ensuite été étendue aux graphes dirigés définissant un voisinage adapté d'une arête orientée [HGK03].

1.4.4 Pyramides de graphes duaux

De la même façon que les graphes d'adjacence ont été étendus aux graphes duaux (Section 1.3.2), [CK93, WK94, KM95, Kro95] ont proposé une extension hiérarchique avec les pyramides de graphes duaux dans le but de différencier les cas d'adjacence entre les régions d'un même niveau. Chaque niveau k de la

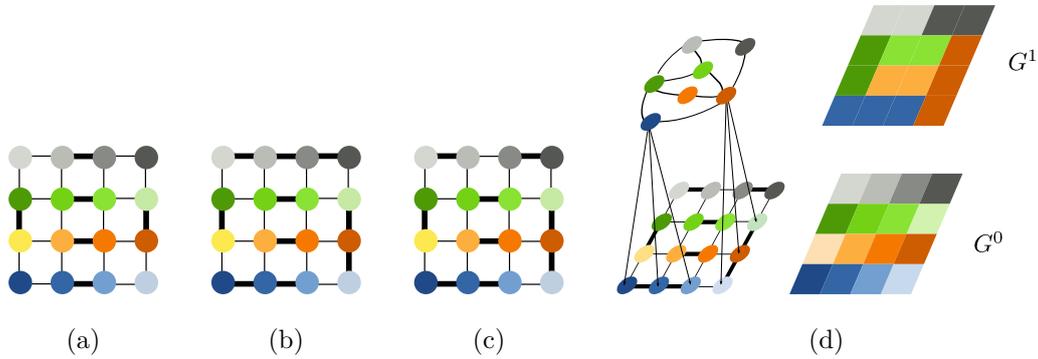


FIGURE 1.26 – Décimation par couplage maximal. (a) Choix d’un couplage maximal composé de 6 arêtes (en gras); (b) Insertion d’arêtes pour connecter tous les sommets; (c) Suppression d’arêtes pour obtenir une forêt d’arbres de profondeur 1; (d) Sélection des sommets survivants et définition des relations parent-enfant.

pyramide est donc un graphe dual (G^k, \bar{G}^k) composé d’un graphe d’adjacence (primal) $G^k = (\mathcal{S}^k, \mathcal{A}^k)$ et d’un graphe de faces (dual) $\bar{G}^k = (\bar{\mathcal{S}}^k, \bar{\mathcal{A}}^k)$. Le processus de décimation permettant de construire G^{k+1} depuis G^k s’interprète alors en termes d’opérations de *contraction* et de *suppression* d’arêtes.

La contraction d’une arête (s, s') identifie les deux sommets s et s' et enlève l’arête qui les relie. La suppression d’une arête a d’un niveau k revient à l’enlever de l’ensemble \mathcal{A}^k . Le processus de décimation peut se décomposer en quatre étapes :

- sélectionner un ensemble \mathcal{SS} de sommets survivants ;
- établir les relations parent-enfant en reliant par une arête chaque sommet non survivant à un sommet survivant ;
- contracter les arêtes construites à l’étape précédente ;
- supprimer les arêtes *redondantes* : une arête est redondante s’il s’agit d’une boucle qui n’entoure pas de sommet survivant ou s’il existe une autre arête ayant les même extrémités et n’entourant pas non plus de sommet survivant (arêtes multiples).

La première et la seconde étape sont effectuées selon le processus de décimation retenu (Section 1.4.3). La troisième étape de ce processus repose sur la définition d’un noyau de contraction tel que défini par Kropatsch [Kro97].

Définition 35 (Noyau de contraction). Soit G^k un graphe d'adjacence des régions. Un noyau de contraction sur G^k est un couple $\mathcal{K} = (\mathcal{S}^k, \mathcal{AN}^k)$ vérifiant :

- $(\mathcal{S}^k, \mathcal{AN}^k)$ est une forêt recouvrante⁹ de G^k ;
- la racine de chaque arbre de $(\mathcal{S}^k, \mathcal{AN}^k)$ appartient à \mathcal{SS}^k .

L'ensemble \mathcal{AN}^k est appelé l'ensemble des arêtes non-survivantes du niveau k . Lors de la contraction des arêtes, le nombre de sommets diminue mais aucune arête définie entre deux sommets survivants n'est supprimée. Par conséquent, le graphe obtenu est susceptible de contenir des boucles vides et des arêtes redondantes qu'il est nécessaire de supprimer.

Nous retrouvons néanmoins les inconvénients des graphes duaux dans leur version hiérarchique. Ce modèle ne permet pas de caractériser localement les relations d'imbrication et ne code pas l'orientation de la partition.

1.4.5 Pyramides combinatoires

Les pyramides de cartes combinatoires, appelées pyramides combinatoires, ont été introduites par Brun et Kropatsch [BK03a] de façon à s'affranchir des limitations induites par les pyramides de graphes en proposant une extension hiérarchique au modèle des cartes combinatoires. Le processus de construction proposé est basé sur celui des pyramides de graphes duaux (donc ascendant). Nous effectuons une analogie avec les pyramides de graphes duaux en nous appuyant sur la définition du dual d'une carte combinatoire (Définition 30) puis en redéfinissant les notions de noyaux d'opérations (Définition 36 et Définition 37). En effet, les pyramides combinatoires sont composées d'une séquence de cartes combinatoires progressivement réduites.

Définition 36 (Noyau de contraction). Soit $G = (\mathcal{B}, \beta_1, \beta_2)$ une carte combinatoire. Soit \mathcal{K} un sous-ensemble symétrique de \mathcal{B} . \mathcal{K} est un noyau de contraction sur G si et seulement si

- \mathcal{K} forme une forêt de G ;
- \mathcal{K} ne contient pas tous les brins de G .

L'ensemble \mathcal{BS} désigne l'ensemble des brins survivants défini par $\mathcal{BS} = \mathcal{B} - \mathcal{K}$. Par définition, $\mathcal{BS} \neq \emptyset$.

9. Une forêt d'arbres est dite *recouvrante* d'un ensemble \mathcal{S} de sommets si et seulement si tous les éléments de \mathcal{S} sont représentés par un nœud d'un arbre de la forêt.

Définition 37 (Noyau de suppression). *Soit $G = (\mathcal{B}, \beta_1, \beta_2)$ une carte combinatoire. Soit \mathcal{K} un sous-ensemble symétrique de \mathcal{B} . \mathcal{K} est un noyau de suppression sur G si et seulement si \mathcal{K} est un noyau de contraction sur \overline{G} .*

Nous pouvons donc considérer un noyau de suppression comme un noyau de contraction appliqué à la carte duale. La définition des noyaux de contraction repose sur des critères d'homogénéité spécifiques aux applications visées. Inversement, les noyaux de suppression sont utilisés pour « nettoyer » les cartes contractées par suppression des arêtes redondantes. La hiérarchie d'une pyramide combinatoire s'établit à travers la notion de *chemin de connexion* (*connecting walk* en anglais) qui s'apparente à la notion de fenêtre de réduction en termes de brins (Définition 38). L'idée consiste à parcourir un ensemble de brins supprimés depuis un brin donné pour obtenir son image dans le niveau suivant.

Définition 38 (Chemin de connexion). *Soit $G = (\mathcal{B}, \beta_1, \beta_2)$ une carte combinatoire. Soit \mathcal{K} un noyau de G et b un brin survivant ($b \in \mathcal{BS}$). Le chemin de connexion associé à b est défini suivant les deux conditions suivantes.*

1. Si \mathcal{K} est un noyau de contraction alors :

$$CW(b) = b\beta_1(b) \dots \beta_1^{n-1}(b) \text{ avec } n = \min\{k \in \mathbb{N}^* \mid \beta_1^k(b) \in \mathcal{BS}\}.$$

2. Si \mathcal{K} est un noyau de suppression alors :

$$CW(b) = b\beta_{12}(b) \dots \beta_{12}^{n-1}(b) \text{ avec } n = \min\{k \in \mathbb{N}^* \mid \beta_{12}^k(b) \in \mathcal{BS}\}.$$

Il est également intéressant de noter que l'on peut directement construire un niveau l depuis un niveau k ($l > k$) par composition des noyaux de contraction/suppression. Ce résultat a permis de généraliser les chemins de connexion aux *séquences de connexion* (Définition 39).

Définition 39 (Séquence de connexion). Soit $G = (\mathcal{B}, \beta_1, \beta_2)$ une carte combinatoire. Soit $\mathcal{K}_0, \dots, \mathcal{K}_n$ une séquence de noyaux de contraction ou de suppression. Une séquence de connexion est définie par une construction récursive :

- $\forall b \in \mathcal{B}, \mathcal{SC}_0(b) = b$;
- Pour chaque niveau $i \in \{1, \dots, n\}$ et pour chaque $b \in \mathcal{BS}^i$:
 - si \mathcal{K}_{i-1} et \mathcal{K}_i sont de même type alors

$$\mathcal{SC}_i(b) = \mathcal{SC}_{i-1}(b_1) \dots \mathcal{SC}_{i-1}(b_p);$$
 - si \mathcal{K}_{i-1} et \mathcal{K}_i sont de type différent alors

$$\mathcal{SC}_i(b) = b_1.\mathcal{SC}_{i-1}^*(\beta_2(b_1)) \dots b_p.\mathcal{SC}_{i-1}^*(\beta_2(b_p));$$

où :

- (b_1, \dots, b_p) correspond au chemin de connexion $\mathcal{CW}^i(b)$;
- $\mathcal{SC}_i^*(b)$ désigne la séquence de connexion $\mathcal{SC}_i(b)$ privée de son premier brin;
- $\mathcal{K}_0 = \emptyset$ et \mathcal{K}_1 ont le même type par convention.

La Définition 38 ainsi que les propriétés des séquences de connexion [Tut63] permettent de proposer un codage dit *implicite* d'une pyramide de cartes combinatoires [BK03b]. L'aspect implicite vient du fait que seule la base de la pyramide est représentée explicitement, les autres niveaux se déduisent d'après les opérations définies par une séquence de noyaux de contraction et de suppression (Définition 40).

Définition 40 (Codage implicite d'une pyramide). Soit P une pyramide de $n + 1$ niveaux définie par une carte initiale $G^0 = (\mathcal{B}, \beta_1, \beta_2)$ et une séquence de n noyaux $\{\mathcal{K}^1, \dots, \mathcal{K}^n\}$. Le codage implicite de P est un triplet $(G^0, \text{état}, \text{niveau})$ où *état* et *niveau* sont deux fonctions vérifiant :

- *état* est une fonction définie de $\{1, \dots, n\}$ à valeur dans les deux états $\{\text{Contracté}, \text{Supprimé}\}$ telle que

$$\forall i, 1 \leq i \leq n, \text{état}(i) = \begin{cases} \text{Contracté} & \text{si } \mathcal{K}^i \text{ est un noyau de contraction} \\ \text{Supprimé} & \text{sinon} \end{cases}$$
- *niveau* est une fonction de \mathcal{B} à valeur dans $\{1, \dots, n + 1\}$ associant à chaque brin de la carte initiale $G^0 = (\mathcal{B}, \beta_1, \beta_2)$, le niveau maximal où il survit. Autrement dit,

$$\forall b \in \mathcal{B}, \text{niveau}(b) = \max\{i \in \{1, \dots, n + 1\} \mid b \in \mathcal{BS}^{i-1}\}.$$

L'opération pour reconstruire un niveau donné a un coût proportionnel au nombre de brins du niveau initial [BK00]. En comparaison des pyramides de graphes duaux, les pyramides combinatoires ont l'avantage de n'avoir qu'une seule structure à maintenir à jour car elles présentent un codage implicite du dual. Ce modèle hiérarchique hérite également des avantages liés aux

cartes combinatoires avec notamment la caractérisation correcte des relations d'imbrication en utilisant l'orientation des frontières.

1.4.6 Pyramides généralisées

Les pyramides de cartes généralisées, appelées *pyramides généralisées* sont définies comme une pile de n -cartes généralisées (Section 1.3.3) successivement réduites. Ce modèle a été introduit par Simon *et al.* [GDL05a] comme une généralisation en dimension n des pyramides combinatoires définies par Brun et Kropatsch [BK03a].

Le modèle formalise les précédentes notions de chemin de connexion et de champ récepteur et apporte une définition étendue de la notion d'orbite avec les *orbites généralisées* [GDL05b]. La réduction d'un niveau au niveau suivant s'effectue par application d'opérations de suppression et de contraction, et conduit à la définition suivante :

Définition 41 (Pyramide de nG -cartes). *Soit $n, m \geq 0$. Une pyramide P de nG -cartes comportant $m + 1$ niveaux est l'ensemble $P = \{G^k\}_{0 \leq k \leq m}$ où :*

- $\forall k, 0 \leq k \leq m, G^k = (\mathcal{B}^k, \alpha_0, \dots, \alpha_n)$ est une nG -carte ;
- $\forall l, 0 \leq k \leq m, G^k$ est obtenue à partir de G^{k-1} par application de l'opération de suppression et contraction simultanées.

Nous pouvons notamment distinguer trois types de représentations des pyramides généralisées [GDL06] : les représentations *explicite*, *hiérarchique* et *implicite*, détaillées ci-dessous et illustrées dans la Figure 1.27.

La représentation explicite

Tous les brins et toutes les involutions de chaque niveau sont représentés ainsi que les relations parent-enfant entre les niveaux. Ce mode de représentation correspond exactement à la définition même du modèle (Définition 41). La représentation explicite présente une redondance de l'information qui se traduit par un surcoût en mémoire. Cependant, elle reste intéressante dans certains cas car elle permet de charger en mémoire uniquement les éléments relatifs à un niveau donné. Un exemple de représentation explicite est proposé dans la Figure 1.27(a).

La représentation hiérarchique

Les différents niveaux de la pyramide sont représentés par le même ensemble de brins. La représentation hiérarchique améliore ainsi la compacité du modèle

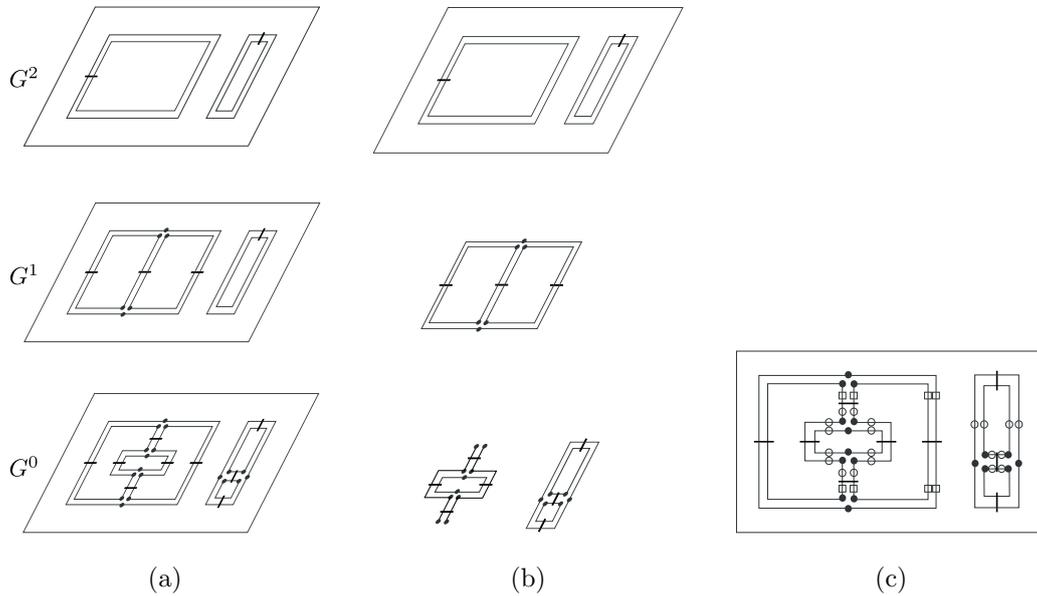


FIGURE 1.27 – Pyramides généralisées. (a) Représentation explicite : chaque niveau représente la $2G$ -carte complète du niveau associé ; (b) Représentation hiérarchique : la duplication des brins est évitée. Les brins ne sont représentés que dans le dernier niveau auquel ils apparaissent ; (c) Représentation implicite : seul le niveau de base est représenté mais un symbole indique le niveau auquel un brin est supprimé. Les symboles \circ et \square indiquent que les brins sont respectivement supprimés dans G^1 et G^2 .

en évitant la duplication inutile des brins. En revanche, les involutions sont représentées de façon explicite. Un exemple de représentation hiérarchique est proposé dans la Figure 1.27(b).

La représentation implicite

De manière similaire au codage implicite dans les pyramides combinatoires, seule la base de la pyramide est représentée ainsi que l'ensemble des opérations menant à la construction des niveaux suivants. Les autres niveaux se déduisent ainsi en simulant l'application de ces opérations. Ce mode de représentation possède la meilleure compacité. Néanmoins, les opérations de manipulation sont rendues plus coûteuses (dépendant essentiellement de la longueur des séquences de connexion). Un exemple de représentation implicite est proposé dans la Figure 1.27(c).

1.5 Conclusion

Au cours de ce chapitre, nous avons présenté différents modèles relationnels pour la représentation d'images 2D. Deux catégories ont été différenciées : les structures hiérarchiques et non hiérarchiques. Ces modèles définissent une partition de l'image (ou une pile de partitions pour les structures hiérarchiques) fondée sur la notion de régions. En Tableau 1.3, nous synthétisons les différentes possibilités offertes par ces modèles en ce qui concerne les relations entre les régions. Ainsi, les colonnes 2, 3 et 4 indiquent si un modèle donné est capable (symbole +) ou non (symbole -) de différencier les différents cas d'adjacence. Les colonnes 5 et 6 renseignent sur la capacité du modèle à appliquer certaines opérations courantes sur les régions comme la fusion et la division de régions. Enfin, la colonne 7 rappelle le type du processus de construction des modèles hiérarchiques en indiquant les constructions ascendantes (par \Uparrow) et descendantes (par \Downarrow).

Chacun des modèles présentés dans ce comparatif possède ses propres avantages et inconvénients. Avec ce tableau, il s'agit surtout de présenter l'ensemble des opérations et des informations relationnelles disponibles pour les régions d'une partition dans une application en traitement d'image. Aussi, une application précise n'aura pas forcément l'utilité d'avoir une représentation exhaustive des relations d'adjacence et des modèles plus simples pourront être privilégiés par rapport à des modèles plus complets. Enfin, le symbole $+/-$ fait référence au cas limite de la représentation des relations d'imbrication présenté pour les modèles de graphes duaux dans la Section 1.3.2. Par extension, nous retrouvons également cet inconvénient pour les pyramides de graphes duaux.

Concernant les structures hiérarchiques, nous remarquons que les structures régulières ont été particulièrement étudiées et que des modèles de construction ascendants et descendants ont été proposés, ces différentes approches provenant de justifications psycho-visuelles. À l'inverse, les méthodes de construction descendantes pour les pyramides irrégulières ont été délaissées certainement en raison des insuffisances des graphes.

Dans le cadre d'une application en histologie, nous avons identifié trois composantes clés que doit proposer un modèle adapté à l'utilisation des pathologistes. Nous avons ainsi besoin d'un modèle capable de représenter une hiérarchie de grandes images ainsi que les relations de multi-adjacence et d'imbrication. De plus, nous souhaitons un modèle dont la construction est descendante pour reproduire l'analyse par focus d'attention des pathologistes. De ce fait, nous proposons de définir un nouveau modèle relationnel hiérarchique à construction descendante utilisant les pyramides combinatoires.

TABLE 1.3 – Tableau comparatif des différents modèles relationnels pour la représentation d'images 2D. Les symboles + et – indiquent la capacité ou l'incapacité du modèle à prendre en compte une propriété des régions. Le symbole +/- indique un support partiel de la propriété. Les symboles ↑ et ↓ différencient les constructions ascendantes et descendantes pour les modèles hiérarchiques.

Modèle	Adjacence	Adjacences multiples	Imbrication	Fusion	Découpe	Construction hiérarchique
RAG	+	–	–	+	–	
Graphes duaux	+	+	+/-	+	–	
Cartes combinatoires	+	+	–	+	+	
Cartes topologiques	+	+	+	+	+	
Cartes généralisées	+	+	+	+	+	
Quadtrees	–	–	–	–	+	↓
Pyramides régulières	–	–	–	+	–	↑
Pyramides de graphes simples	+	–	–	+	–	↑
Pyramides de graphes duaux	+	+	+/-	+	–	↑
Pyramides combinatoires	+	+	+	+	–	↑
Pyramides généralisées	+	+	+	+	–	↑

Dans la mesure où un unique niveau de la pyramide peut être associé à une image de taille importante, nous introduisons dans le chapitre suivant le modèle de carte topologique tuilée, fondé sur le modèle de carte topologique et permettant la représentation de grandes images.

Chapitre 2

Cartes topologiques tuilées

Sommaire

2.1	Introduction	66
2.2	Tuiles topologiques	67
2.2.1	Définition du modèle de tuile topologique	68
2.2.2	Équivalences au bord des tuiles	69
2.2.3	Relations d'adjacence entre tuiles	71
2.3	Cartes tuilées	75
2.3.1	Abstraction des bords fictifs	77
2.3.2	Définition d'opérateurs	79
2.3.3	Définition du modèle de cartes tuilées	82
2.4	Construction d'une première partition	83
2.4.1	Extraction incrémentale de la carte topologique d'une tuile	84
2.4.2	Connexion de tuiles adjacentes	86
2.4.3	Itération sur toutes les tuiles	89
2.5	Raffinement d'une région par un processus de décomposition/fusion	90
2.5.1	Première étape : décomposition	91
2.5.2	Deuxième étape : fusion	94
2.5.3	Extension du processus de décomposition/fusion aux cartes tuilées	96
2.6	Structures de données et implantation	100
2.6.1	Structures de données	100
2.6.2	Déchargement d'une tuile topologique	102
2.6.3	Chargement d'une tuile topologique	104
2.7	Conclusion	104

2.1 Introduction

Dans le domaine du traitement d'images, l'objectif d'une opération de segmentation consiste à proposer un découpage de l'image en régions homogènes permettant de délimiter des formes ou des zones d'intérêt ce qui en fait une étape critique dans de nombreuses applications comme la reconnaissance d'images, le suivi de mouvement, l'indexation, etc.

Par ailleurs, la segmentation manuelle d'une image requiert beaucoup de temps et conduit à la réalisation d'une partition qui sera différente suivant la personne qui l'aura réalisée. Cela justifie l'intérêt de développer des méthodes de segmentation automatisées ou semi-automatisées, rapides et dont le résultat déterministe peut être utilisé comme élément de comparaison.

Le traitement d'images histologiques, comme d'autres domaines, demande une analyse de l'image tenant compte de critères topologiques : une lésion est par exemple une zone accolée à du tissu conjonctif ou graisseux (relation d'adjacence), un noyau est une zone imbriquée dans le cytoplasme d'une cellule (relation d'imbrication).

En outre, certaines applications en segmentation demandent à traiter des images de taille très importante. Cette contrainte soulève de nombreux défis techniques notamment liés à la quantité restreinte de mémoire disponible. Toujours dans le cas du traitement d'images histologiques, certaines images peuvent atteindre des dimensions de $32\,000 \times 32\,000$ pixels. Ainsi, proposer une segmentation automatisée ou semi-automatisée de ce type d'images requiert une structure de données :

- capable de représenter les adjacences et imbrications des objets de la partition ;
- adaptée au traitement d'images de grandes tailles.

À travers ce chapitre, nous présentons les étapes principales de la construction d'un nouveau modèle de représentation d'image 2D répondant à ces contraintes. Dans la Section 2.2, nous étudions les possibilités d'une décomposition des cartes topologiques en tuiles. Cette décomposition entraîne des problèmes de bords importants. C'est pourquoi nous approfondissons la représentation des relations d'adjacence entre deux tuiles. Nous définissons ensuite un nouveau modèle de cartes tuilées dans la Section 2.3 qui supprime les effets de bords introduits par la décomposition en tuiles. Dans la Section 2.4, nous proposons un algorithme pour la construction de notre modèle de carte tuilée. La méthode que nous proposons est incrémentale et ne demande qu'un nombre limité de tuiles en mémoire ce qui garantit la faisabilité de l'opération, même dans le cadre des grandes images. Dans la Section 2.5, nous étudions les opérations de raffinement et de fusion de régions dans le cas de notre modèle.

Enfin, nous détaillons dans la Section 2.6 les structures de données utilisées pour les opérations de chargement et de déchargement sur disque d'une tuile topologique.

2.2 Tuiles topologiques

Une carte topologique est un modèle de représentation d'image codant la géométrie et la topologie d'une partition (Section 1.3.4). Néanmoins, si nous considérons le cas d'une image de grande taille, l'utilisation de ce modèle n'est pas envisageable d'un point de vue occupation mémoire pour représenter toute l'image. Notre objectif consiste ainsi à proposer un nouveau modèle fondé sur les cartes topologiques qui soit adapté à la représentation d'images de grandes tailles.

Lorsqu'il n'est pas possible ou trop coûteux de représenter un élément dans sa globalité, une solution consiste à le décomposer en sous-ensembles de plus petite taille appelés *tuiles*. Cette décomposition dispose généralement de propriétés permettant de reconstruire facilement l'ensemble initial à partir de ces sous-ensembles. L'intérêt de proposer une division d'une carte en tuiles tient au fait qu'une majorité des traitements à appliquer sur une image peuvent être locaux et nécessitent donc un nombre réduit de tuiles en mémoire. Ainsi, les tuiles inutilisées peuvent être stockées sur disque dur pour libérer l'espace mémoire nécessaire.

La problématique d'une décomposition d'une carte topologique en tuiles peut être abordée suivant différentes approches. Une première approche consiste à considérer une tuile comme un ensemble connexe de régions. Cette solution entraîne néanmoins plusieurs inconvénients. Par exemple, il n'est pas possible de borner l'occupation mémoire d'une tuile car une région peut contenir toute l'image. De plus, le positionnement des tuiles dans le référentiel de l'image est difficile. Une seconde approche considère une décomposition géométrique suivant un pavage du plan. Dans le cas d'une image, un pavage régulier fournit une décomposition adéquate dans la mesure où il construit une partition simple en cellules de taille fixe. En considérant les trois différents pavages définis en dimension deux (Section 1.1.1) une tuile est donc une forme géométrique carrée, triangulaire ou hexagonale. Comme les images que nous traitons utilisent un pavage carré, il nous paraît naturel de privilégier ce type de pavage.

Par conséquent, nous définissons dans la Section 2.2.1 le modèle de tuile topologique pour la représentation d'une subdivision rectangulaire d'une image. Dans la Section 2.2.2, nous définissons une relation d'équivalence entre les

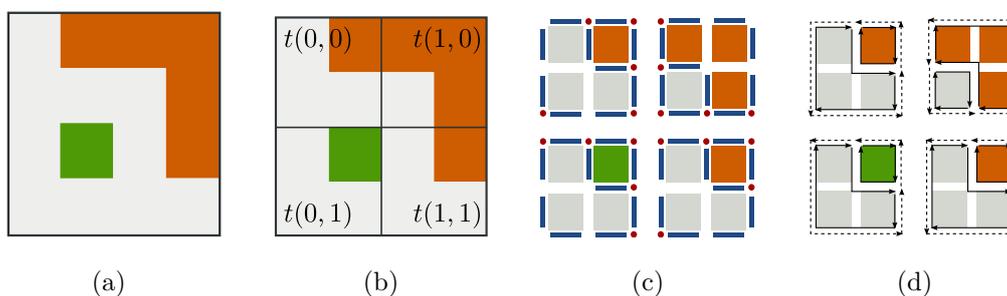


FIGURE 2.1 – Décomposition d’une image en tuiles. (a) Image originale; (b) Décomposition en quatre tuiles suivant un pavage régulier carré; (c) Représentation des frontières interpixel des quatre subdivisions. Les disques et les segments désignent les pointels et les lignels frontières; (d) Cartes combinatoires associées aux quatre subdivisions. Les brins en pointillés appartiennent à la région infinie.

cellules interpixel appartenant aux bords des tuiles. Dans la Section 2.2.3, en imposant l’existence d’une bijection entre les brins du bord de deux tuiles adjacentes, nous construisons un nouvel opérateur β'_2 permettant d’obtenir un ensemble de tuiles topologiques dont les bords sont correctement mis en relation.

2.2.1 Définition du modèle de tuile topologique

Intuitivement, une tuile topologique est à une carte topologique ce qu’une tuile est à une image : une subdivision rectangulaire d’un ensemble telle que la juxtaposition de toutes les tuiles conduit à la reconstruction de l’ensemble initial. Par conséquent, nous définissons une tuile topologique (Définition 42) comme une carte topologique locale composée d’une carte combinatoire, d’une matrice d’éléments interpixel associée à la subdivision rectangulaire, ainsi qu’un arbre des régions comme illustré par la Figure 2.1.

Définition 42 (Tuile topologique). *Soit I une image de largeur L et de hauteur H . Soit T un pavage rectangulaire de I composé de n lignes et de m colonnes. Une tuile topologique, notée $t(i, j)$ est la carte topologique associée à la subdivision (i, j) de T où $\forall i, 0 \leq i < m, \forall j, 0 \leq j < n$:*

- $(L/m, H/n)$ sont les dimensions de la tuile en largeur et en hauteur ;
- $(i \times L/m, j \times H/n)$ sont les coordonnées de l’extrémité haut/gauche de la tuile dans le référentiel de l’image ;

Deux tuiles topologiques t et t' sont dites adjacentes si elles ont un bord en commun dans le référentiel de l’image. Cette relation, notée $t \text{ adj } t'$,

est trivialement symétrique. Puisque notre définition d'une tuile topologique considère chaque subdivision comme une carte topologique indépendante, il est nécessaire de reporter les relations d'adjacence entre les tuiles sur les cartes pour retrouver un modèle global de représentation de l'image.

2.2.2 Équivalences au bord des tuiles

D'un point de vue géométrique, deux tuiles adjacentes ont un bord commun correspondant à une même séquence d'éléments interpixel. En pratique, ce bord commun est dupliqué dans chacune des deux tuiles adjacentes. Nous définissons ainsi la notion de cellules interpixel équivalentes (Définition 43 et Définition 44) pour distinguer ces cellules appartenant au bord des tuiles.

Notons que de façon générale, nous exprimons les coordonnées d'une tuile dans le référentiel de l'image. Cependant, nous nous plaçons dans un repère local à chaque tuile lorsque nous parlons des coordonnées de ses éléments interpixel. Ainsi, les coordonnées (i, j) d'un pointel désignent ses coordonnées par rapport à la tuile et non par rapport à l'image.

Dans la mesure où les frontières des régions sont représentées par des $(0, 1)$ -chemins, nous nous intéressons particulièrement à l'équivalence entre les pointels (0-cellules) et les lignels (1-cellules).

Définition 43 (0-cellules équivalentes). *Soient c et c' deux 0-cellules de coordonnées respectives (i, j) et (i', j') . Les deux 0-cellules sont équivalentes si et seulement si elles ont les mêmes coordonnées dans le repère de l'image.*

Autrement dit, dans le cas d'un pavage carré composé de tuiles de largeur l et de hauteur h , deux pointels p et p' de coordonnées respectives (i, j) et (i', j') appartenant à deux tuiles adjacentes distinctes t et t' sont équivalents si et seulement si l'une des conditions suivantes est satisfaite :

1. $i' = i + l$ et $j' = j$;
2. $i' = i - l$ et $j' = j$;
3. $j' = j + h$ et $i' = i$;
4. $j' = j - h$ et $i' = i$.

Les conditions (1), (2), (3) et (4) correspondent respectivement à une adjacence à gauche, à droite, en haut et en bas de la tuile t' par rapport à t . Par exemple, dans la Figure 2.2(a), les pointels p_1 et p'_1 sont équivalents.

Définition 44 (1-cellules équivalente). *Deux 1-cellules sont équivalentes si et seulement si leurs 0-cellules incidentes sont équivalentes.*

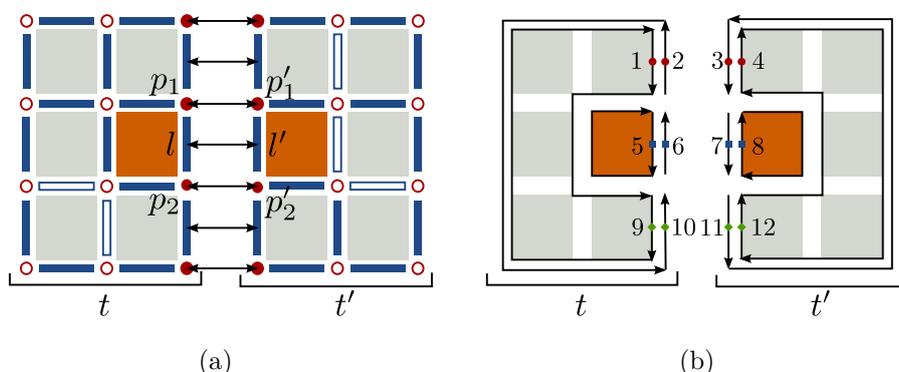


FIGURE 2.2 – Entités équivalentes au niveau des bords entre deux tuiles adjacentes t et t' . (a) Entre les k -cellules : les doubles flèches désignent les pointels et lignels équivalents ; (b) Entre les arêtes : les arêtes du bord des tuiles marquées d'un même symbole sont équivalentes.

À partir des définitions de 0-cellules et 1-cellules équivalentes, nous pouvons étendre cette notion aux $(0, 1)$ -chemins (Définition 45).

Définition 45 ($(0, 1)$ -chemins équivalents). *Deux $(0, 1)$ -chemins (c_0, c_1, \dots, c_n) et $(c'_0, c'_1, \dots, c'_n)$ sont équivalents si et seulement si $\forall i, 0 \leq i \leq n$, les cellules c_i et c'_i sont équivalentes.*

Nous notons $e \sim e'$ deux entités (pointels, lignels ou $(0, 1)$ -chemins) équivalentes. Par exemple, dans la Figure 2.2(a), le lignel l (resp. l') est incident aux deux pointels p_1 et p_2 (resp. p'_1 et p'_2). Comme $p_1 \sim p'_1$ et $p_2 \sim p'_2$, les lignels l et l' sont équivalents. La relation \sim est une relation d'équivalence (Proposition 1).

Proposition 1 (Relation d'équivalence). *La relation binaire \sim , définie sur l'ensemble des 0-cellules (resp. 1-cellules, $(0, 1)$ -chemins) équivalentes de T , est une relation d'équivalence.*

Démonstration. La relation \sim entre deux pointels équivalents est réflexive, symétrique et transitive puisque les coordonnées des pointels dans le référentiel de l'image sont identiques. La relation \sim est une relation d'équivalence entre les lignels (resp. $(0, 1)$ -chemins) par équivalence des pointels incidents (resp. des 0-cellules et des 1-cellules composant le chemin). \square

Nous pouvons remarquer que deux entités équivalentes et appartenant à une même tuile sont égales.

2.2.3 Relations d'adjacence entre tuiles

Nous représentons chaque partition d'une tuile du pavage de l'image par une carte topologique. De façon à garantir une cohérence des relations topologiques entre les tuiles, nous cherchons à obtenir un modèle topologique global à l'image.

Étant donné qu'une arête définit deux $(0, 1)$ -chemins maximaux, l'arête a est dite équivalente à l'arête a' si et seulement si un des deux $(0, 1)$ -chemins maximaux de a est équivalent à un des deux $(0, 1)$ -chemins maximaux de a' . Dans l'exemple de la Figure 2.2(b), nous obtenons $\{1, 2\} \sim \{3, 4\}$, $\{5, 6\} \sim \{7, 8\}$ et $\{9, 10\} \sim \{11, 12\}$.

Puisque chaque tuile est une subdivision indépendante, il s'agit de définir les relations d'adjacence au niveau des brins appartenant au bord des tuiles. Pour cela, nous commençons par introduire plusieurs notations permettant de caractériser différents sous-ensembles des brins d'une tuile topologique t :

- r_∞^t désigne la région infinie de t ;
- \mathcal{B}^t désigne l'ensemble des brins de t ;
- $\mathcal{B}_\infty^t = \{b \in \mathcal{B}^t \mid \text{région}(b) = r_\infty^t\}$ désigne l'ensemble des brins appartenant à la région infinie de t ;
- $\mathcal{B}_\partial^t = \{b \in \mathcal{B}^t \mid \text{région}(\beta_2(b)) = r_\infty^t\}$ désigne l'ensemble des brins du bord de t .

Par construction, excepté dans le cas des quatre coins de l'image, nous imposons qu'aucun brin n'appartienne à deux côtés distincts d'une tuile en autorisant éventuellement la présence d'un sommet de degré 2 aux extrémités des tuiles. Notons que l'introduction des sommets de degré 2 aux extrémités des tuiles entraîne que la carte combinatoire d'une tuile topologique n'est plus forcément minimale au niveau des bords. Néanmoins, cette contrainte est nécessaire à la définition d'une relation d'adjacence entre tuiles topologiques.

En effet, nous pouvons ainsi décomposer chacun des deux ensembles \mathcal{B}_∂^t et \mathcal{B}_∞^t en quatre sous-ensembles $\mathcal{B}_\partial^t = \bigcup_{i=0}^3 \mathcal{B}_{\partial,i}^t$, $\mathcal{B}_\infty^t = \bigcup_{i=0}^3 \mathcal{B}_{\infty,i}^t$, correspondant respectivement aux quatre côtés (gauche, haut, droite, bas) de la tuile. Par exemple, dans la Figure 2.3(a), nous avons :

- $\mathcal{B}_{\partial,0}^t = \{7\}$ et $\mathcal{B}_{\infty,0}^t = \{1\}$;
- $\mathcal{B}_{\partial,1}^t = \{8\}$ et $\mathcal{B}_{\infty,1}^t = \{6\}$;
- $\mathcal{B}_{\partial,2}^t = \{9, 13\}$ et $\mathcal{B}_{\infty,2}^t = \{4, 5\}$;
- $\mathcal{B}_{\partial,3}^t = \{11, 14\}$ et $\mathcal{B}_{\infty,3}^t = \{2, 3\}$.

Nous introduisons ensuite une fonction $\text{bord} : T \times T \rightarrow \{\emptyset, 0, 1, 2, 3\}$ telle que $\text{bord}(t, t')$ soit égal à \emptyset si les deux tuiles t et t' ne sont pas adjacentes et soit égal à la relation d'adjacence entre t et t' dans le cas

contraire. Ainsi, pour tout couple de tuiles adjacentes t et t' , nous avons $\text{bord}(t, t') = \text{bord}(t', t) + 2 \pmod{4}$.

Par conséquent, l'ensemble $\mathcal{B}_{\partial, \text{bord}(t, t')}^t$ désigne l'ensemble des brins du bord d'une tuile t par rapport à une tuile adjacente t' . Nous imposons alors l'existence d'une bijection $\varphi^{t, t'}$ (Définition 46) entre les deux ensembles $\mathcal{B}_{\partial, \text{bord}(t, t')}^t$ et $\mathcal{B}_{\partial, \text{bord}(t', t)}^{t'}$ et telle que les deux arêtes incidentes aux brins b et $\varphi^{t, t'}(b)$ soient deux arêtes équivalentes. Notons qu'il est toujours possible de trouver au moins une bijection $\varphi^{t, t'}$ en décomposant les brins de $\mathcal{B}_{\partial, \text{bord}(t, t')}^t$ et $\mathcal{B}_{\partial, \text{bord}(t', t)}^{t'}$ en un ensemble de brins élémentaires dont le plongement ne comprend qu'un seul lignel (Section 1.3.4).

Définition 46 (Bijection φ). *Soient t et t' deux tuiles topologiques adjacentes (t adj t'). Il existe une bijection $\varphi^{t, t'} : \mathcal{B}_{\partial, \text{bord}(t, t')}^t \rightarrow \mathcal{B}_{\partial, \text{bord}(t', t)}^{t'}$ telle que $\forall b \in \mathcal{B}_{\partial, \text{bord}(t, t')}^t$, $\langle \beta_2 \rangle(b)$ et $\langle \beta_2 \rangle(\varphi^{t, t'}(b))$ sont deux arêtes équivalentes.*

Dans la suite de ce manuscrit, nous simplifions la notation des ensembles de brins d'une tuile par rapport à une tuile adjacente en notant :

- $\mathcal{B}_{\partial}^{t, t'} = \mathcal{B}_{\partial, \text{bord}(t, t')}^t$;
- $\mathcal{B}_{\infty}^{t, t'} = \mathcal{B}_{\infty, \text{bord}(t, t')}^t$.

Notons que par symétrie, l'adjacence entre t et t' s'exprime par la bijection inverse de l'adjacence entre t' et t (Proposition 2).

Proposition 2. *Soient t et t' deux tuiles adjacentes. La bijection $\varphi^{t, t'}$ entre t et t' est égale à l'inverse de la bijection $\varphi^{t', t}$ entre t' et t ($\varphi^{t, t'} = (\varphi^{t', t})^{-1}$).*

Démonstration. Soit $b \in \mathcal{B}_{\partial}^{t, t'}$.

Le brin $\varphi^{t', t} \circ \varphi^{t, t'}(b) \in \mathcal{B}_{\partial}^{t', t}$ est l'unique brin de $\mathcal{B}_{\partial}^{t', t}$ tel que :

$$\langle \beta_2 \rangle(\varphi^{t', t} \circ \varphi^{t, t'}(b)) \sim \langle \beta_2 \rangle(\varphi^{t, t'}(b)).$$

De même, $\varphi^{t, t'}(b)$ est l'unique brin de $\mathcal{B}_{\partial}^{t, t'}$ tel que :

$$\langle \beta_2 \rangle(\varphi^{t, t'}(b)) \sim \langle \beta_2 \rangle(b).$$

Par transitivité de la relation d'équivalence :

$$\langle \beta_2 \rangle(b) \sim \langle \beta_2 \rangle(\varphi^{t', t} \circ \varphi^{t, t'}(b)) \in \mathcal{B}^t.$$

2.2. Tuiles topologiques

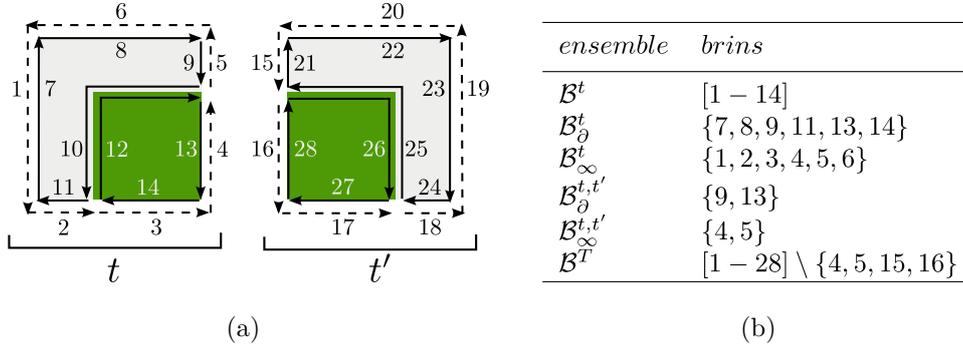


FIGURE 2.3 – Différents ensembles caractéristiques de brins entre deux tuiles adjacentes. Les brins en pointillés appartiennent à la région infinie. (a) Deux tuiles adjacentes t et t' ; (b) Ensembles caractéristiques de la tuile t . La notation $[i - j]$ désigne l'ensemble des brins i à j .

Puisque deux arêtes possédant le même plongement et appartenant à la même tuile sont égales, nous obtenons :

$$\begin{aligned}
 \langle \beta_2 \rangle (b) &= \{b, \beta_2(b)\} \\
 &= \langle \beta_2 \rangle (\varphi^{t',t} \circ \varphi^{t,t'}(b)) \\
 &= \{\varphi^{t',t} \circ \varphi^{t,t'}(b), \beta_2(\varphi^{t',t} \circ \varphi^{t,t'}(b))\}
 \end{aligned}$$

Or, $\{b, \varphi^{t',t} \circ \varphi^{t,t'}(b)\} \subset \mathcal{B}_\partial^{t,t'}$ et $\{\beta_2(b), \beta_2(\varphi^{t',t} \circ \varphi^{t,t'}(b))\} \subset \mathcal{B}_\infty^{t,t'}$, avec

$$\mathcal{B}_\partial^{t,t'} \cap \mathcal{B}_\infty^{t,t'} = \emptyset.$$

Nous avons donc $b = \varphi^{t',t} \circ \varphi^{t,t'}(b)$ et $\beta_2(b) = \beta_2(\varphi^{t',t} \circ \varphi^{t,t'}(b))$.

Par conséquent, $\varphi^{t',t} \circ \varphi^{t,t'} = Id. \Rightarrow \varphi^{t',t} = (\varphi^{t,t'})^{-1}$. □

Lors de la construction d'une tuile topologique t , nous nous assurons d'établir la bijection $\varphi^{t,t'}$ en tenant compte de la composition du bord d'une tuile t' adjacente. La décomposition illustrée dans la Figure 2.4(a) présente ainsi une bijection entre les ensembles $\mathcal{B}_\partial^{t,t'}$ et $\mathcal{B}_\infty^{t',t}$, pour tous les couples (t, t') de tuiles adjacentes.

Soit \mathcal{B}^T l'ensemble des brins d'un ensemble T de tuiles, privé des brins appartenant à la région infinie entre deux tuiles adjacentes :

$$\mathcal{B}^T = \bigcup_{t \in T} \mathcal{B}^t \setminus \bigcup_{t \text{ adj } t'} \mathcal{B}_\infty^{t,t'}.$$

De façon à abstraire la décomposition en tuiles, nous introduisons un nouvel opérateur β'_2 (Définition 47) défini sur \mathcal{B}^T . Le comportement de cet opérateur est identique à celui de β_2 , excepté pour les brins du bord des tuiles qui sont reliés aux brins du bord de la tuile adjacente. Notons que les brins de la région infinie d'une tuile marquant le bord de l'image sont bien inclus dans \mathcal{B}^T . Par exemple, dans la Figure 2.4(b), les brins 5 et 6 appartiennent à \mathcal{B}^T contrairement aux brins 2 et 3. Afin de simplifier les figures, si le coin d'une tuile correspond au coin de l'image, nous ne le représentons pas par un sommet de degré 2. Par exemple, les brins 5 et 6 désignent à la fois le bord droit et le bord haut de la tuile $t(1, 0)$.

Définition 47 (Opérateur β'_2). *Soit T un ensemble de tuiles topologiques. L'opérateur β'_2 , défini sur \mathcal{B}^T , vérifie :*

$$\beta'_2 : \mathcal{B}^T \rightarrow \mathcal{B}^T$$

$$b \mapsto \begin{cases} \varphi^{t,t'}(b) & \text{si } \exists (t, t') \in T \times T \text{ tel que } t \text{ adj } t' \text{ et } b \in \mathcal{B}_\partial^{t,t'} \\ \beta_2(b) & \text{sinon.} \end{cases}$$

D'un point de vue topologique, les brins de \mathcal{B}_∂^t sont reliés à la fois à un brin de \mathcal{B}_∞ par l'involution β_2 et au brin correspondant dans la tuile adjacente par l'involution β'_2 (appartenant à $\mathcal{B}_\partial^{t'}$). Deux cas de figure se présentent :

- soit il existe une tuile t' adjacente le long du bord d'une tuile t contenant b et il existe un unique brin $\varphi^{t,t'}(b)$ incident à une arête équivalente ;
- soit il n'existe pas de tuile adjacente (cas des tuiles appartenant au bord de l'image) et dans ce cas, β'_2 est égal à β_2 .

Dans la Figure 2.4(b), les arêtes $\{1, 2\}$ et $\{3, 4\}$ sont équivalentes. Le brin 1 est relié par β_2 au brin 2 appartenant à la région infinie de $t(0, 0)$ (représenté en pointillés) et au brin $\varphi^{t(0,0),t(1,0)}(1) = 4$ de la tuile adjacente $t(1, 0)$ par β'_2 . En revanche, bien que le brin 5 appartienne au bord de la tuile $t(1, 0)$, il n'existe pas de tuile adjacente le long du bord contenant 5 (bord de l'image). Dans ce cas là, nous avons $\beta'_2(5) = \beta_2(5) = 6$.

Proposition 3 (Involution β'_2). *Soit T un ensemble de tuiles topologiques. L'opérateur β'_2 est une involution sur \mathcal{B}^T .*

Démonstration. Soit $b \in \mathcal{B}^T$. Il existe une unique tuile t telle que $b \in \mathcal{B}^t \cap \mathcal{B}^T$.

Si $\exists t' \text{ adj } t \mid b \in \mathcal{B}_\partial^{t,t'}$ alors $\beta'_2(b) = \varphi^{t,t'}(b) \in \mathcal{B}_\partial^{t',t}$ donc $\beta_2^2(b) = \varphi^{t',t} \circ \varphi^{t,t'}(b) = b$ (d'après la Proposition 2).

Sinon, $\beta'_2(b) = \beta_2(b) \in \mathcal{B}^t$.

- si $\beta_2(b) \in \mathcal{B}_\infty^{t,t'}$ alors $b \in \mathcal{B}_\partial^{t,t'}$ ce qui est refusé par hypothèse donc $\beta_2(b) \in \mathcal{B}^T$;

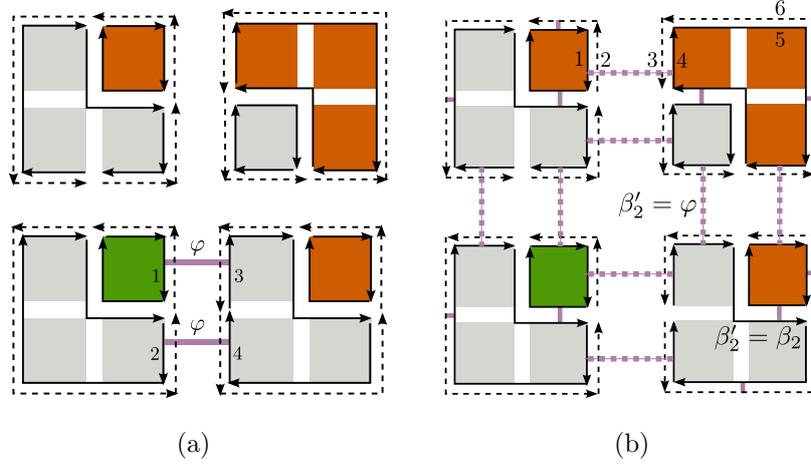


FIGURE 2.4 – Construction de l’opérateur β'_2 entre les quatre tuiles de la Figure 2.1. (a) La décomposition en brins élémentaires du bord des tuiles garantit l’existence d’une bijection φ mettant en relation les brins incidents à des arêtes équivalentes. La bijection $\varphi^{t(0,1),t(1,1)}$ est représentée par des segments gras; (b) Construction de l’involution β'_2 reliant les brins du bord des tuiles aux brins du bord de la tuile adjacente incidents à l’arête équivalente ($\beta'_2 = \varphi$, segments en pointillés), les brins d’une même arête sinon ($\beta'_2 = \beta_2$, segments gras).

- si $\exists t' \text{ adj } t \mid \beta_2(b) \in \mathcal{B}_\partial^{t,t'}$ alors $b \in \mathcal{B}_\infty^{t,t'}$ ce qui est impossible car $b \in \mathcal{B}^T$.
Ainsi, $\beta_2(b) \in \mathcal{B}^T - \mathcal{B}_\partial^{t,t'}$ donc $\beta'_2(\beta_2(b)) = \beta_2^2(b) = \beta_2(b) = b$.

□

En conclusion, notre définition de l’opérateur β'_2 nous permet de définir un ensemble de tuiles dont les bords sont correctement mis en relation, assurant la cohérence du modèle global de représentation de l’image.

2.3 Cartes tuilées

La décomposition en tuiles d’une carte topologique offre l’avantage de maîtriser la quantité de mémoire nécessaire lors d’un traitement, en fixant comme paramètre les dimensions d’une tuile. La définition de l’involution β'_2 (Section 2.2.3) garantit une cohérence globale en représentant les relations d’adjacence entre les différentes tuiles topologiques. Ceci étant, l’approche retenue pour le découpage d’une carte topologique en tuiles entraîne deux conséquences majeures : l’apparition éventuelle de bords fictifs et la répartition

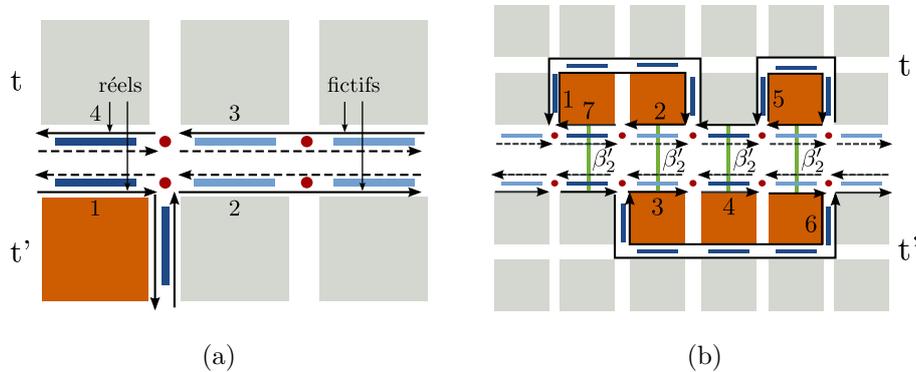


FIGURE 2.5 – Exemples de conséquences liées au découpage d'une carte topologique en tuiles topologiques. (a) Décomposition des bords d'une tuile en bords réels et fictifs; (b) Répartition des régions sur plusieurs tuiles.

des arêtes et régions sur plusieurs tuiles. Ces deux conséquences sont détaillées ci-dessous et illustrées dans la Figure 2.5.

La première conséquence de la subdivision est que le découpage géométrique peut introduire l'apparition de bords qui ne correspondent pas forcément au bord d'une région existante. Ces bords sont alors considérés comme fictifs. Cependant, il existe des cas où ce découpage correspond précisément à la frontière d'une région, d'où la nécessité de différencier les deux cas de figure (Figure 2.5(a)).

La seconde conséquence de la subdivision en tuiles est que des régions peuvent se trouver artificiellement découpées entre plusieurs tuiles (Figure 2.5(b)). Par conséquent, les algorithmes de base, comme par exemple le parcours du contour d'une région, doivent être adaptés pour ne pas tenir compte de ce découpage.

Afin de répondre à ces deux problématiques, nous définissons un modèle global fondé sur le modèle de carte topologique qui est indépendant de la décomposition en tuiles. Dans la Section 2.3.1, nous étudions la notion de bord fictif qui conduit à la définition d'un ensemble de brins désignant uniquement les frontières réelles de l'image tuilée. Dans la Section 2.3.2, nous introduisons deux nouveaux opérateurs définis sur l'ensemble des frontières réelles : une permutation δ_1 et une involution δ_2 permettant de caractériser les sommets, arêtes et faces d'une carte tuilée de manière similaire aux opérateurs β_1 et β_2 dans le cadre des cartes topologiques. Dans la Section 2.3.3, nous présentons notre modèle de carte tuilée qui nous permet d'abstraire le découpage en tuiles.

2.3.1 Abstraction des bords fictifs

Notons \mathcal{R} l'ensemble des régions appartenant à un ensemble de tuiles T . La notion de bord fictif repose sur la définition d'un prédicat symétrique $Oracle : \mathcal{R} \times \mathcal{R} \rightarrow \{\text{vrai}, \text{faux}\}$ indiquant si deux régions sont identiques. Un bord fictif est codé par une arête commune à deux régions r et r' dont l'oracle $Oracle(r, r')$ est *vrai*. Réciproquement, si $Oracle(r, r')$ est *faux* alors le bord est *réel*.

En pratique, nous utilisons une marque sur les brins du bord d'une tuile indiquant si l'arête incidente est un bord fictif ou réel. L'identification des bords fictifs entre deux tuiles adjacentes t et t' peut s'effectuer en parcourant en parallèle les brins reliés par β'_2 . L' $Oracle$ est ensuite évalué entre les régions $r = \text{région}(b)$ et $r' = \text{région}(b')$, pour chaque couple de brins $(b, b' = \beta'_2(b))$ avec $b \in \mathcal{B}_\partial^{t, t'}$ et $b' \in \mathcal{B}_\partial^{t', t}$.

Puisque $Oracle$ est défini sur les régions, quel que soit le couple de brins (b, b') considéré vérifiant $\text{région}(b) = r$ et $\text{région}(b') = r'$, le résultat de la fonction $Oracle$ entre $\text{région}(b)$ et $\text{région}(b')$ reste identique (Proposition 4).

Proposition 4. *Soit T un ensemble de tuiles topologiques. Soit un brin $b \in \mathcal{B}^T : \forall b' \in \langle \beta_1 \rangle(b) \mid \beta'_2(b') \in \langle \beta_1 \rangle(\beta'_2(b))$,*

$$Oracle(\text{région}(b'), \text{région}(\beta'_2(b'))) = Oracle(\text{région}(b), \text{région}(\beta'_2(b))).$$

Démonstration. Tous les brins d'une même orbite $\langle \beta_1 \rangle$ appartiennent à une même région. Par conséquent,

- $b' \in \langle \beta_1 \rangle(b) \Rightarrow \text{région}(b') = \text{région}(b)$;
- $\beta'_2(b') \in \langle \beta_1 \rangle(\beta'_2(b)) \Rightarrow \text{région}(\beta'_2(b')) = \text{région}(\beta'_2(b))$.

Ainsi, $Oracle(\text{région}(b'), \text{région}(\beta'_2(b'))) = Oracle(\text{région}(b), \text{région}(\beta'_2(b)))$. \square

Dans notre cas, nous nous intéressons uniquement aux bords fictifs introduits sur le bord des tuiles. Nous proposons ainsi de restreindre la définition de bord fictif (Définition 48) aux arêtes appartenant au bord d'une tuile.

Définition 48 (Bord fictif). *Soit T un ensemble de tuiles topologiques. Une arête a est un bord fictif si et seulement si il existe deux tuiles adjacentes t et t' et un brin $b \in \mathcal{B}_\partial^{t, t'}$ tels que $a = (b, \beta_2(b))$ et que $Oracle(\text{région}(b), \text{région}(\beta'_2(b)))$ est *vrai*.*

Un bord qui n'est pas un bord fictif est un *bord réel*. Notons que notre définition des bords fictifs est cohérente dans la mesure où deux arêtes équivalentes représentent un bord de même nature (Proposition 5) : si une

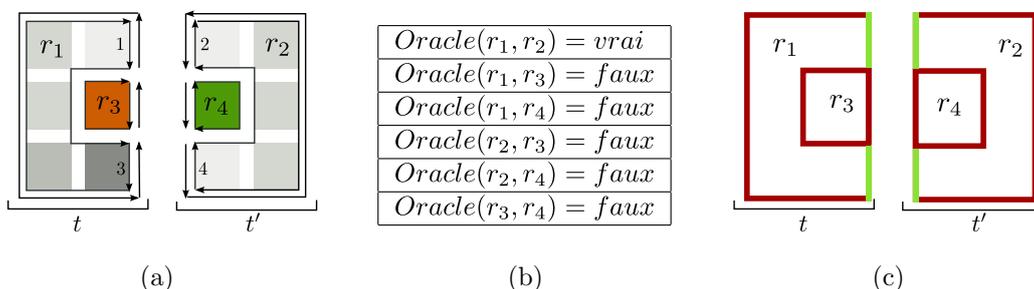


FIGURE 2.6 – Partitionnement des bords d'un ensemble de tuiles en bords réels et fictifs d'après un *Oracle*. (a) Deux tuiles topologiques adjacentes t et t' ; (b) Définition d'un *Oracle* symétrique sur les régions; (c) Bords réels (segments sombres) et bords fictifs (segments clairs).

arête représente un bord fictif (*resp.* réel), toute arête équivalente représente un bord fictif (*resp.* réel).

Proposition 5. *Soit T un ensemble de tuiles topologiques. Soient t et t' deux tuiles adjacentes de T et un brin $b \in \mathcal{B}_\beta^{t,t'}$. L'arête $\langle \beta_2 \rangle(b)$ code un bord fictif si et seulement si l'arête $\langle \beta_2 \rangle(\beta'_2(b))$ code un bord fictif.*

Démonstration. Soit $b \in \mathcal{B}^T$. D'après la Proposition 3, β'_2 est une involution. De plus, le prédicat *Oracle* est symétrique donc :

$$Oracle(\text{région}(b), \text{région}(\beta'_2(b))) = Oracle(\text{région}(\beta'_2(b)), \text{région}(\beta'^2_2(b))).$$

Par conséquent, $\langle \beta_2 \rangle(\beta'_2(b))$ et $\langle \beta_2 \rangle(b)$ codent un bord de même nature. \square

La Figure 2.6 illustre la définition du prédicat *Oracle* sur l'exemple de deux régions multi-adjacentes r_1 et r_2 appartenant à deux tuiles différentes t et t' . D'après la Proposition 4 nous obtenons $Oracle(\text{région}(1), \text{région}(2)) = Oracle(\text{région}(3), \text{région}(4))$ et donc d'après la Proposition 5, les arêtes $\{1, \beta_2(1)\}$, $\{2, \beta_2(2)\}$, $\{3, \beta_2(3)\}$ et $\{4, \beta_2(4)\}$ codent un bord de même nature (fictif dans cet exemple).

La définition des bords fictifs nous permet de proposer une partition de l'ensemble \mathcal{B}^T en deux sous-ensembles \mathcal{B}_R^T et \mathcal{B}_F^T désignant respectivement les brins de \mathcal{B}^T incidents à un bord réel et les brins incidents à un bord fictif (ou non réel).

Définition 49. *Soit T un ensemble de tuiles topologiques. L'ensemble \mathcal{B}_R^T des brins de T incidents à un bord fictif est le sous-ensemble de \mathcal{B}^T tel que :*

$$\mathcal{B}_R^T = \{b \in \mathcal{B}^T \mid (b, \beta_2(b)) \text{ code un bord fictif}\}.$$

Définition 50. Soit T un ensemble de tuiles topologiques. L'ensemble \mathcal{B}_R^T des brins de T incidents à un bord réel est le complémentaire de \mathcal{B}_R^T dans \mathcal{B}^T :

$$\mathcal{B}_R^T = \mathcal{B}^T \setminus \mathcal{B}_R^T.$$

Notons que la Proposition 5 nous indique que la permutation β'_2 préserve le caractère réel ou fictif des brins (Corollaire 1).

Corollaire 1. Soit T un ensemble de tuiles topologiques.

- $\beta'_2(\mathcal{B}_R^T) \subset \mathcal{B}_R^T$;
- $\beta'_2(\mathcal{B}_R^T) \subset \mathcal{B}_R^T$.

Démonstration. Soit $b \in \mathcal{B}_R^T$. Alors, $\langle \beta_2 \rangle(\beta'_2(b))$ code un bord fictif. D'après la Proposition 5, $\langle \beta'_2 \rangle(b)$ code un bord fictif et $\beta'_2(b) \in \mathcal{B}_R^T$. De la même façon, $\forall b \in \mathcal{B}_R^T$, $\langle \beta_2 \rangle(b)$ code un bord réel et donc $\beta'_2(b) \in \mathcal{B}_R^T$. \square

De la sorte, l'ensemble de brins \mathcal{B}_R^T représente uniquement les frontières réelles d'un ensemble T de tuiles topologiques.

2.3.2 Définition d'opérateurs

De façon à pouvoir identifier une région répartie sur plusieurs tuiles, nous introduisons deux nouveaux opérateurs δ_1 et δ_2 définis sur \mathcal{B}_R^T . Intuitivement, nous cherchons à reproduire le comportement des opérateurs β_1 et β_2 des cartes topologiques sur un ensemble de tuiles topologiques dont les bords fictifs sont ignorés.

Opérateur δ_2

L'opérateur δ_2 se construit comme la restriction de l'opérateur β'_2 à l'ensemble \mathcal{B}_R^T . Par conséquent, étant donné un brin b , l'opérateur δ_2 permet d'obtenir le brin homologue sur la face opposée mais ceci indépendamment du découpage en tuiles.

Définition 51 (Opérateur δ_2). Soit T un ensemble de tuiles topologiques. L'opérateur δ_2 est la restriction à \mathcal{B}_R^T de l'opérateur β'_2 : $\delta_2 = \beta'_2|_{\mathcal{B}_R^T}$.

Proposition 6 (Involution δ_2). Soit T un ensemble de tuiles topologiques. L'opérateur δ_2 est une involution sur \mathcal{B}_R^T .

Démonstration. L'opérateur δ_2 est bien défini de \mathcal{B}_R^T dans \mathcal{B}_R^T du fait du Corollaire 1. Puisque β'_2 est une involution (Proposition 3) et que δ_2 est la restriction de β'_2 à \mathcal{B}_R^T , δ_2 est une involution. \square

$$\begin{array}{ccc}
 \beta_2 : \bigcup_{t \in T} \mathcal{B}^t & \longrightarrow & \bigcup_{t \in T} \mathcal{B}^t \\
 \downarrow & & \downarrow \\
 \beta'_2 : \mathcal{B}^T & \longrightarrow & \mathcal{B}^T \\
 \downarrow & & \downarrow \\
 \delta_2 : \mathcal{B}_R^T & \longrightarrow & \mathcal{B}_R^T
 \end{array}
 \quad
 \begin{array}{l}
 \mathcal{B}^T = \bigcup_{t \in T} \mathcal{B}^t \setminus \bigcup_{t' \text{ adj } t} \mathcal{B}_\infty^{t,t'} \\
 \mathcal{B}_R^T = \mathcal{B}^T \setminus \mathcal{B}_R^T
 \end{array}$$

FIGURE 2.7 – Restrictions successives des domaines de définition des différentes involutions d'un ensemble T de tuiles topologiques.

Nous présentons dans la Figure 2.7 un récapitulatif des domaines de définition pour les différentes involutions β_2 , β'_2 et δ_2 introduites sur un ensemble de tuiles.

Notons que les trois involutions β_2 , β'_2 et δ_2 sont définies sur des ensembles de brins successivement réduits depuis l'ensemble de tous les brins de toutes les tuiles (domaine de définition de β_2), à l'ensemble des brins n'appartenant pas à la région infinie entre deux tuiles adjacentes (β'_2) puis à l'ensemble des brins incidents à un bord réel (δ_2), ce qui se traduit par la relation suivante :

$$\mathcal{B}_R^T \subseteq \mathcal{B}^T \subseteq \bigcup_{t \in T} \mathcal{B}^t.$$

Opérateur δ_1

Nous définissons l'opérateur δ_1 de façon à pouvoir parcourir un bord d'une région potentiellement répartie sur plusieurs tuiles. Nous cherchons ainsi à copier le comportement de l'opérateur β_1 en étendant son domaine d'application à l'ensemble de brins des tuiles adjacentes. La construction de l'opérateur δ_1 (Définition 52) repose sur la notion de bord fictif : étant donné un brin b , nous définissons $\delta_1(b) = \beta_1(b)$ si le brin $\beta_1(b)$ appartient à un bord réel. Dans le cas contraire ($\beta_1(b)$ appartient à un bord fictif), nous utilisons le cycle $(\beta_1 \circ \beta'_2)^*$ pour déterminer le premier brin appartenant à un bord réel et incident au sommet codé par $\beta_1(b)$. Dans la mesure où le degré des noeuds est borné dans une carte topologique (à 4 dans le cas d'une grille carrée), le calcul de l'image d'un brin par δ_1 implique un nombre borné d'itérations et a donc

une complexité maximale bornée. De cette façon, nous supposons que le calcul de l'image d'un brin par δ_1 s'effectue en temps constant.

Définition 52 (Opérateur δ_1). *Soit T un ensemble de tuiles topologiques. L'opérateur δ_1 , défini sur \mathcal{B}_R^T , est tel que :*

$$\begin{aligned} \delta_1 : \mathcal{B}_R^T &\rightarrow \mathcal{B}_R^T \\ b &\mapsto \beta_1 \circ (\beta_1 \circ \beta'_2)^k(b) \end{aligned}$$

avec $k = \min\{p \in \mathbb{N} \mid \beta_1 \circ (\beta_1 \circ \beta'_2)^p(b) \in \mathcal{B}_R^T\}$

Différentes configurations peuvent être rencontrées au niveau du bord d'une tuile. Ces cas de figure se limitent à trois dans le cadre d'utilisation d'une grille de pixels 4-connexes. En effet, considérons un sommet s et une arête a , incidente à s et codant un bord réel. Il existe au moins une seconde arête incidente à ce même sommet et codant un bord réel sinon a est une arête pendante (ce qui est contradictoire avec la définition de notre modèle). Étant donné un bord réel, nous avons alors un maximum de $C_3^2 = 3$ combinaisons de deux bords fictifs parmi trois bords incidents.

Considérons par exemple le cas de l'intersection entre quatre tuiles illustré Figure 2.8. Nous considérons comme un même sommet les quatre sommets représentés par un disque. Soit une fonction *Oracle* valant *vrai* si les deux régions ont la même étiquette. Supposons que le brin 1 soit incident à un bord réel. Notre définition de l'opérateur δ_1 nous permet d'obtenir le brin suivant incident à un bord réel appartenant à la même région répartie sur plusieurs tuiles. En effet, dans l'exemple proposé, nous avons $\beta_1 \circ (\beta_1 \circ \beta'_2)^0(1) = 2$, $\beta_1 \circ (\beta_1 \circ \beta'_2)^1(3) = 6$ et $\beta_1 \circ (\beta_1 \circ \beta'_2)^2(7) = 12$. Dans la Figure 2.8(a), le brin 2 appartient à un bord réel donc $\delta_1(1) = 2$ et $k = 0$. Dans la Figure 2.8(b), le brin 4 appartient à un bord fictif et le brin 6 à un bord réel donc $\delta_1(3) = 6$ et $k = 1$. Dans la Figure 2.8(c), les brins 8 et 10 appartiennent à un bord fictif et le brin 12 à un bord réel donc $\delta_1(7) = 12$ et $k = 2$.

Proposition 7 (Permutation δ_1). *Soit T un ensemble de tuiles topologiques. L'opérateur δ_1 est une permutation sur les brins appartenant à un bord réel de l'image (\mathcal{B}_R^T).*

Démonstration. Puisque l'opérateur δ_1 est défini de \mathcal{B}_R^T sur \mathcal{B}_R^T , démontrer que δ_1 est une permutation sur \mathcal{B}_R^T revient à démontrer que δ_1 est injective. Considérons deux brins b et b' tels que $\delta_1(b) = \delta_1(b')$. Alors, il existe $(k, k') \in \mathbb{N} \times \mathbb{N}$ tel que $(\beta_1 \circ \beta'_2)^k(b) = (\beta_1 \circ \beta'_2)^{k'}(b')$. Si $k = k'$, alors $b = b'$ car $\beta_1 \circ \beta'_2$ est une permutation sur \mathcal{B}^T . Dans le cas contraire, considérons $k > k'$. Alors, $(\beta_1 \circ \beta'_2)^{k-k'}(b) = b'$ et b' appartient à un bord réel donc k n'est pas le plus petit

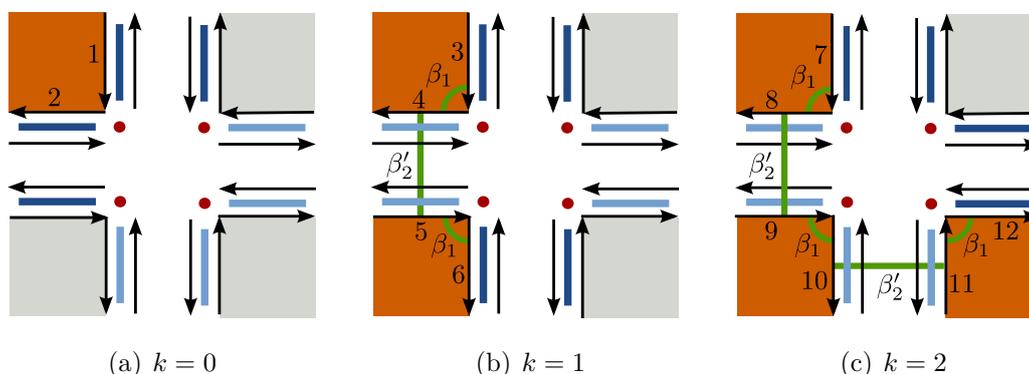


FIGURE 2.8 – Les trois différentes configurations possibles entre bords réels et bords fictifs autour d’un sommet entre quatre tuiles adjacentes (chaque pixel appartient à une tuile différente). Les segments de couleur sombre (*resp.* claire) représentent des bords réels (*resp.* fictifs).

entier tel que $(\beta_1 \circ \beta'_2)^k(b)$ appartienne à \mathcal{B}_R^T , ce qui est contradictoire avec la définition de k . Par conséquent, δ_1 est une permutation sur \mathcal{B}_R^T . \square

2.3.3 Définition du modèle de cartes tuilées

À partir des deux définitions des opérateurs δ_2 et δ_1 (Définition 51 et Définition 52), nous établissons un nouveau modèle pour la représentation d’un ensemble de tuiles topologiques (Définition 53) qui nous permet d’abstraire le découpage géométrique.

Définition 53 (Carte tuilée). *Soit T un ensemble de tuiles topologiques. Une carte tuilée est un triplet $C = (\mathcal{B}_R^T, \delta_1, \delta_2)$ où :*

- δ_1 est une permutation sur \mathcal{B}_R^T définie d’après la Définition 52 ;
- δ_2 est une involution sur \mathcal{B}_R^T définie d’après la Définition 51.

L’ensemble \mathcal{B}_R^T muni des deux opérateurs δ_1 et δ_2 définit ainsi une carte combinatoire (Définition 27). De la même façon que l’opérateur $\beta_0 = \beta_1^{-1}$ est défini pour des raisons pratiques dans le cadre des cartes combinatoires, nous notons δ_0 la permutation inverse de δ_1 .

L’abstraction du découpage géométrique introduit par les tuiles permet de considérer d’une façon globale les sommets, arêtes et faces répartis sur plusieurs tuiles (et qui correspondent donc respectivement à un ensemble de sommets, d’arêtes et de faces appartenant à différentes tuiles). En outre, un sommet (*resp.* une arête, une face) est codé dans une carte tuilée par un cycle $(\delta_1 \circ \delta_2)^*$ (*resp.* δ_2^* , δ_1^*) comme illustré dans la Figure 2.9.

2.4. Construction d'une première partition

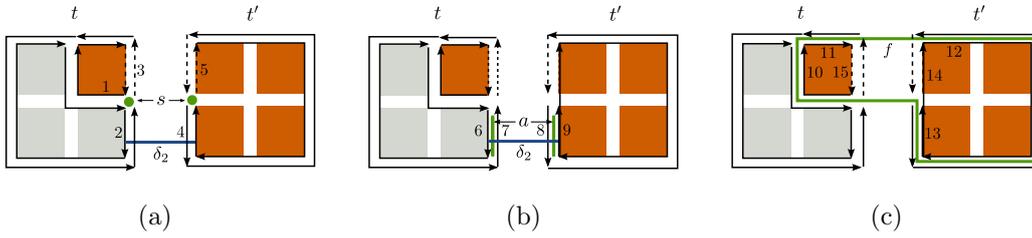


FIGURE 2.9 – Cartes tuilées : représentation des sommets, arêtes et faces. Les brins en pointillés appartiennent à un bord fictif. (a) Caractérisation du sommet $s = (1, 2)$ par le cycle $(\delta_1 \circ \delta_2)^*(1)$. Le sommet s correspond aux deux sommets $(\beta_1 \circ \beta_2)^*(1) = (1, 2, 3)$ appartenant à t et $(\beta_1 \circ \beta_2)^*(4) = (4, 5)$ appartenant à t' dont les plongements équivalents sont représentés par deux cercles pleins ; (b) Caractérisation de l'arête $a = (6, 9)$ par le cycle $\delta_2^*(6)$. L'arête a correspond aux deux arêtes $\beta_2^*(6) = (6, 7)$ appartenant à t et $\beta_2^*(9) = (9, 8)$ appartenant à t' ; (c) Caractérisation de la face $f = (10, 11, 12, 13)$ par le cycle δ_1^* . La face f correspond aux deux faces $\beta_1^*(10) = (10, 11, 15)$ appartenant à t et $\beta_2^*(12) = (12, 13, 14)$ appartenant à t' .

En raison de la décomposition en tuiles, une frontière peut être répartie sur plusieurs tuiles. D'un point de vue local, les cartes combinatoires des tuiles sont minimales, excepté au niveau des bords. Ainsi, nous perdons la propriété qu'une frontière d'une carte tuilée est un unique couple $(b, \delta_2(b))$. En revanche, comme le plongement des sommets de degré 2 appartient aux bords des tuiles, nous obtenons le fait qu'une frontière est une séquence d'arêtes séparées par des sommets de degré 2. Par exemple, dans la Figure 2.9(a), la frontière séparant les deux faces est composée des deux arêtes $(1, \delta_2(1))$ et $(2, \delta_2(2))$, séparées par le sommet s de degré 2.

2.4 Construction d'une première partition

Différents algorithmes ont été proposés pour la construction d'une carte topologique à partir d'une partition. En effet, il est envisageable de construire une carte topologique de niveau 0 (Section 1.3.4) puis de la simplifier. Cette approche dite *naïve* implique une étape du processus comportant un nombre maximal de brins ce qui peut conduire à des problèmes de mémoire pour stocker la structure de données. Ainsi, diverses méthodes d'extraction incrémentales ont été étudiées afin d'éviter cette étape initiale coûteuse.

L'approche proposée par Brun *et al.* [BMD03] se base sur une construction itérative du modèle de carte topologique pour représenter une partition : chaque

étape garantit la cohérence du modèle de carte topologique, ce qui entraîne une mise à jour obligatoire des relations d'imbrication des régions après chaque opération de création ou de suppression de région.

Un algorithme d'extraction linéaire a été proposé par Braquelaire et Domenger [BD99] permettant d'extraire une carte topologique. Une région initiale correspondant à un étiquetage faible de l'image est raffinée en deux phases : la détection des éléments interpixel frontières lors d'un balayage des pixels de l'image et la construction de la carte en étudiant le voisinage des pointels frontières.

L'algorithme d'extraction proposé par Damiand *et al.* [DBF04] permet de construire le modèle de carte topologique en un seul balayage de l'image. De plus, cette méthode est incrémentale et permet de construire indifféremment les trois niveaux de représentation d'une carte topologique présentés dans la Section 1.3.4.

Nous nous proposons d'adapter ce dernier algorithme d'extraction au modèle des cartes topologiques tuilées. Nous rappelons dans la Section 2.4.1 les étapes principales de l'algorithme original. Dans la Section 2.4.2, nous détaillons l'opération nécessaire à l'adaptation de l'algorithme au modèle tuilé, à savoir, comment effectuer les coutures par β'_2 entre deux tuiles adjacentes. Enfin, dans la Section 2.4.3, nous proposons un algorithme s'appuyant sur les deux opérations précédentes, permettant d'extraire une carte topologique tuilée et ainsi créer une première partition de l'image.

2.4.1 Extraction incrémentale de la carte topologique d'une tuile

L'extraction incrémentale de la carte topologique d'une tuile utilise l'algorithme d'extraction proposé par Damiand *et al.* [DBF04] qui consiste à créer les trois modèles (matrice des éléments interpixel, arbre d'imbrication des régions et carte combinatoire) à partir d'une image. La construction de la matrice des éléments interpixel consiste à allouer un tableau de la taille de l'image (Section 1.2.3). Le calcul de l'arbre d'imbrication des régions nécessite une liste ordonnée de l'ensemble des régions de la carte et la carte minimale associée.

La construction de la carte combinatoire est effectuée de façon incrémentale lors du parcours de tous les pixels de l'image, ligne par ligne, du pixel en haut à gauche au pixel en bas à droite. Cette opération est décrite par l'Algorithme 1. Une fois la carte construite, l'Algorithme 2 construit l'arbre d'imbrication des régions.

Carte combinatoire

L'Algorithme 1 d'extraction d'une carte topologique [DBF04] permet une construction incrémentale de la carte combinatoire d'une tuile sans passer par le niveau 0 qui est trop coûteux en mémoire. Le principe de cette méthode consiste à effectuer un balayage des pixels de l'image. Pour chaque pixel p , une région élémentaire (Section 1.3.4) est créée et reliée aux deux régions élémentaires voisines de gauche et du dessus (ligne 1). Un prédicat *Oracle* : $\mathcal{R} \times \mathcal{R} \rightarrow \{\text{vrai}, \text{faux}\}$ (où \mathcal{R} est l'ensemble des régions de l'image) détermine si ces deux bords doivent être conservés (ligne 2, ligne 3). Une phase de simplification supprime les sommets internes de degré 2 pour assurer la minimalité de la carte (ligne 4). Cette étape de simplification est restreinte aux sommets internes qui n'appartiennent pas au bord de la tuile pour optimiser la connexion avec une tuile adjacente.

Algorithme 1 : extraire_carte_d_une_tuile

Données : Une tuile t de largeur l et de hauteur h .

Un *Oracle*.

Résultat : La carte topologique correspondant à t dont le bord n'est pas simplifié.

pour $j = 0$ à h **faire**

pour $i = 0$ à l **faire**

$p \leftarrow p(i, j)$;

$p' \leftarrow p(i - 1, j)$;

$p'' \leftarrow p(i, j - 1)$;

$r' \leftarrow \text{région}(p')$;

$r'' \leftarrow \text{région}(p'')$;

1 construire la région élémentaire r correspondant à p ;

2 **si** *Oracle*(r, r') *est vrai* **alors**

 | | supprimer l'arête entre p et p' ;

3 **si** *Oracle*(r, r'') *est vrai* **alors**

 | | supprimer l'arête entre p et p'' ;

4 simplification des sommets internes de degré 2;

Étant donné que chaque pixel n'est parcouru qu'une seule fois lors du balayage de l'image, la complexité de l'Algorithme 1 est linéaire en fonction du nombre de pixels [DBF04]. L'opération d'extraction d'une carte topologique permet d'obtenir un modèle complet de représentation d'image comportant l'ensemble des relations topologiques et la géométrie des frontières des régions. Le prédicat *Oracle* permet de construire une partition en utilisant par exemple

un critère de segmentation tel que : $Oracle(r_1, r_2) = vrai$ pour tout couple (r_1, r_2) de régions ayant une même couleur moyenne. Enfin, le fait de garder des sommets de degré 2 au niveau des bords lors de l'extraction est utilisé dans l'étape suivante de connexion des tuiles adjacentes en fournissant une décomposition en brins élémentaires, garantissant l'existence d'une bijection $\varphi^{t,t'}$ entre $\mathcal{B}_\partial^{t,t'}$ et $\mathcal{B}_\partial^{t',t}$.

Arbre d'imbrication des régions

L'Algorithme 2 permet de construire l'arbre des régions représentant la relation d'imbrication d'une carte topologique minimale C dont les brins sont étiquetés par région. L'algorithme prend en paramètre la liste ordonnée R des régions de C , triée par ordre croissant suivant le pixel le plus en haut à gauche de la région (noté *premier pixel*). Cet ordre correspond à l'ordre de découverte des régions dans la carte (Algorithme 1). La racine de l'arbre correspond à la région infinie de C (r_∞). La liste R est initialement composée de régions n'ayant ni région englobante, ni région imbriquée. L'idée principale de l'algorithme est de considérer que la première région $r \in R$ qui n'a pas été traitée est la première région d'une composante connexe imbriquée dans une région englobante $père(r)$ (ligne 1). Les relations d'imbrications entre les régions sont ensuite définies pour chaque région r' appartenant à la composante connexe de r privée de $père(r)$. Chaque composante connexe de brins est parcourue une seule fois conduisant à une complexité linéaire suivant le nombre de brins de la carte C . La preuve de la complexité de l'algorithme est proposée en 2D par Damiand *et al.* [DBF04].

2.4.2 Connexion de tuiles adjacentes

Considérons deux tuiles topologiques adjacentes t et t' . Par connexion de tuiles, nous entendons la construction de la bijection β'_2 entre les brins appartenant à $\mathcal{B}_\partial^{t,t'}$ et $\mathcal{B}_\partial^{t',t}$ mettant en relation les arêtes équivalentes. L'opération de connexion entre deux tuiles adjacentes se résume en trois étapes principales détaillées dans l'Algorithme 3.

La première étape établit une bijection entre $\mathcal{B}_\partial^{t,t'}$ et $\mathcal{B}_\partial^{t',t}$ en s'appuyant sur le fait que les bords des tuiles t et t' n'ont pas été simplifiés lors de l'opération d'extraction. Les deux ensembles $\mathcal{B}_\partial^{t,t'}$ et $\mathcal{B}_\partial^{t',t}$ sont alors composés de brins élémentaires.

La deuxième étape effectue les coutures par β'_2 lors d'un parcours en parallèle des brins de $\mathcal{B}_\partial^{t,t'}$ et de $\mathcal{B}_\partial^{t',t}$. Ce parcours est établi de façon géométrique pour chaque couple de brins équivalents. En pratique, nous

2.4. Construction d'une première partition

Algorithme 2 : Construction de l'arbre d'imbrication des régions

Données : Une carte combinatoire minimale C ;

La liste triée R des régions de C .

Résultat : Construction de l'arbre d'imbrication des régions.

$père(r_\infty) \leftarrow r_\infty$;

pour chaque région $r \in R$ telle que $père(r) = nil$ **faire**

1	$b \leftarrow rep(r)$; $père(r) \leftarrow région(\beta_2(b))$; pour chaque brin $b' \in \langle \beta_1, \beta_2, \rangle(b)$ faire <table style="border-collapse: collapse; margin-left: 2em;"> <tr> <td style="border-right: 1px solid black; padding-right: 5px;"></td> <td style="padding-left: 5px;">$r' \leftarrow région(b')$;</td> </tr> <tr> <td style="border-right: 1px solid black; padding-right: 5px;"></td> <td style="padding-left: 5px;">si $père(r') = nil$ alors</td> </tr> <tr> <td style="border-right: 1px solid black; padding-right: 5px;"></td> <td style="padding-left: 5px;"> <table style="border-collapse: collapse; margin-left: 2em;"> <tr> <td style="border-right: 1px solid black; padding-right: 5px;"></td> <td style="padding-left: 5px;">$père(r') \leftarrow père(r)$;</td> </tr> <tr> <td style="border-right: 1px solid black; padding-right: 5px;"></td> <td style="padding-left: 5px;">ajouter la région r' comme imbriquée dans $père(r')$ dans la même composante connexe que r;</td> </tr> </table> </td> </tr> </table>		$r' \leftarrow région(b')$;		si $père(r') = nil$ alors		<table style="border-collapse: collapse; margin-left: 2em;"> <tr> <td style="border-right: 1px solid black; padding-right: 5px;"></td> <td style="padding-left: 5px;">$père(r') \leftarrow père(r)$;</td> </tr> <tr> <td style="border-right: 1px solid black; padding-right: 5px;"></td> <td style="padding-left: 5px;">ajouter la région r' comme imbriquée dans $père(r')$ dans la même composante connexe que r;</td> </tr> </table>		$père(r') \leftarrow père(r)$;		ajouter la région r' comme imbriquée dans $père(r')$ dans la même composante connexe que r ;
	$r' \leftarrow région(b')$;										
	si $père(r') = nil$ alors										
	<table style="border-collapse: collapse; margin-left: 2em;"> <tr> <td style="border-right: 1px solid black; padding-right: 5px;"></td> <td style="padding-left: 5px;">$père(r') \leftarrow père(r)$;</td> </tr> <tr> <td style="border-right: 1px solid black; padding-right: 5px;"></td> <td style="padding-left: 5px;">ajouter la région r' comme imbriquée dans $père(r')$ dans la même composante connexe que r;</td> </tr> </table>		$père(r') \leftarrow père(r)$;		ajouter la région r' comme imbriquée dans $père(r')$ dans la même composante connexe que r ;						
	$père(r') \leftarrow père(r)$;										
	ajouter la région r' comme imbriquée dans $père(r')$ dans la même composante connexe que r ;										

parcourons respectivement par β_0 et β_1 les brins de $\mathcal{B}_\infty^{t,t'}$ et de $\mathcal{B}_\infty^{t',t}$. En effet, si $b \in \mathcal{B}_\infty^t$, alors $\beta_1(b) \in \mathcal{B}_\infty^t$. Comme la décomposition en brins élémentaires de la première étape nous assure qu'à chaque lignel correspond un unique brin, nous pouvons coudre par β'_2 les deux brins de \mathcal{B}^T correspondant à chaque couple de lignels équivalents (ligne 1 de l'Algorithme 3).

La troisième étape simplifie la carte en supprimant les sommets de degré 2 [Dom92, Bru96, BD99, BDM03, DL03]. Notons que la suppression d'un sommet de degré 2 dans une carte tuilée entraîne deux opérations de suppression lorsque son plongement appartient au bord d'une tuile (suppression d'un sommet dans les deux tuiles adjacentes). Cette étape de simplification étant effectuée lors du parcours en parallèle des brins de $\mathcal{B}_\infty^{t,t'}$ et de $\mathcal{B}_\infty^{t',t}$, nous détectons simultanément la présence éventuelle des bords fictifs en évaluant le prédicat *Oracle* sur les couples $(région(\beta_2(b)), région(\beta_2(b')))$ avec $b \in \mathcal{B}_\infty^{t,t'}$ et $b' \in \mathcal{B}_\infty^{t',t}$. Notons qu'une fois les bords fictifs détectés, nous disposons alors des opérateurs δ_1 et δ_2 .

Ces trois étapes sont synthétisées par l'Algorithme 3 et sont respectivement illustrées par les trois sous-figures de la Figure 2.10. L'exemple proposé met en avant trois couples de brins élémentaires (1,2), (3,4), (5,6). Après les opérations de couture par β'_2 , le sommet $(\delta_1 \circ \delta_2)^*(3) = (3, 2)$ de degré 2 appartenant à la carte tuilée est supprimé ce qui revient à supprimer les deux sommets s_1 et s'_1 de degré 2 dans les tuiles adjacentes t et t' .

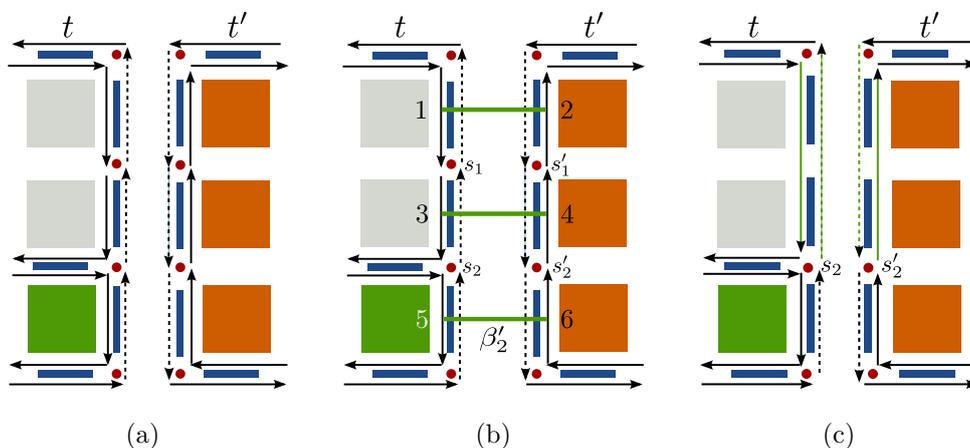


FIGURE 2.10 – Connexion entre deux tuiles adjacentes t et t' . (a) Construction d'une bijection entre les bords des tuiles par décomposition en brins élémentaires; (b) Couture des brins par β'_2 : les trois couples de brins (1,2), (3,4) et (5,6) sont reliés par l'involution β'_2 ; (c) Simplification : la suppression du sommet $\{2,3\}$ de degré 2 de la carte tuilée entraîne la suppression des deux sommets s_1 et s'_1 .

Algorithme 3 : connecter_tuiles

Données : Deux tuiles adjacentes t et t' ayant leurs bords décomposés en arêtes élémentaires.

Résultat : Les brins des bords de t et t' sont reliés par β'_2 .

Les sommets de degré 2 entre t et t' sont supprimés.

$b \leftarrow$ le brin de $\mathcal{B}_\infty^{t,t'}$ tel que $\beta_1(b) \notin \mathcal{B}_\infty^{t,t'}$;

tant que $\beta_0(b) \in \mathcal{B}_\partial^{t,t'}$ **faire**

1 $\left[\begin{array}{l} b' \leftarrow \beta_2 \circ \varphi^{t,t'} \circ \beta_2(b); \\ \text{couture par } \beta'_2 \text{ de } \beta_2(b) \text{ avec } \beta_2(b'); \\ b \leftarrow \beta_0(b); \end{array} \right.$

supprimer les sommets de degré 2 entre t et t' ;

2.4.3 Itération sur toutes les tuiles

En combinant l'algorithme d'extraction (Algorithme 1) et l'algorithme de connexion de tuiles adjacentes (Algorithme 3), nous pouvons proposer une méthode pour extraire une carte tuilée. Cette méthode est décrite par l'Algorithme 4.

L'idée principale est d'extraire chaque tuile topologique t en appliquant l'Algorithme 1 sur la subdivision (i, j) d'une image I (ligne 1). Notons que la simplification des sommets ne s'applique qu'aux sommets internes pour conserver la décomposition du bord de t en arêtes élémentaires. Ensuite, nous connectons la tuile extraite avec ses voisines de gauche et du haut (ligne 2, ligne 4). Les sommets de degré 2 sur les bords des tuiles appartenant aux bords de l'image sont également simplifiés (ligne 3, ligne 5, ligne 6 et ligne 7).

Algorithme 4 : extraire_carte_tuilée

Données : Une image I de largeur L et de hauteur H .

Un *Oracle*.

La largeur l et la hauteur h d'une tuile.

Résultat : La carte tuilée correspondant à I .

```

pour  $j = 0$  à  $H/h - 1$  faire
  pour  $i = 0$  à  $L/l - 1$  faire
1    $t \leftarrow$  extraire_carte_d_une_tuile( $t(i, j)$ , Oracle);
   si il existe une tuile adjacente à gauche à  $t$  alors
2     connecter_tuiles( $t, t(i - 1, j)$ );
   sinon
3     simplifier le bord gauche de  $t$ ;
   si il existe une tuile adjacente en haut de  $t$  alors
4     connecter_tuiles( $t, t(i, j - 1)$ );
   sinon
5     simplifier le bord haut de  $t$ ;
   si  $j = H/h - 1$  alors
6     simplifier le bord bas de  $t$ ;
7   simplifier le bord droit de  $t$ ;

```

Par le biais de l'Algorithme 4, nous avons une solution permettant de réaliser la première étape de construction du modèle de carte tuilée à partir d'une image. Notons qu'une fois que tous les bords d'une tuile ont été connectés, la tuile peut être déchargée de la mémoire et enregistrée sur disque dur. Ce

procédé d'extraction nécessite donc au maximum trois tuiles simultanément en mémoire. Il permet ainsi d'extraire des images de grande taille sans être limité par la contrainte de l'espace mémoire disponible. Néanmoins, l'opération, globale à toute l'image, est peu adaptée au raffinement d'une région donnée. Or, le raffinement d'une région constitue une opération importante pour l'utilisation du modèle. C'est pourquoi nous définissons dans la Section 2.5 les opérations de modifications permettant de raffiner une carte tuilée.

2.5 Raffinement d'une région par un processus de décomposition/fusion

De manière générale, l'extraction d'une carte topologique permet d'obtenir une première partition en définissant un ensemble de régions d'après un critère d'homogénéité donné. Cependant, il est parfois nécessaire de raffiner le résultat obtenu en choisissant un critère plus précis. Deux approches ont été étudiées dans le cadre des cartes topologiques 3D : le raffinement d'une région suivant un guide et le raffinement par un processus de décomposition/fusion [Dup09]. Si nous transposons les opérations en 2D, la division par un guide s'appuie sur un ensemble de $(0, 1)$ -chemin correspondants aux frontières des nouvelles régions à insérer. Le processus de décomposition/fusion applique successivement un éclatement en régions élémentaires et une étape de fusion globale. Nous nous intéressons à cette seconde option.

Dans un premier temps, nous définissons l'opération de raffinement d'une région r par un processus de décomposition/fusion dans le cadre d'une tuile topologique. Le principe général consiste à insérer toutes les arêtes possibles dans r pour créer un ensemble de régions élémentaires puis à les fusionner suivant un critère d'homogénéité appelé *critère de fusion*. Ces deux étapes de décomposition puis de fusion à l'intérieur d'une même tuile font respectivement l'objet de la Section 2.5.1 et de la Section 2.5.2. Étant donné que ces deux opérations sont appliquées à l'intérieur d'une même tuile topologique, nous avons $\delta_1 = \beta_1$ et $\delta_2 = \beta_2$.

Dans un second temps, nous étendons ce processus aux cartes tuilées dans la Section 2.5.3. Nous proposons ainsi un algorithme de décomposition/fusion d'une région, potentiellement répartie sur plusieurs tuiles. Nous étudions également l'impact des décompositions/fusions sur l'arbre d'imbrication des régions et l'algorithme permettant de raffiner globalement une carte tuilée.

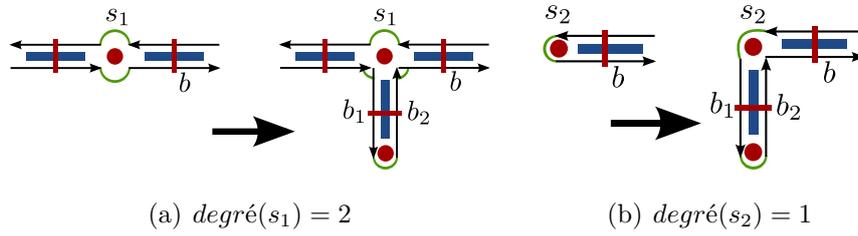


FIGURE 2.11 – Opération d'insertion d'une arête pendante élémentaire $\{b_1, b_2\}$ sur un sommet. (a) Insertion sur le sommet s_1 de degré 2; (b) Insertion sur le sommet s_2 de degré 1.

2.5.1 Première étape : décomposition

L'opération de décomposition transforme une région en un ensemble de régions élémentaires, chacune étant associée à un unique pixel. Cette décomposition repose sur l'opération de base d'insertion d'une arête pendante élémentaire détaillée par l'Algorithme 5 et illustrée par la Figure 2.11. Nous proposons de réaliser la décomposition en propageant des insertions d'arêtes depuis les brins des bords (internes et externes) de la région. Comparée à l'approche intuitive qui consiste à créer une région par pixel lors d'un balayage des pixels de la région, cette méthode nous permet de conserver les relations d'imbrication initiales dans l'arbre des régions. Cette opération est détaillée par l'Algorithme 6 dont les quatre étapes principales sont illustrées dans la Figure 2.12.

Insertion d'une arête pendante élémentaire

L'opération d'insertion consiste à créer une nouvelle arête pendante élémentaire a au niveau d'un sommet s incident à un brin b (Algorithme 5). La première étape crée l'arête pendante élémentaire a composée des deux brins b_1 et b_2 (ligne 1). La deuxième étape allume le lignel l du plongement des deux brins b_1 et b_2 ($\delta_2(b_2) = b_1$) orthogonal à $\text{lignel}(b)$ (ligne 2). Le symbole \perp fait référence au lignel suivant en tournant autour du pointel dans le sens anti-trigonométrique (par exemple, dans la Figure 2.11(a), $\text{lignel}(b)^\perp = \text{lignel}(b_2)$). Enfin, la dernière étape met à jour les coutures des brins concernés (ligne 3). Étant donné que nous travaillons uniquement en 4-connexité, seules deux configurations sont envisageables lors de l'opération d'insertion sur un sommet s d'une arête pendante élémentaire composée d'un unique lignel l : soit l est 0-adjacent à un seul lignel incident à s (sommet de degré 1) ou soit l est 0-adjacent à au moins deux lignels incidents à s (sommet de degré 2 ou 3).

Algorithme 5 : insérer_arête_pendante_élémentaire

Données : Une carte tuilée C .

Un sommet s incident à un brin b .

Résultat : Insertion d'une arête pendante élémentaire a , orthogonale à $\text{lignel}(b)$.

Sorties : Les deux brins de l'arête a insérée.

- 1 créer une arête pendante élémentaire $a = (b_1, b_2)$;
 - 2 allumer $\text{pointel}(b_2)$ et $\text{lignel}^\perp(b)$;
 - 3 couture par β_1 de $\beta_0(b)$ avec b_1 ;
 - 4 couture par β_1 de b_2 avec b ;
- retourner** (b_1, b_2) ;
-

L'opération d'insertion d'une arête pendante élémentaire sur un sommet est illustrée dans la Figure 2.11.

Décomposition d'une région en régions élémentaires

Décomposer une région r en régions élémentaires consiste à créer une région pour chacun des pixels appartenant à r . Pour cela, nous insérons un nombre maximal d'arêtes pendantes élémentaires (Algorithme 5) dans la région suivant un processus de propagation depuis le bord externe de la région. Lorsque toutes les arêtes ont été insérées, nous effectuons les coutures entre les brins ayant un plongement avec un pointel commun.

L'opération décomposant une région r en un ensemble de régions élémentaires est décrite par l'Algorithme 6 :

- **ligne 1** : création d'une liste contenant initialement tous les brins élémentaires correspondants aux bords internes et externes de la région ;
- **ligne 2** : puisque la permutation β_1 connecte deux brins consécutifs suivant une orientation positive autour d'une face, nous nous intéressons uniquement à l^\perp . S'il reste d'autres brins autour du sommet incident à $\beta_2(b)$, ils seront traités ultérieurement au cours des prochaines itérations.
- **ligne 3** : l'opération d'insertion d'arête utilise la procédure décrite par l'Algorithme 5.
- **ligne 4** : si $a = (b, \beta_2(b))$ est une arête pendante, les deux brins de a doivent être correctement cousus. L'idée consiste à rechercher, de façon géométrique, les arêtes perpendiculaires à a comme illustré dans la Figure 2.13. Nous utilisons une structure de tableau associatif¹ associant

1. Comme par exemple la structure *map* de la bibliothèque standard du C++ (*Standard Template Library* en anglais).

2.5. Raffinement d'une région par un processus de décomposition/fusion

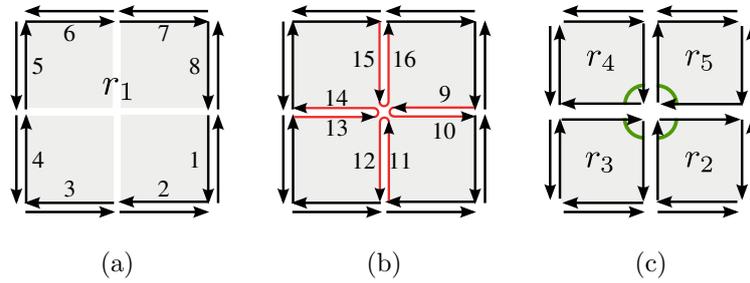


FIGURE 2.12 – Les trois étapes du processus de décomposition d'une région en régions élémentaires. (a) Une région initiale r_1 dont les bords sont décomposés en arêtes élémentaires; (b) Insertion de quatre arêtes pendantes élémentaires pendant le traitement des brins 1, 3, 5 et 7; (c) Opérations de couture sur les arêtes pendantes lors du traitement des brins 9 et 13.

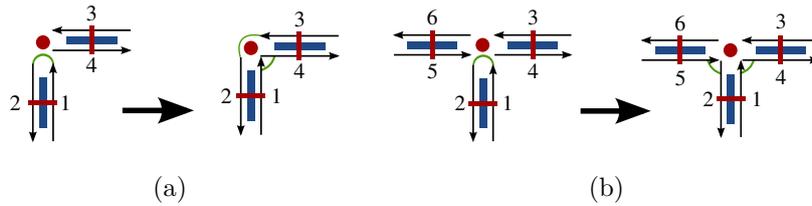


FIGURE 2.13 – Opérations de couture par β_1 (arcs de cercle) lors du traitement de l'arête pendante (1, 2). (a) Une seule arête : 1-couture entre (1, 4) et (3, 2); (b) Au moins deux arêtes : 1-couture entre (1, 4) et (5, 2).

un doublet à un brin. L'opération coud par β_1 les deux brins de a selon qu'il y a une ou deux arêtes perpendiculaires. Le processus garantit qu'il existe au moins une arête perpendiculaire (insérée pendant l'itération précédente traitant $\beta_2(b)$). Si quatre arêtes sont incidentes au même pointel, le traitement de a coud deux arêtes, les deux autres étant cousues ultérieurement (comme dans le cas présenté dans la Figure 2.12).

Le processus de décomposition d'une région en régions élémentaires garantit que chaque lignel à l'intérieur de la région initiale est un lignel frontière. De plus, le processus garantit qu'il ne peut rester aucune arête pendante. En effet, la carte combinatoire est initialement minimale, la région à diviser ne contient aucune arête pendante. Quand une arête pendante est insérée, elle est ajoutée à la *liste* des arêtes à traiter. Ses deux brins incidents sont ensuite correctement cousus lors de l'étape 4 d'une itération ultérieure.

Algorithme 6 : décomposer_région_dans_une_tuile

Données : Une tuile t .

Une région $r \in t$ dont le bord est décomposé en arêtes élémentaires.

Résultat : La région est décomposée en régions élémentaires à l'intérieur de t .

Sorties : L'ensemble des arêtes insérées.

$liste \leftarrow$ liste vide de brins;

$res \leftarrow$ liste vide de brins;

$s \leftarrow$ structure associant un brin à un doublet;

pour chaque brin $b \in r$ **faire**

- 1 | ajouter b à $liste$;
- | insérer $(b, doublet(b))$ dans s ;

tant que $\exists b \in liste \mid b$ n'est pas marqué **faire**

- | marquer b ;
- 2 | $l \leftarrow$ $lignel^\perp(b)$;
- | **si** l n'est pas un lignel frontière **alors**
- 3 | | $(b_i, b_j) \leftarrow$ insérer_arête_pendante_élémentaire(b);
- | | ajouter b_i et b_j à $liste$;
- | | insérer $(b_i, doublet(b_i))$ et $(b_j, doublet(b_j))$ dans s ;
- | | ajouter b_i à res ;
- | **sinon si** $\beta_2(b) = \beta_0(b)$ **alors**
- 4 | | $b' \leftarrow$ brin associé à l dans s ;
- | | couture par β_1 de b' avec b ;

retourner res

2.5.2 Deuxième étape : fusion

La deuxième étape du processus de raffinement d'une région consiste à déterminer parmi les régions résultant d'un processus de décomposition quelles sont celles à fusionner en vue d'obtenir la nouvelle partition. L'opération de fusion est un processus global à la région initiale qui opère sur la *liste* de brins correspondant aux arêtes qui ont été insérées. Étant donné que tout couple de régions adjacentes peut être fusionné, n'importe quelle subdivision de la région initiale peut être représentée.

Initialement, chaque brin b contenu dans la *liste* correspond à une arête $(b, \beta_2(b))$ incidente à un couple de régions adjacentes (r, r') . Pour chaque brin de la liste, nous évaluons par le biais d'un *Oracle* (ou critère de fusion) si les deux régions adjacentes doivent être fusionnées. Deux cas de figure sont

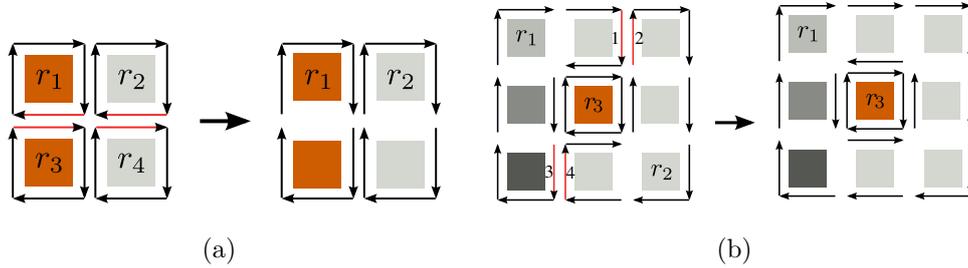


FIGURE 2.14 – Exemples de configurations rencontrées pendant la fusion de deux régions adjacentes. (a) Liste initiale de quatre régions élémentaires résultant de la décomposition d'une même région ; (b) Multi-adjacence entre (r_1, r_2) : si $(1, 2)$ a été supprimée, $(3, 4)$ doit l'être aussi, indépendamment du résultat du critère de fusion.

à différencier : l'adjacence simple et la multi-adjacence. Dans le premier cas, l'arête est supprimée si le critère de fusion est vérifié pour le couple (r, r') (Figure 2.14(a)). En cas de multi-adjacence, plusieurs arêtes séparent r et r' . Notre définition de l'*Oracle* (Section 2.3.1) garantit que le résultat de $Oracle(r, r')$ est le même quel que soit l'arête considérée. Par conséquent, si le critère de fusion détermine la fusion des deux régions au niveau d'une arête, il faut alors supprimer les arêtes communes restantes, indépendamment du résultat du critère sur ces arêtes (Figure 2.14(b)). Ce processus de fusion est illustré par l'Algorithme 7 dont les étapes principales sont les suivantes :

- **ligne 1** : nous utilisons la *liste* des arêtes insérées pour des raisons d'optimisation. En effet, cela évite de parcourir l'ensemble des régions de la carte en nous restreignant aux régions résultant de la découpe d'une même région initiale. Nous pouvons noter qu'il est nécessaire de stocker un seul brin b par arête (l'autre brin étant $\beta_2(b)$) ;
- **ligne 2** : la première condition vise à détecter les cas de multi-adjacence. En effet, prenons l'exemple illustré dans la Figure 2.14(b) et supposons que lors d'une première itération, nous supprimons l'arête $(1, 2)$. Le brin 4 appartenant à r_2 est alors étiqueté comme appartenant à r_1 . Lors du traitement de l'arête $(3, 4)$, nous détectons que $région(3) = région(4)$ et nous supprimons l'arête indépendamment du critère de fusion. La seconde condition utilise un critère de fusion entre les deux régions adjacentes le long de l'arête courante $(b, \beta_2(b))$ pour déterminer si les deux régions doivent être fusionnées ;
- **ligne 3** : extinction du plongement de l'arête incident au brin b dans la matrice des éléments interpixel ;

- **ligne 4** : l'étape de fusion des régions $région(b)$ et $région(\beta_2(b))$ fait appel à une structure d'ensembles disjoints codée par une forêt d'arbres union-find (Section 1.2.1);
- **ligne 5** : l'opération garantit que les deux brins sont supprimés conformément aux opérations de suppression définies par Damiano et Lienhardt [DL03].

Algorithme 7 : fusionner_région_dans_une_tuile

1 **Données** : Une tuile topologique t .
 Un *Oracle*.
 Un ensemble B de brins résultant de la décomposition d'une région r .

Résultat : Les régions satisfaisant le critère *Oracle* sont fusionnées.

pour chaque $b \in B$ **faire**

2 **si** $région(b) = région(\beta_2(b))$ **ou** *Oracle*($région(b)$, $région(\beta_2(b))$) **est**
 vrai alors

3 | éteindre le plongement de b ;

4 | fusionner $région(b)$ et $région(\beta_2(b))$;

5 | supprimer l'arête $(b, \beta_2(b))$;

L'unique contrainte appliquée aux opérations de découpe et de fusion est de préserver les frontières de la région initiale. La fusion est restreinte aux régions générées par l'opération de découpe d'une même région. Par conséquent, à l'intérieur d'une même région initiale, notre opération de fusion n'est soumise à aucune contrainte particulière, hormis le fait de ne pas conserver d'arêtes pendantes. Elle permet donc de créer une région à partir de n'importe quel ensemble connexe de pixels et n'importe quelle partition de la région initiale peut être définie par le processus de décomposition/fusion.

2.5.3 Extension du processus de décomposition/fusion aux cartes tuilées

Nous pouvons intégrer le procédé de décomposition/fusion d'une région dans une tuile à un processus global permettant de raffiner une carte tuilée. Pour cela, nous commençons par étendre ce procédé à une région d'une carte tuilée. Étant donné qu'une région d'une carte tuilée est potentiellement répartie sur plusieurs tuiles, nous détaillons la mise à jour des arbres d'imbrication impactés par ce raffinement. Nous utilisons ensuite notre opération de décomposition/fusion pour le raffinement global d'une carte tuilée.

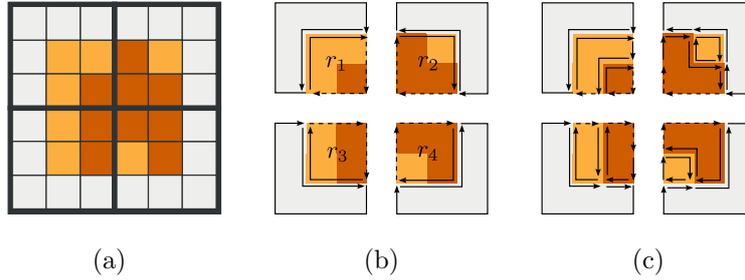


FIGURE 2.15 – Décomposition/fusion d'une région dans une carte tuilée. (a) Image initiale composée de quatre tuiles; (b) Carte tuilée initiale comportant une région répartie sur les quatre tuiles en quatre régions r_1 , r_2 , r_3 et r_4 appartenant chacune à une unique tuile; (c) Décomposition/fusion des quatre régions r_1 , r_2 , r_3 et r_4 .

Décomposition/fusion d'une région dans une carte tuilée

Étant donné qu'une région d'une carte tuilée est un ensemble de régions appartenant à différentes tuiles, une première approche au raffinement d'une région consiste à appliquer directement les deux algorithmes de décomposition et de fusion d'une région dans une tuile (Algorithme 6 et Algorithme 7) en substituant les opérateurs δ_1 et δ_2 aux opérateurs β_1 et β_2 . Cependant, le principal avantage du modèle de carte topologique tuilée est de fournir un modèle adapté à la représentation de grandes images en faisant abstraction de la contrainte technique d'espace mémoire disponible. En effet, chaque tuile pouvant être déchargée sur disque, les opérations de construction et de modification sont réalisables si elles sont incrémentales et ne requièrent qu'un nombre restreint de tuiles à un instant donné. Or, dans le cas du raffinement d'une région par un procédé de décomposition/fusion, la méthode présentée agit de façon globale sur toute la région. Si la région est étendue sur un grand nombre de tuiles, la faisabilité mémoire de l'opération n'est pas garantie. Il est néanmoins possible de réaliser cette découpe de façon incrémentale en considérant l'ensemble des régions appartenant à une seule tuile qui composent la région de la carte tuilée. La définition des opérateurs δ_1 et δ_2 (Section 2.3.3) permet de retrouver cet ensemble de régions qui peut ainsi être enregistré explicitement ou calculé à l'aide d'un parcours du bord extérieur de la région de la carte tuilée par la permutation δ_1 . Enfin, l'utilisation des arbres d'imbrication des différentes tuiles nous permet de retrouver les régions imbriquées.

L'opération de décomposition/fusion d'une région appartenant à une carte tuilée peut s'effectuer morceau par morceau sur chaque région appartenant à une unique tuile, ne nécessitant que deux tuiles simultanément chargées en

mémoire et garantissant ainsi la faisabilité mémoire de l'opération. Ce processus est décrit par l'Algorithme 8 et est illustré dans la Figure 2.15.

Algorithme 8 : Décomposition/fusion d'une région dans une carte tuilée

Données : Une carte tuilée C .

Un *Oracle*.

Une région r de C dont le bord dans chaque tuile est composé d'arêtes élémentaires.

Résultat : La région r est raffinée.

$brins \leftarrow$ liste vide de brins;

pour chaque tuile t contenant r faire

- | | |
|---|---|
| 1 | $r' \leftarrow$ la région de t correspondant à r ; |
| 2 | $brins \leftarrow$ décomposer_région_dans_une_tuile(t, r'); |
| 3 | fusionner_région_dans_une_tuile($t, Oracle, brins$); |
| | détecter les bords fictifs de t ; |
| | décharger t ; |
-

La première étape de l'Algorithme 8 est la décomposition/fusion des régions de r : lors du traitement d'une région r' appartenant à r , seule la tuile t contenant r' est chargée en mémoire. La région r' est raffinée en appliquant successivement l'Algorithme 6 et l'Algorithme 7 (ligne 1 et ligne 2). La deuxième étape, dont le résultat est illustré dans la Figure 2.15(c), consiste à mettre à jour les bords fictifs potentiellement introduits par le raffinement entre la tuile t et ses tuiles adjacentes (ligne 3). Cette étape n'est pas optimisée mais n'est pas coûteuse en pratique : en effet, l'opération est effectuée une première fois pour les bords des tuiles de t contenant un brin de la région à raffiner ainsi qu'une seconde fois lors du traitement de la tuile adjacente. Lors de cette étape, les tuiles voisines de t doivent être chargées en mémoire.

Mise à jour de l'arbre d'imbrication des régions

L'algorithme de construction de l'arbre d'imbrication des régions (Algorithme 2) prend en paramètre la liste ordonnée des régions de la carte et la carte minimale. Pour des raisons de performance et de simplicité, il est préférable de reconstruire l'arbre d'imbrication des régions une fois le processus de raffinement de régions terminé plutôt que de maintenir à jour la structure après chaque création d'une région élémentaire et chaque fusion entre deux régions. Ainsi, la mise à jour de l'arbre après le raffinement d'une région par un processus de décomposition/fusion se déroule en trois étapes :

1. construction initiale de la liste ordonnée des régions ;

2. remplacement de la région raffinée par la liste des nouvelles régions créées ;
3. calcul de l'arbre d'imbrication.

Le fait d'utiliser le *premier pixel* d'une région pour établir le tri permet d'assurer que l'ordre est bien préservé lors du remplacement d'une région par la liste des régions en laquelle elle a été décomposée. Notons que ce processus s'applique aux régions appartenant à une unique tuile et non aux régions de la carte tuilée car la structure d'arbre d'imbrication est locale à chaque tuile.

Raffinement d'une carte tuilée

Le raffinement d'une carte tuilée s'appuie sur la définition d'un prédicat `critère_de_décomposition` : $\mathcal{R} \rightarrow \{vrai, faux\}$ permettant d'évaluer si une région doit être raffinée. Par exemple, nous pouvons envisager de raffiner une région $r = \{r_0, \dots, r_n\}$, avec r_i ($0 \leq i \leq n$) appartenant à une unique tuile s'il existe une région r_i dont le critère de décomposition est *vrai*. Nous proposons alors une méthode en deux étapes.

La première étape utilise le critère de décomposition pour déterminer l'ensemble des régions de la carte tuilée à raffiner lors d'un parcours global des régions de chaque tuile. Cette opération s'effectue de façon incrémentale en chargeant/déchargeant les tuiles suivant un balayage de la tuile en haut à gauche jusqu'à la tuile en bas à droite. Nous utilisons alors le critère de décomposition des régions pour obtenir la liste des régions à raffiner.

La deuxième étape effectue le raffinement de chaque région en utilisant notre processus de décomposition/fusion. De façon à pouvoir conserver les relations d'imbrication, après chaque opération de décomposition/fusion, la liste des régions d'une tuile est mise à jour en remplaçant la région raffinée par l'ensemble des régions résultant de sa décomposition.

Nous proposons une implantation de cette méthode à travers l'Algorithme 9 dont les quatre points principaux sont : la sélection des régions de la carte devant être raffinées (**ligne 1**), l'étape de décomposition/fusion des régions (**ligne 2** et **ligne 3**) et la simplification de la carte (**ligne 4**) puis la reconstruction des arbres d'imbrication (**ligne 5**).

L'étape de décomposition des bords de la région r dans chaque tuile s'appuie sur l'opération d'insertion d'un sommet de degré 2 dans une carte tuilée [Dom92, Bru96, BD99, BDM03, DL03]. Dans le cas où le sommet est inséré sur un brin b appartenant au bord d'une tuile ($\beta_2(b) \neq \beta'_2(b)$) cette opération se traduit par deux insertions de sommets dans les deux tuiles adjacentes pour garantir la validité de la carte tuilée.

Algorithme 9 : Raffinement d'une carte tuilée par décomposition/fusion des régions.

Données : Une carte tuilée minimale C .

Un *Oracle*.

Résultat : Une carte tuilée minimale C' raffinée.

pour chaque région $r \in \mathcal{R}$ **faire**

- 1 **si** *critère_de_décomposition*(r) **est vrai** **alors**
 - 2 décomposer le bord de r dans chaque tuile en arêtes élémentaires;
 - 3 décomposition/fusion(C , *Oracle*, r);
 - 4 supprimer les sommets de degré 2;
 - 5 calculer les arbres d'imbrication pour chaque tuile modifiée;
-

2.6 Structures de données et implantation

Les opérations de chargement et de déchargement permettent de basculer entre la représentation en mémoire et sur disque dur d'une tuile topologique. Ces opérations sont particulièrement importantes car elles sont au cœur de la manipulation d'une structure tuilée. Elles permettent d'assurer qu'un nombre minimal de tuiles est gardé en mémoire. Cela garantit la faisabilité d'un traitement donné sur une image de taille importante. Par conséquent, nous explicitons la structure de données utilisée pour ces opérations.

Dans la Section 2.6.1, nous présentons les différentes structures utilisées pour la représentation sur disque d'une tuile topologique. Dans la Section 2.6.2, nous proposons un premier algorithme pour le déchargement d'une tuile topologique à partir de ces structures. Dans la Section 2.6.3, nous étudions l'opération inverse de chargement en mémoire d'une tuile précédemment enregistrée sur disque dur.

2.6.1 Structures de données

Afin de pouvoir basculer entre la représentation en mémoire et sur disque d'une carte topologique, nous utilisons un identifiant entier, unique et consécutif pour caractériser chaque entité (brins et régions).

Paramètres d'un brin

Chaque brin b est enregistré dans une structure appelée `structure_brin` détaillée dans le Tableau 2.1. Ainsi, étant donné un brin b , ses images par la permutation β_1 et l'involution β_2 sont enregistrées à l'aide des identifiants

TABLE 2.1 – `structure_brin` : enregistrement d'un brin b .

Champ	Description
<code>id</code>	identifiant de b
β_1	identifiant de $\beta_1(b)$
β_2	identifiant de $\beta_2(b)$
<code>doublet</code>	coordonnées de $pointel(b)$ et orientation de $lignel(b)$
<code>région</code>	identifiant de $région(b)$

appliqués aux brins. Le champ `doublet` permet de connaître la direction initiale du plongement du brin en enregistrant le couple $(pointel(b), lignel(b))$. Notons que l'image par β'_2 d'un brin n'est pas sauvegardée. En effet, comme l'opération de connexion des tuiles adjacentes est très peu coûteuse en pratique et que cette opération est utilisée à chaque fois qu'une tuile est chargée en mémoire, nous recalculons les connexions par β'_2 au chargement de la tuile. Enfin, à partir des opérateurs β_1 , β_2 et β'_2 , nous pouvons retrouver les opérateurs δ_1 et δ_2 des cartes tuilées (Définition 51 et Définition 52).

Paramètres d'une région

Chaque région r est enregistrée dans une structure appelée `structure_région` détaillée dans le Tableau 2.2. Les champs enregistrés se décomposent en trois catégories. Tout d'abord, les informations directement relatives à la région r avec les champs `id`, `représentant` et `premier pixel`. Ensuite, des paramètres associés à la région sont enregistrés comme par exemple le nombre de pixels ou la somme des niveaux de gris des pixels de r avec le champ `paramètres`. Enfin, les trois champs `père`, `fils` et `cc` permettent de reconstruire l'arbre des relations d'imbrication entre les régions en indiquant respectivement le parent, le fils et la composante connexe suivante d'une région dans l'arbre.

Paramètres d'une tuile

Nous définissons une troisième structure appelée `structure_tuile` encapsulant les structures pour les brins et les régions d'une tuile topologique. Cette structure se présente sous la forme indiquée par le Tableau 2.3. L'`en-tête` stocke la taille des structures et permet donc d'allouer l'espace mémoire nécessaire en conséquence lors du chargement de la tuile. Les coordonnées de la tuile permettent de la positionner dans le référentiel de l'image. Enfin, les

TABLE 2.2 – `structure_région` : enregistrement d'une région r .

Champ	Description
<code>id</code>	identifiant de r
<code>représentant</code>	identifiant du brin représentant $rep(r)$
<code>premier pixel</code>	coordonnées du pixel haut/gauche de r
<code>paramètres</code>	données associées à r
<code>père</code>	identifiant de $père(r)$
<code>fil</code>	identifiant de $fil(r)$
<code>cc</code>	identifiant de $CC(r)$

TABLE 2.3 – `structure_tuile` : enregistrement d'une tuile topologique.

Champ	Description
<code>en-tête</code>	taille en octets des différentes structures
<code>coordonnées</code>	positionnement de t dans l'image
<code>géométrie</code>	matrice de Khalinsky associée à t
<code>brins</code>	<code>structure_brin</code> de l'ensemble des brins de t
<code>régions</code>	<code>structure_région</code> de l'ensemble des régions de t

trois derniers éléments permettent de reconstruire la carte topologique de la tuile lors des opérations de chargement et de déchargement.

2.6.2 Déchargement d'une tuile topologique

Notre modèle de tuile topologique correspond à une carte topologique et donc à une combinaison des trois modèles : carte combinatoire, matrice d'éléments interpixel et arbre d'imbrication des régions. Lors du déchargement d'une tuile, ces trois modèles sont enregistrés sur disque dur pour être reconstruits lors de l'opération de chargement.

En pratique, l'opération de déchargement d'une tuile topologique consiste à enregistrer la `structure_tuile` correspondante sur disque dur avant de supprimer son modèle en mémoire. Si la tuile a déjà été enregistrée une première fois, l'opération de déchargement ne met à jour la `structure_tuile` que si la tuile a été modifiée.

Carte combinatoire

La carte combinatoire correspondant à une tuile est un ensemble de brins muni des deux opérateurs β_1 et β_2 . Puisque la structure brin contient l'identifiant d'un brin et les identifiants de ses images par β_1 et β_2 , la sauvegarde de la carte est assurée par l'enregistrement de tous les brins.

Matrice des éléments interpixel

La matrice des éléments interpixel est stockée explicitement sous la forme de données binaires correspondant à la matrice de Khalimsky. Notons que le plongement des brins est enregistré par le champ `doublet` de la `structure_brin`.

Arbre d'imbrication des régions

L'enregistrement de l'arbre d'imbrication des régions se fait de façon implicite lors de la sauvegarde de l'ensemble des régions de la carte. En effet, la `structure_région` contient les champs `père` et `fils` permettant de reconstruire les relations d'imbrication entre les régions (les nœuds de l'arbre sont les régions, les arêtes sont les relations d'imbrication).

Algorithme 10 : Enregistrement d'une tuile

Données : Une tuile topologique t .

Résultat : La `structure_tuile` s correspondant à la tuile t .

si s n'existe pas ou t est modifiée **alors**

- 1 | ajouter la matrice de Khalimsky de t à s ;
 - | **pour chaque** $brin\ b \in t$ **faire**
 - 2 | | ajouter `structure_brin(b)` à s ;
 - | | **pour chaque** $région\ r \in t$ **faire**
 - 3 | | | ajouter `structure_région(r)` à s ;
 - 4 | | insérer l'en-tête de s ;
-

L'Algorithme 10 illustre l'opération de sauvegarde du modèle de tuile topologique t sous la forme d'un fichier correspondant à la `structure_tuile(t)`. La première étape sauvegarde les frontières des régions (ligne 1), la deuxième étape enregistre la carte combinatoire lors d'un parcours des brins (ligne 2), la troisième enregistre les régions et permet de reconstruire l'arbre des relations d'imbrication (ligne 3) et la dernière étape renseigne la taille de la matrice de

Khalimsky, le nombre de brins et le nombre de régions dans l'en-tête du fichier contenant la `structure_tuile`.

2.6.3 Chargement d'une tuile topologique

L'opération de chargement d'une tuile t va lire le fichier qui lui correspond sur disque et construit une nouvelle tuile d'après ces informations avant de l'ajouter à la carte tuilée. Nous créons d'abord les instances des brins et des régions de la tuile topologique avant de renseigner leurs relations et propriétés. Cette opération ne peut pas se faire en une seule passe car les brins et les régions sont liés entre eux dans les deux sens (une région possède un brin représentant, un brin connaît sa région) : une région doit être créée avant d'être assignée à un brin et vice versa.

Ainsi, dans un premier temps, le nombre de brins et le nombre de régions, stockés dans la `structure_tuile` permettent de créer les instances de toutes les entités. Nous nous servons ensuite de ces instances pour l'initialisation de deux tableaux associatifs associant chaque instance à son identifiant.

Dans un second temps, nous reconstruisons les relations de la carte topologique en renseignant les champs de chaque instance en utilisant au besoin les deux tableaux associatifs précédents pour obtenir l'instance d'un brin ou d'une région correspondant à un identifiant donné.

L'Algorithme 11 détaille l'opération de chargement d'une tuile topologique depuis un fichier. Cette opération se déroule en deux phases. La première consiste à créer les ensembles libres de brins et de régions (`ligne 1` et `ligne 2`). La seconde effectue les connexions entre les entités à travers les opérations d'association (`ligne 3` et `ligne 4`). Ces opérations utilisent la structure de tableau associatif pour récupérer l'entité (brin ou région) correspondant à l'identifiant du champ enregistré. La dernière phase permet d'établir l'involution β'_2 entre les tuiles adjacentes (`ligne 5`).

2.7 Conclusion

Au cours de ce chapitre, nous avons défini une structure de données capable de représenter l'information topologique d'une image tout en étant adaptée au traitement d'images de grande taille. Nous avons ainsi proposé de baser ce modèle sur les cartes topologiques en introduisant une notion de tuile permettant de garantir la faisabilité mémoire de la construction du modèle indépendamment de la taille de l'image. Nous avons ensuite étudié l'impact de la subdivision artificielle en tuiles du modèle. Cette étude nous a amené à

Algorithme 11 : Chargement d'une tuile

Données : Un fichier contenant la `structure_tuile` s d'une tuile topologique t .

Résultat : Une tuile topologique t .

$n \leftarrow$ nombre de brins;

$m \leftarrow$ nombre de régions;

$B \leftarrow$ structure associant un brin à un identifiant;

$R \leftarrow$ structure associant une région à un identifiant;

associer le champ `géométrie` de s à t ;

- 1 **pour** $i = 1$ à n **faire**
 | créer un brin b_i d'identifiant i dans B ;
 - 2 **pour** $i = 1$ à m **faire**
 | créer une région r_i d'identifiant i dans R ;
 - 3 **pour chaque** *brin* b **faire**
 | associer $\beta_1(b)$ à b ;
 | associer $\beta_2(b)$ à b ;
 | associer $\text{région}(b)$ à b ;
 | associer $\text{doublet}(b)$ à b ;
 - 4 **pour chaque** *région* r **faire**
 | associer $\text{rep}(r)$ à r ;
 | associer $\text{père}(r)$ à r ;
 | associer $\text{fils}(r)$ à r ;
 | associer $\text{CC}(r)$ à r ;
 | associer $\text{paramètres}(r)$ à r ;
 | associer $\text{premier pixel}(r)$ à r ;
 - 5 **pour chaque** *tuile voisine* t' de t *chargée en mémoire* **faire**
 | calculer β'_2 entre t et t' ;
-

proposer une solution pour abstraire ce découpage grâce à la notion de bord fictif et aux opérateurs δ_1 et δ_2 . Ce travail nous a ainsi permis de définir le modèle de carte tuilée.

Nous avons ensuite détaillé les opérations de construction de notre modèle en proposant un algorithme d'extraction incrémental. Notre extraction s'appuie sur l'opération de connexion entre deux tuiles adjacentes qui permet d'établir les relations topologiques entre les brins du bord des tuiles. La construction d'une partition nécessite un espace mémoire borné, directement dépendant de la taille d'une tuile. À partir de cette première partition, nous avons proposé un processus global de raffinement d'une carte tuilée par une opération de décomposition/fusion des régions. Enfin, nous avons précisé l'implantation

des opérations de chargement et de déchargement d'une tuile topologique permettant d'obtenir une empreinte mémoire minimale lors des traitements.

Dans le chapitre suivant, nous proposons une extension hiérarchique du modèle de carte tuilée pour la représentation de grandes images multi-résolutions.

Chapitre 3

Pyramides descendantes tuilées

Sommaire

3.1	Introduction	108
3.2	Définition du modèle de pyramide descendante tuilée	109
3.2.1	Pyramides d'images	110
3.2.2	Pyramides de tuiles	116
3.2.3	Pyramides descendantes tuilées	120
3.3	Représentation des relations hiérarchiques	122
3.3.1	Projection des (0,1)-chemins	123
3.3.2	Relations <i>prédécesseur/successeur</i> dans une carte tuilée	127
3.4	Utilisation pratique des relations hiérarchiques	130
3.4.1	Identification des arêtes filles	131
3.4.2	Identification des régions filles	133
3.5	Construction d'une pyramide descendante tuilée	136
3.5.1	Projection des frontières	136
3.5.2	Extraction des successeurs d'une tuile	138
3.5.3	Extraction d'une pyramide	142
3.6	Conclusion	143

3.1 Introduction

Au cours du Chapitre 2, nous avons défini une structure de données capable de représenter l'information topologique et géométrique d'une partition d'une image par le modèle de carte topologique tuilée. Ce modèle est une extension des cartes topologiques adaptée à la représentation de grandes images.

Cependant, les images à traiter sont des images multi-résolution composées de quatre à cinq niveaux dont le plus détaillé peut atteindre une taille de $32\,000 \times 32\,000$ pixels. À chaque niveau, l'analyse des pathologistes par *focus d'attention* distingue différentes zones d'intérêt (comme par exemple les zones tumorales) devant être explorées plus précisément à la résolution supérieure.

En raison de la taille de ces images, certaines structures ne peuvent être perçues qu'à une résolution donnée. Par exemple, les structures tissulaires sont uniquement visibles à basse résolution tandis que les structures cellulaires ne peuvent être observées qu'à la résolution la plus importante.

Or, le modèle des cartes topologiques tuilées ne permet pas d'utiliser des critères hiérarchiques pour segmenter ces images. Nous définissons dans ce chapitre une structure hiérarchique étendant le modèle des cartes tuilées pour construire un modèle qui soit capable de reproduire l'analyse des pathologistes.

Au cours du Chapitre 1, nous avons présenté les différents modèles hiérarchiques existants. Cependant, la plupart de ces modèles adoptent un processus de construction ascendant. La construction de ces modèles repose sur des simplifications successives d'une sur-segmentation de l'image. Afin de garantir un maximum de flexibilité au processus construisant la pyramide, la sur-segmentation initiale correspond souvent à une partition codant chaque pixel par une région.

Dans le cas des images histologiques, il s'agit donc de représenter la partition d'une image de taille $32\,000 \times 32\,000$ pixels. Bien que le modèle de carte tuilée garantisse la faisabilité de la construction d'une telle partition, nous avons préféré adopter une approche descendante pour la construction de notre modèle pour les raisons suivantes :

1. Si nous désirons construire un modèle suivant l'approche des pathologistes alors une construction hiérarchique descendante est la solution naturelle pour reproduire l'analyse par *focus d'attention*.
2. Les différents niveaux de la hiérarchie permettent de représenter les structures qui sont uniquement visibles à une résolution donnée.
3. Étant données les contraintes techniques liées aux besoins en mémoire disponible, une approche descendante permet d'optimiser l'espace mémoire nécessaire et les temps de traitement en ne raffinant que les régions d'intérêt.

4. Chaque structure tissulaire peut se concevoir comme la réunion des structures cellulaires qui la compose. Nous avons donc une structure à la fois multi-résolution et multi-échelle.

Dans ce chapitre, nous commençons par définir dans la Section 3.2 le modèle de pyramide descendante tuilée comme une combinaison de trois modèles hiérarchiques : une pyramide d'images codant différentes résolutions d'une même image, une pyramide de tuiles définissant un pavage régulier et hiérarchique de la pyramide d'images et une pyramide de cartes tuilées construite sur la pyramide de tuiles. Ainsi, ce modèle permet de traiter par blocs de grandes images hiérarchiques tout en étant capable de représenter les relations topologiques d'adjacence et d'imbrication d'une hiérarchie de partitions. Dans la Section 3.3, nous explicitons la manière dont sont représentées les relations hiérarchiques dans notre modèle. Nous détaillons pour cela l'opération de projection des frontières ainsi que les relations *prédécesseur/successeur* entre les brins et les régions. Dans la Section 3.4, nous nous intéressons à l'utilisation pratique des relations hiérarchiques introduites afin de retrouver les arêtes et les régions filles résultant d'une opération de découpe au niveau supérieur. Dans la Section 3.5, nous proposons une méthode de construction descendante de notre modèle. Notre approche s'appuie sur une première étape de projection des frontières entre deux niveaux du modèle et une seconde étape d'extraction de la carte topologique des tuiles. Nous mettons en avant les contraintes hiérarchiques du processus d'extraction et nous présentons une méthode de construction utilisant le principe du focus d'attention.

3.2 Définition du modèle de pyramide descendante tuilée

Afin de proposer un modèle capable de représenter les relations d'adjacence et d'imbrication de grandes images hiérarchiques, nous procédons en trois étapes. Premièrement, nous définissons une pyramide d'images comme une pile d'images de résolution croissante (Section 3.2.1). Deuxièmement, nous associons une pyramide de tuiles à cette pyramide d'images en décomposant chaque niveau suivant un pavage régulier et rectangulaire (Section 3.2.2). Troisièmement, nous définissons le modèle de pyramide descendante tuilée comme la combinaison d'une pyramide d'images, d'une pyramide de tuiles et d'une pyramide de cartes tuilées dont les relations hiérarchiques sont représentées à travers des relations *prédécesseur/successeur* définies entre les brins et les régions (Section 3.2.3).

3.2.1 Pyramides d'images

Nous définissons une pyramide d'images (ou image multi-résolution) comme une suite d'images (I^0, \dots, I^n) de résolution croissante. Soient l^k et h^k la largeur et la hauteur de I^k , $k \in \{0, \dots, n\}$. Nous introduisons alors la notion de *ratio* entre deux niveaux consécutifs en notant r_x^k (*resp.* r_y^k) les rapports l^{k+1}/l^k (*resp.* h^{k+1}/h^k). Nous pouvons remarquer que $r_x^k \geq 1$ et $r_y^k \geq 1$. Enfin, nous nous restreignons au cas des ratios entiers. Nous formalisons ces notions dans la Définition 54. Lorsque le ratio est identique pour toute la pyramide, nous notons ces ratios r_x et r_y .

Définition 54 (Pyramide d'images). *Soit I une image. Une pyramide d'images associée à I est une suite d'images $\mathcal{I} = (I^0, \dots, I^n)$ avec $I^n = I$ et $\forall k, 0 \leq k < n$:*

- I^k est une image de largeur l^k et de hauteur h^k , déduite de I^{k+1} par un processus de décimation [BK03a];
- $r_x^k = l^{k+1}/l^k$ et $r_y^k = h^{k+1}/h^k$ sont deux ratios entiers supérieurs ou égaux à 1.

La construction d'une hiérarchie de partitions peut s'effectuer suivant un processus ascendant à partir de la résolution la plus importante I^n . Il existe différentes possibilités pour construire la résolution I^k à partir de la résolution I^{k+1} . Une première méthode consiste à associer pour tout pixel p de I^k , la couleur et/ou un vecteur d'attributs moyens, calculé récursivement sur le champ récepteur (Définition 34) du pixel p . D'autres approches ont également été envisagées comme par exemple une construction à base d'ondelettes [SRPH10].

Nous cherchons à établir une représentation interpixel d'une pyramide d'images \mathcal{I} . Étant donné que \mathcal{I} est une pyramide régulière, sa représentation interpixel constitue également une structure hiérarchique régulière. Par conséquent, nous proposons d'adapter la notion de fenêtre de réduction (Section 1.4.1) aux ensembles de pointels définis aux différents niveaux d'une pyramide descendante.

La fenêtre de réduction de taille $m \times n$ d'un pixel de niveau k est un ensemble de $m \times n$ pixels au niveau $k + 1$ (voir Définition 33). Du point de vue de la représentation interpixel, la fenêtre de réduction d'une 2-cellule est un sous-complexe cellulaire composé de $m \times n$ 2-cellules (Section 1.1.3).

De la même façon que pour un pixel, nous définissons la fenêtre de réduction de taille $m \times n$ d'un pointel p comme un ensemble de $m \times n$ pointels au niveau suivant (Figure 3.1).

Considérons le niveau I^k d'une pyramide d'images \mathcal{I} . Soient l et h la largeur et la hauteur de I^k . Soit $p \in I^k$ un pointel de niveau k et de coordonnées (i, j) ,

3.2. Définition du modèle de pyramide descendante tuilée

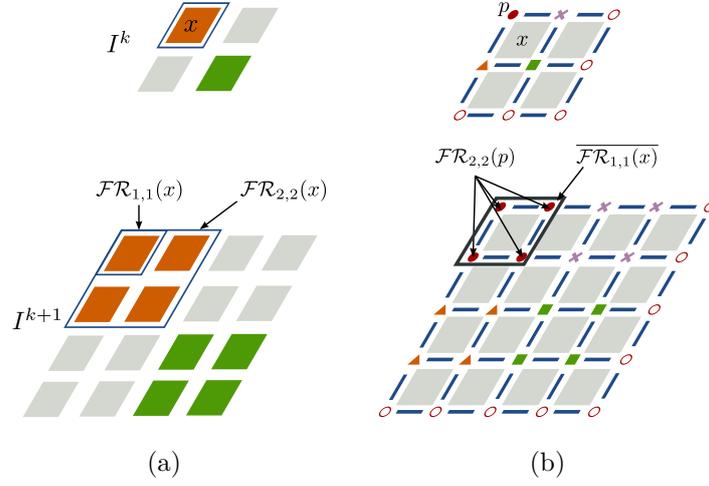


FIGURE 3.1 – Fenêtres de réduction dans une pyramide d’images composée de deux niveaux avec un ratio $r_x^k = r_y^k = 2$. (a) Entre les pixels : une fenêtre de taille $m \times n$ correspond à $m \times n$ pixels au niveau suivant ; (b) Entre les pointels : $\mathcal{FR}_{2,2}(p) = \overline{\mathcal{FR}_{1,1}(x)} \cap \mathcal{P}^{k+1}$. Les quatre symboles disque, carré, croix et triangle permettent d’identifier les pointels du niveau $k + 1$ appartenant à la fenêtre de réduction d’un même parent.

avec $0 \leq i < l$ et $0 \leq j < h$. Soit $x \in I^k$ le pixel de coordonnées (i, j) . D’après notre système de coordonnées (Section 1.2.3), p est le pointel incident en haut à gauche de x . Nous définissons la *fenêtre de réduction* de taille $m \times n$ de p comme l’ensemble des pointels appartenant à l’adhérence de la fenêtre de réduction de taille $(m - 1) \times (n - 1)$ de x .

Pour les pointels appartenant aux bords droit ou bas de l’image ($i = l$ ou $j = h$), il n’y a pas de pixel correspondant dans l’image et nous devons traiter séparément ces cas de figure. Néanmoins, la représentation des éléments interpixel sous la forme d’un tableau de dimensions $(l + 1) \times (h + 1)$ (une matrice de Khalimsky) permet de considérer le pixel x de coordonnées (i, j) . Dans ce cas, nous considérons la fenêtre de réduction d’un pointel ou d’un pixel comme une fenêtre de largeur 1 si $i = l$ et de hauteur 1 si $j = h$ (Figure 3.2).

De la sorte, nous étendons la notion de fenêtre de réduction d’un pixel aux pointels par la Définition 55.

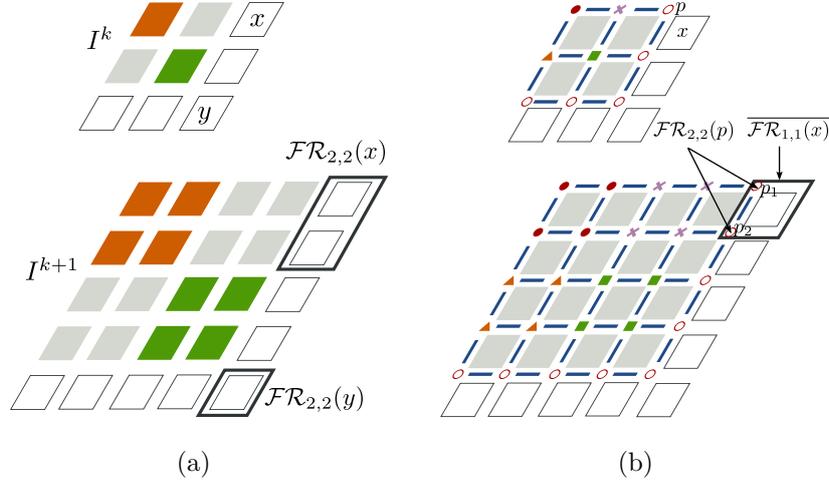


FIGURE 3.2 – Cas particuliers des bords droit et bas pour les fenêtres de réduction sur l'exemple d'une pyramide d'images composée de deux niveaux avec un ratio $r_x^k = r_y^k = 2$. La fenêtre est tronquée dans le cas des bords droit et bas. (a) Entre les pixels : $\mathcal{FR}_{2,2}(x) = \mathcal{FR}_{1,2}(x)$ et $\mathcal{FR}_{2,2}(y) = \mathcal{FR}_{1,1}(y)$; (b) Entre les pointels : $\mathcal{FR}_{2,2}(p) = \{p_1, p_2\}$.

Définition 55 (Fenêtre de réduction d'un pointel). Soit I^k le niveau k de largeur l et de hauteur h d'une pyramide d'images \mathcal{I} . Soit $p \in I^k$ le pointel de coordonnées (i, j) et $x \in I^k$ le pixel de coordonnées (i, j) . Notons \mathcal{P}^k l'ensemble des pointels du niveau I^k . La fenêtre de réduction de taille $m \times n$ de p , notée $\mathcal{FR}_{m,n}(p)$, est telle que :

- si $0 \leq i < l$ et $0 \leq j < h$, $\mathcal{FR}_{m,n}(p) = \overline{\mathcal{FR}_{m-1,n-1}(x)} \cap \mathcal{P}^{k+1}$;
- si $i = l$, $\mathcal{FR}_{m,n}(p) = \overline{\mathcal{FR}_{1,n-1}(x)} \cap \mathcal{P}^{k+1}$;
- si $j = h$, $\mathcal{FR}_{m,n}(p) = \overline{\mathcal{FR}_{m-1,1}(x)} \cap \mathcal{P}^{k+1}$.

La Proposition 8 permet d'obtenir les coordonnées des pointels appartenant à la fenêtre de réduction d'un même pointel.

Proposition 8. Soit $p \in \mathcal{P}^k$ un pointel de niveau k et de coordonnées (i, j) . La fenêtre de réduction de taille $m \times n$ de p est un ensemble d'au plus $m \times n$ pointels de \mathcal{P}^k tel que :

$$\mathcal{FR}_{m,n}(p) = \{p' = (a, b) \in \mathcal{P}^{k+1} \mid ir_x^k \leq a < ir_x^k + m \text{ et } jr_y^k \leq b < jr_y^k + n\}.$$

Démonstration. Supposons $r_x^k > 1$ et $r_y^k > 1$. Soit x le pixel de coordonnées (i, j) du niveau k .

3.2. Définition du modèle de pyramide descendante tuilée

$$\mathcal{FR}_{m,n}(x) = \{x' \in I^{k+1} \mid ir_x^k \leq i' < ir_x^k + m \text{ et } jr_y^k \leq j' < jr_y^k + n\}$$

D'après notre système de coordonnées de la représentation interpixel (Section 1.2.3), les quatre pointels incidents au pixel x sont les pointels de coordonnées (i, j) , $(i + 1, j)$, $(i, j + 1)$ et $(i + 1, j + 1)$. Notons p le pointel de \mathcal{P}^k de coordonnées (a, b) . Ainsi :

$$\forall x \in I^k, \bar{x} \cap \mathcal{P}^k = \{p \in \mathcal{P}^k \mid i \leq a \leq i + 1 \text{ et } j \leq b < j + 1\}$$

Notons que l'adhérence d'un ensemble fini de pixels E est la réunion de l'adhérence de chacun des pixels de E . Dans le cas des fenêtres de réduction :

$$\overline{\mathcal{FR}_{m,n}(x)} = \bigcup_{x' \in \mathcal{FR}_{m,n}(x)} \bar{x}'$$

Afin de simplifier les notations, désignons respectivement par p et x le pointel et le pixel de coordonnées (i, j) du niveau k . Notons x' le pixel de coordonnées (i', j') du niveau $k + 1$ et p' le pointel de coordonnées (a, b) du niveau $k + 1$.

$$\begin{aligned} \mathcal{FR}_{m,n}(p) &= \overline{\mathcal{FR}_{m-1,n-1}(x)} \cap P^{k+1} \\ &= \bigcup_{x' \in \mathcal{FR}_{m-1,n-1}(x)} \bar{x}' \cap \mathcal{P}^{k+1} \\ &= \bigcup_{x' \in \mathcal{FR}_{m-1,n-1}(x)} \{p' \in \mathcal{P}^{k+1} \mid i' \leq a \leq i' + 1 \text{ et } j' \leq b < j' + 1\} \\ &= \{p' \in \mathcal{P}^{k+1} \mid ir_x^k \leq a < ir_x^k + (m - 1) + 1 \text{ et } jr_y^k \leq b < jr_y^k + (n - 1) + 1\} \\ &= \{p' \in \mathcal{P}^{k+1} \mid ir_x^k \leq a < ir_x^k + m \text{ et } jr_y^k \leq b < jr_y^k + n\} \end{aligned}$$

□

La fenêtre de réduction de taille $m \times n$ d'un pointel p est donc un ensemble connexe de $m \times n$ pointels p' au niveau $k + 1$. Comme dans le cas des pixels (Section 1.4.1), la notion de fenêtre de réduction entre les pointels code une hiérarchie telle que :

$\forall p' \in \mathcal{FR}(p)$, p est le parent de p' et p' est un fils de p .

La fenêtre de réduction d'un pointel (*resp.* pixel) est un ensemble connexe de pointels (*resp.* pixels) au niveau suivant. Nous introduisons la notion de *projeté direct* d'un pointel (*resp.* pixel) afin de désigner le pointel (*resp.* pixel) le plus en haut à gauche de cette fenêtre dans la Définition 56 (*resp.* Définition 57).

Définition 56 (Projeté direct d'un pointel). *Soit $p \in \mathcal{P}^k$ un pointel de coordonnées (i, j) . Le projeté direct de p , noté $P_{\text{direct}}(p)$, est l'unique pointel de $\mathcal{FR}_{1,1}(p)$.*

Définition 57 (Projeté direct d'un pixel). *Soit $p \in I^k$ un pixel de coordonnées (i, j) . Le projeté direct de p , noté $P_{\text{direct}}(p)$, est l'unique pixel de $\mathcal{FR}_{1,1}(p)$.*

Dans la Proposition 9, nous montrons que l'utilisation d'une fenêtre de réduction de taille $r_x^k \times r_y^k$ construit une pyramide de pointels non trouée et non recouvrante, de la même façon que pour une hiérarchie de pixels.

Proposition 9. *Soit \mathcal{I} une pyramide d'images composée de k niveaux. La relation $\mathcal{FR}_{r_x^k, r_y^k}$ code une pyramide de pointels non trouée et non recouvrante :*

$$\forall p' \in \mathcal{P}^{k+1}, \exists ! p \in \mathcal{P}^k \mid p' \in \mathcal{FR}_{r_x^k, r_y^k}(p).$$

Démonstration. Supposons qu'il existe p' de coordonnées (a, b) appartenant à \mathcal{P}^{k+1} tel que pour tout pointel p de coordonnées (i, j) appartenant à \mathcal{P}^k , $p' \notin \mathcal{FR}_{r_x^k, r_y^k}(p)$. D'après la Proposition 8, nous avons :

$$p' \notin \mathcal{FR}_{r_x^k, r_y^k}(p) \Rightarrow \begin{cases} ir_x^k > a \text{ ou } a \geq ir_x^k + r_x^k \\ jr_y^k > b \text{ ou } b \geq jr_y^k + r_y^k \end{cases}$$

pour tout pointel $p \in \mathcal{P}^k$ donc valable pour tout couple (i, j) vérifiant $0 \leq i \leq l^k, 0 \leq j \leq h^k$. Considérons le couple $(\lfloor \frac{a}{r_x^k} \rfloor, \lfloor \frac{b}{r_y^k} \rfloor)$ ¹ :

$$p' \notin \mathcal{FR}_{x,y}(p) \Rightarrow \begin{cases} a < \lfloor \frac{a}{r_x^k} \rfloor r_x^k \text{ ou } a - r_x^k \geq \lfloor \frac{a}{r_x^k} \rfloor r_x^k \\ b < \lfloor \frac{b}{r_y^k} \rfloor r_y^k \text{ ou } b - r_y^k \geq \lfloor \frac{b}{r_y^k} \rfloor r_y^k \end{cases}$$

Or, nous avons $\frac{a}{r_x^k} - 1 < \lfloor \frac{a}{r_x^k} \rfloor \leq \frac{a}{r_x^k}$ et $\frac{b}{r_y^k} - 1 < \lfloor \frac{b}{r_y^k} \rfloor \leq \frac{b}{r_y^k}$ donc :

$$p' \notin \mathcal{FR}_{x,y}(p) \Rightarrow \begin{cases} a < \lfloor \frac{a}{r_x^k} \rfloor r_x^k < a \text{ ou } a - r_x^k \geq \lfloor \frac{a}{r_x^k} \rfloor r_x^k > a - r_x^k \\ b < \lfloor \frac{b}{r_y^k} \rfloor r_y^k < b \text{ ou } b - r_y^k \geq \lfloor \frac{b}{r_y^k} \rfloor r_y^k > b - r_y^k \end{cases}$$

Les deux configurations étant impossibles, pour tout pointel $p' \in \mathcal{P}^{k+1}$, il existe un pointel $p \in \mathcal{P}^k$ tel que $p' \in \mathcal{FR}_{r_x^k, r_y^k}(p)$ et la pyramide est non trouée.

1. La notation $\lfloor \cdot \rfloor$ désigne l'opérateur partie entière.

3.2. Définition du modèle de pyramide descendante tuilée

Soient p_1 et p_2 deux pointels de \mathcal{P}^k de coordonnées respectives (i_1, j_1) et (i_2, j_2) . Supposons $p' \in \mathcal{FR}_{r_x^k, r_y^k}(p_1)$ et $p' \in \mathcal{FR}_{r_x^k, r_y^k}(p_2)$. Alors,

- $i_1 r_x^k \leq a < i_1 r_x^k + r_x^k$ et $i_2 r_x^k \leq a < i_2 r_x^k + r_x^k$
- $j_1 r_y^k \leq b < j_1 r_y^k + r_y^k$ et $j_2 r_y^k \leq b < j_2 r_y^k + r_y^k$

Nous avons $\frac{a}{r_x^k} - 1 < i_1 \leq \frac{a}{r_x^k} \Rightarrow \lfloor \frac{a}{r_x^k} \rfloor \leq i_1 \leq \frac{a}{r_x^k} \Rightarrow i_1 = \lfloor \frac{a}{r_x^k} \rfloor$ car i_1 est entier. Le même raisonnement peut être établi pour i_2 donc $i_1 = i_2 = \lfloor \frac{a}{r_x^k} \rfloor$. De la même façon, $j_1 = j_2 = \lfloor \frac{b}{r_y^k} \rfloor$ et donc $p_1 = p_2$. Pour tout pointel de \mathcal{P}^{k+1} , il n'existe qu'un unique parent dans \mathcal{P}^k et la pyramide est non recouvrante. \square

Étudions à présent la préservation de la connexité entre deux niveaux consécutifs d'une pyramide de pointels. La Proposition 10 établit un premier résultat dans ce sens, en démontrant que l'union des fenêtres de réduction de deux pointels adjacents est connexe.

Proposition 10. *Soit \mathcal{I} une pyramide d'images composée de n niveaux. Étant donné deux pointels p et p' adjacents d'un niveau k , l'union de leurs fenêtres de réduction est un ensemble connexe de pointels du niveau $k + 1$. Pour tout couple de pointels $(p, p') \in \mathcal{P}^k \times \mathcal{P}^k$ tel que $p \in V_4(p')$, nous avons :*

$$\mathcal{FR}_{r_x^k, r_y^k}(p) \cup \mathcal{FR}_{r_x^k, r_y^k}(p') \text{ est connexe dans } \mathcal{P}^{k+1}.$$

Démonstration. Soient p et p' deux pointels de coordonnées (i, j) et (i', j') de \mathcal{P}^k et tels que $|i - i'| + |j - j'| = 1$. Chacune des fenêtres de réduction étant connexe, il suffit de trouver un couple de pointels adjacents dans $\mathcal{FR}_{r_x^k, r_y^k}(p) \times \mathcal{FR}_{r_x^k, r_y^k}(p')$. Sans perte de généralité, plaçons-nous dans la configuration $i - i' = 1$ et $j - j' = 0$. Dans ce cas, considérons les deux pointels de \mathcal{P}^{k+1} de coordonnées $(i r_x^k, j r_y^k)$ et $(i' r_x^k + r_x^k - 1, j' r_y^k)$ appartenant respectivement aux fenêtres de réduction de p et p' . Alors, $|i r_x^k - (i' r_x^k + r_x^k - 1)| + |r_y^k j - r_y^k j'| = |r_x^k(i - i') - r_x^k - 1| + r_y^k |j - j'| = 1$ et les deux pointels sont bien adjacents.

De la même façon, dans les trois autres cas, nous pouvons considérer les couples de pointels suivants :

- $(i r_x^k + r_x^k - 1, r_y^k j)$ et $(i' r_x^k, r_y^k j')$ si $i - i' = -1$ et $j - j' = 0$;
- $(i r_x^k, r_y^k j)$ et $(i' r_x^k, r_y^k j' + r_y^k - 1)$ si $i - i' = 0$ et $j - j' = 1$;
- $(i r_x^k, r_y^k j + r_y^k - 1)$ et $(i' r_x^k, r_y^k j')$ si $i - i' = 0$ et $j - j' = -1$.

Ces pointels étant adjacents dans \mathcal{P}^{k+1} , il existe toujours deux pointels $p_1 \in \mathcal{FR}_{r_x^k, r_y^k}(p)$ et $p_2 \in \mathcal{FR}_{r_x^k, r_y^k}(p')$ tels que p_1 et p_2 sont connexes dans \mathcal{P}^{k+1} . \square

Corrolaire 2. Pour tout ensemble de pointels connexe dans \mathcal{P}^k , l'union de leurs fenêtres de réduction est connexe dans \mathcal{P}^{k+1} :

$$\forall C \text{ connexe dans } \mathcal{P}^k, \bigcup_{p \in C} \mathcal{FR}_{r_x^k, r_y^k}(p) \text{ est connexe dans } \mathcal{P}^{k+1}.$$

Démonstration. Soit $n \in \mathbb{N}^*$. Soit C un ensemble connexe de pointels de \mathcal{P}^k . Pour tout couple (p, p') dans C^2 , il existe un chemin $c = \{p_1 = p \dots p_n = p'\}$ reliant p à p' dans \mathcal{P}^k . Par définition, nous avons $\mathcal{FR}_{r_x^k, r_y^k}(p_1)$ est connexe dans \mathcal{P}^{k+1} . Supposons $\bigcup_{i=1}^k \mathcal{FR}_{r_x^k, r_y^k}(p_i)$ connexe dans \mathcal{P}^{k+1} . L'union des deux fenêtres $\mathcal{FR}_{r_x^k, r_y^k}(p_k) \cup \mathcal{FR}_{r_x^k, r_y^k}(p_{k+1})$ est connexe dans \mathcal{P}^{k+1} d'après la Proposition 10 donc $\bigcup_{i=1}^{k+1} \mathcal{FR}_{r_x^k, r_y^k}(p_i)$ est connexe dans \mathcal{P}^{k+1} . Par récurrence, $\bigcup_{i=1}^n \mathcal{FR}_{r_x^k, r_y^k}(p_i)$ est donc connexe dans \mathcal{P}^{k+1} . \square

3.2.2 Pyramides de tuiles

Une pyramide d'images associe une représentation multi-résolution à une image I permettant par exemple d'effectuer certaines opérations à faible résolution dont les résultats sont projetés puis raffinés à une résolution plus importante. Cependant, dans le cas des grandes images, une unique résolution peut être trop importante pour être chargée en mémoire. De la même façon que dans le Chapitre 2, nous utilisons une décomposition de chaque niveau en un ensemble de tuiles suivant un pavage régulier rectangulaire.

L'ensemble des tuiles définies aux différents niveaux de la pyramide d'images forme une structure hiérarchique (Section 1.4.1) appelée *pyramide de tuiles*. Une pyramide de tuiles utilise un système de représentation à trois coordonnées pour l'ensemble des tuiles topologiques, notées $t(i, j, k)$, où (i, j, k) indique les coordonnées (i, j) de la tuile t au niveau k . Pour des raisons d'optimisation, nous nous intéressons uniquement au cas d'une structure non recouvrante (pas de duplication de l'information). Par ailleurs, de façon à ce que toute la hiérarchie soit représentée, nous nous plaçons également dans le cadre d'une pyramide non trouée.

En d'autres termes, nous cherchons à définir une hiérarchie régulière de pavages de façon à ce que l'ensemble des filles de chaque tuile forme une partition de celle-ci. Pour cela, considérons une tuile de largeur l et de hauteur h appartenant à un niveau k . La projection de cette tuile au niveau $k + 1$ fournit une image de taille $(r_x^k \times l) \times (r_y^k \times h)$. Cette image peut être pavée par ses $p \times q$ tuiles filles (Figure 3.3(a)) si et seulement si :

3.2. Définition du modèle de pyramide descendante tuilée

- $r_x^k \times l \pmod p = 0$;
- $r_y^k \times h \pmod q = 0$.

Si aucune contrainte particulière n'est imposée sur (l, h) , ces contraintes s'écrivent alors :

- $r_x \pmod p = 0$;
- $r_y \pmod q = 0$.

Deux solutions sont alors privilégiées :

- le cas $p = q = 1$ correspond à un nombre de tuiles constant par niveau de taille $(r_x^k \times l) \times (r_y^k \times h)$;
- le cas $p = r_x$ et $q = r_y$ correspond à une taille de tuile constante égale à $l \times h$.

Ces deux approches sont respectivement illustrées dans la Figure 3.3(b) et Figure 3.3(c). Nous avons implanté ces deux approches dans notre modèle et nous les détaillons ci-après. Notons enfin que dans le cas d'une image mono-résolution, la résolution de chaque niveau de l'image est la même et les deux approches sont strictement équivalentes.

Nombre de tuiles constant par niveau

La première approche consiste à conserver un nombre constant de tuiles par niveau. De la sorte, les tuiles ne sont pas découpées d'un niveau à l'autre ce qui induit des relations hiérarchiques (relations *prédécesseur/successeur*) uniques entre les tuiles des différents niveaux : chaque tuile d'un niveau k a exactement une tuile *prédécesseur* appartenant au niveau $k - 1$ et une tuile *successeur* appartenant au niveau $k + 1$, à l'exception des tuiles du sommet et de la base de la pyramide qui n'ont pas de *prédécesseur* et respectivement pas de *successeur*. L'inconvénient de cette méthode est que la taille d'une tuile d'un niveau k dépend de la taille d'une tuile au premier niveau et du facteur d'échelle entre les résolutions I^0 et I^k . Cette approche n'est donc pas utilisable en pratique pour des images multi-résolution présentant un ratio important entre le premier et le dernier niveau car la taille des tuiles du dernier niveau devient alors trop importante.

Taille de tuile constante

La seconde approche définit une taille constante de tuiles, quel que soit le niveau dans la pyramide : l'occupation mémoire d'une tuile dépend uniquement du choix de ses dimensions. En contre-partie, cette approche entraîne une opération de découpe récursive des tuiles qui induit des relations hiérarchiques similaires aux modèles de quadrees (Section 1.4.1) : chaque tuile d'un niveau

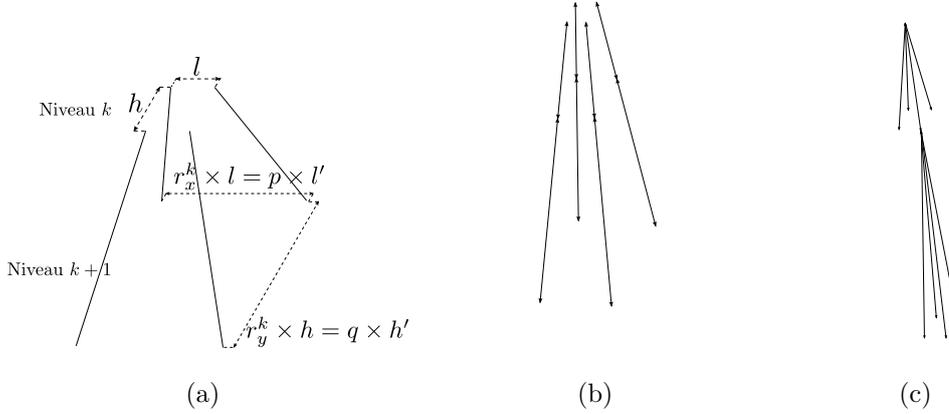


FIGURE 3.3 – Pavages hiérarchiques. Le cas $r_x \bmod p = r_y \bmod q = 0$ permet d’obtenir une pyramide régulière non trouée et non recouvrante. (a) Problématique de découpage d’une tuile parente en un ensemble de tuiles filles de taille $l' \times h'$; (b) Nombre de tuiles constant par niveau; (c) Taille des tuiles constante.

k dispose d’une unique tuile *prédécesseur* appartenant au niveau $k - 1$ et d’un ensemble de tuiles *successeur* appartenant au niveau $k + 1$.

Nous formalisons cette problématique du choix de subdivision en étendant la notion de fenêtre de réduction aux tuiles (Définition 58).

Définition 58 (Fenêtre de réduction d’une tuile). *Soit une pyramide de $n + 1$ pavages rectangulaires. La fenêtre de réduction d’une tuile t d’un niveau k ($0 \leq k < n$) correspond à l’ensemble des filles de t dans la structure hiérarchique de tuiles. Cette relation, notée $\mathcal{FR}(t)$, code la hiérarchie par :*

$$\forall t' \in \mathcal{FR}(t), t \text{ est le parent de } t' \text{ et } t' \text{ est la fille de } t.$$

Soit $\mathcal{I} = (I^0, \dots, I^n)$ une pyramide d’images, nous construisons pour chaque résolution I^k un pavage T^k constitué de subdivisions rectangulaires. Soient deux résolutions successives I^k et I^{k+1} et r_x, r_y leurs ratios. Nous proposons de définir la notion de *pyramide de tuiles* (Définition 59) à partir de la notion de fenêtre de réduction d’une tuile (Définition 58). Nous nous restreignons au cas particulier d’une pyramide régulière, non trouée et non recouvrante composée de tuiles de taille constante.

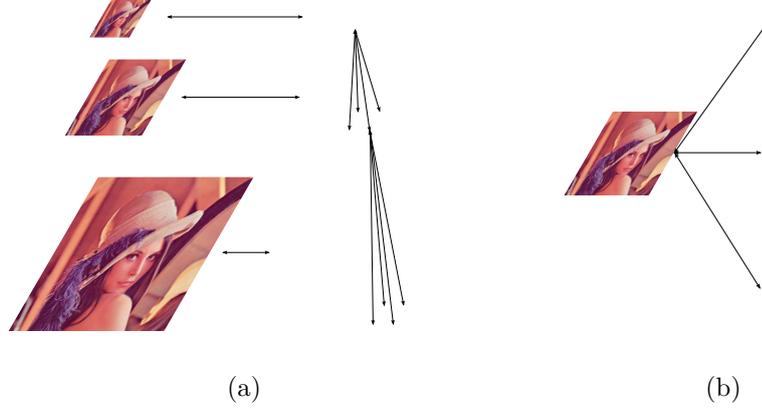


FIGURE 3.4 – Construction d’une pyramide régulière de tuiles sur une pyramide régulière d’images. (a) Pyramide de tuiles associée à une image multi-résolution (ratio constant égal à 4); (b) Pyramide de tuiles associée à une image mono-résolution.

Définition 59 (Pyramide de tuiles). *Soit I une image. Une pyramide de tuiles associée à I est un doublet $(\mathcal{I}, \mathcal{T} = (T^0, \dots, T^n))$ où $\forall k, 0 \leq k < n$:*

- $\mathcal{I} = (I^0, \dots, I^n)$ est une pyramide d’images, non trouée, et non recouvrante associée à I ;
- $T^k = \{t(i, j, k)\}_{(i,j) \in [0, l^k/l[\times [0, h^k/h[}$ est un pavage rectangulaire composé de tuiles de taille $l \times h$, associé à la résolution I^k de largeur l^k et de hauteur h^k ;
- $\mathcal{FR}(t(i, j, k)) = \{t(i', j', k + 1) \mid i' \in \{ir_x^k, \dots, ir_x^k + r_x^k\} \text{ et } j' \in \{jr_y^k, \dots, jr_y^k + r_y^k\}\}$ est la fenêtre de réduction de la tuile $t(i, j, k)$.

Ainsi, une tuile d’origine (x, y) et de dimensions (l, h) du niveau k est la tuile parente de $r_x^k \times r_y^k$ tuiles de niveau $k + 1$ et de dimensions $l \times h$, appelées tuiles filles. Cette tuile parente code la subdivision géométrique d’origine $(x \times r_x, y \times r_y)$. Deux exemples de pyramides de tuiles sont illustrés par la Figure 3.4. Dans la Figure 3.4(a), nous associons une pyramide régulière de tuiles de taille constante à une image multi-résolution présentant un ratio constant égal à 4 ($r_x^k = r_y^k = 2$). Dans ce cas, nous obtenons un partitionnement de l’image en tuiles similaire à la découpe récursive utilisée par les modèles de quadrees (Section 1.4.1). Dans la Figure 3.4(b), nous illustrons le cas dégénéré d’une image présentant un ratio constant égal à 1 : la pyramide d’images est alors mono-résolution. Le traitement d’images mono-résolution apparaît donc comme un cas particulier de notre modèle.

3.2.3 Pyramides descendantes tuilées

De manière à pouvoir traiter de grandes images multi-résolution, nous cherchons à définir un modèle hiérarchique descendant étendant le modèle de cartes tuilées pour représenter une pyramide d'images. Chaque niveau de ce modèle est une carte tuilée correspondant à un raffinement du niveau précédent. Ce raffinement est obtenu par l'application d'opérations de découpe des régions à chaque niveau ce qui induit des relations hiérarchiques entre les régions.

Considérons d'un point de vue hiérarchique l'opération de décomposition/fusion d'une région r appartenant à une carte tuilée que nous avons définie dans la Section 2.5.3. Soit un niveau k contenant r et un niveau $k + 1$ contenant l'ensemble $R = \{r_1, \dots, r_n\}$ des régions résultant du raffinement de r . L'opération de découpe induit une hiérarchie entre r et les régions de R que nous représentons par des fonctions *prédécesseur/successeur* entre les régions : pour chaque région $r_i \in R$, r est le *prédécesseur* de r_i et r_i est un *successeur* de r . Le raffinement d'une région peut entraîner une découpe des arêtes qui correspondent à ses bords. Nous représentons les relations hiérarchiques induites entre les arêtes par des fonctions *prédécesseur/successeur* entre les brins correspondant. De la sorte, chaque région (*resp.* brin) de G^k dispose d'au moins un descendant dans G^{k+1} et chaque région (*resp.* brin) de G^k a *au plus* un antécédent dans G^{k-1} .

Les travaux sur les modèles multi-échelles réalisés par Koenderink [Koe84] permettent de dégager le *principe de causalité* selon lequel toute structure observée à une échelle grossière doit trouver une cause à une échelle plus fine. Dans le cas d'une pyramide descendante, ce principe se traduit par la contrainte que chaque frontière d'un niveau k doit exister à tout niveau $l > k$ [GCL06]. Cette contrainte garantit une cohérence du modèle pyramidal en assurant qu'il n'y a pas de perte d'information lors d'un changement de niveau. L'opération de découpe d'une région doit donc conserver les bords. L'étape de fusion est ainsi restreinte aux régions issues de la décomposition de r en régions élémentaires. Si nous considérons deux cartes tuilées G^k et G^{k+1} , où G^{k+1} s'obtient par raffinement des régions de G^k , la préservation des bords des régions de G^k dans G^{k+1} définit une *structure causale*. Nous nous appuyons sur ce principe de causalité pour définir le modèle de pyramide descendante tuilée (Définition 60). Les structures régulières apportent quant à elles une description multi-résolution de l'image et une borne sur la quantité de mémoire requise pour construire chaque carte, garantissant ainsi la faisabilité de la construction de la pyramide.

3.2. Définition du modèle de pyramide descendante tuilée

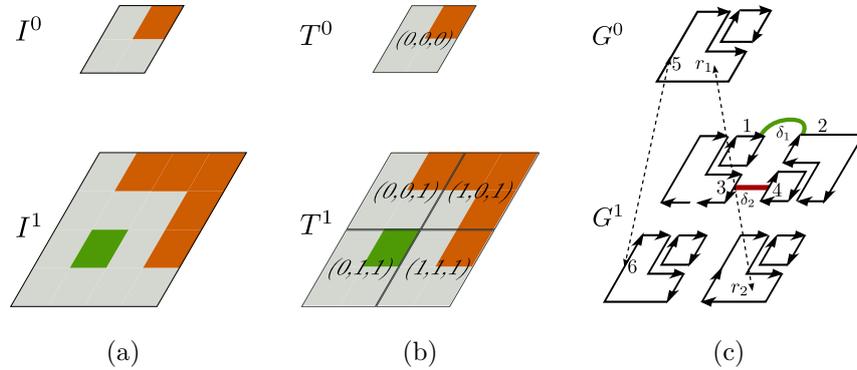


FIGURE 3.5 – Représentation d’une pyramide descendante tuilée composée de deux niveaux. (a) Pyramide régulière d’images ($r_x = r_y = 2$); (b) Pyramide régulière de tuiles de taille constante ($\text{largeur} = \text{hauteur} = 2\text{px}$); (c) Pyramide irrégulière de cartes tuilées : chaque niveau est composé d’une carte tuilée.

Définition 60 (Pyramide descendante tuilée). Soit I une image. Une pyramide descendante tuilée est un triplet $(\mathcal{I}, \mathcal{T}, \mathcal{G} = (G^0, \dots, G^n))$ où $\forall k, 0 \leq k < n$:

- $(\mathcal{I}, \mathcal{T})$ est une pyramide de tuiles de $n + 1$ niveaux associée à I (Définition 59);
- G^k est la carte tuilée s’appuyant sur le pavage de I^k par T^k ;
- G^{k+1} est définie sur (I^{k+1}, T^{k+1}) et se déduit de G^k par une projection de ses frontières et des opérations de raffinement de ses régions.

Notre définition d’un modèle de pyramide descendante tuilée est donc une combinaison des trois modèles hiérarchiques illustrés dans la Figure 3.5.

En premier lieu, nous disposons d’une pyramide d’images régulière (Figure 3.5(a)) représentant différentes résolutions de l’image, permettant ainsi la représentation d’images multi-résolution.

Ensuite, chaque niveau de la pyramide d’images est décomposé en tuiles suivant un pavage régulier, définissant une pyramide de tuiles et donc une seconde structure régulière (Figure 3.5(b)). Notons qu’une pyramide de tuiles est une structure similaire à un quadtree lorsque le ratio entre deux niveaux consécutifs est égal à 4.

Enfin, nous construisons une pyramide irrégulière en associant une carte tuilée (Définition 53, page 82) à chaque pavage de chaque niveau de la pyramide de tuiles. Le modèle de carte tuilée utilisé pour coder chacun des niveaux de la hiérarchie établit une pyramide irrégulière (Figure 3.5(c)) pour la représentation de la hiérarchie de partitions associée à l’image multi-résolution. Par conséquent, une pyramide de cartes tuilées dispose des

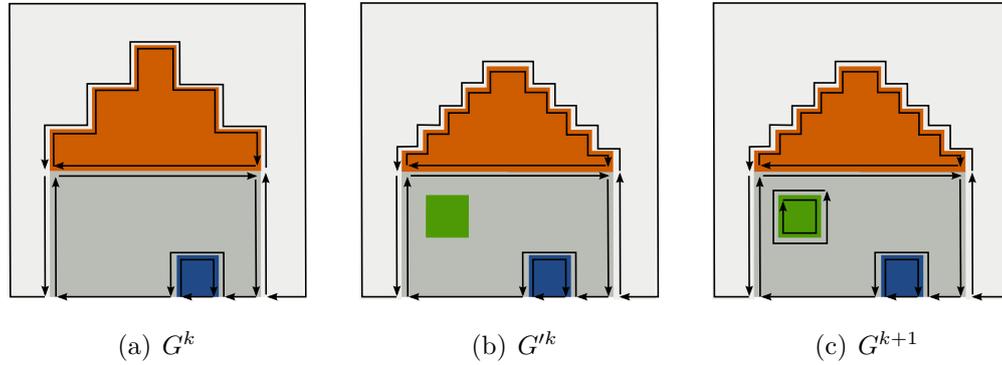


FIGURE 3.6 – Schéma de construction d’une pyramide descendante tuilée. (a) Partition d’un niveau k ; (b) Construction de G'^k par projection de G^k au niveau $k+1$; (c) Construction de G^{k+1} par raffinement.

propriétés inhérentes aux structures irrégulières comme la connexité des régions sur un même niveau ou la représentation des régions allongées (Section 1.4.1).

Notons enfin que la pyramide détermine les tuiles devant être chargées en mémoire ou enregistrées sur disque dur et assure la propagation des modifications d’une tuile aux tuiles adjacentes. Cela permet de garantir la cohérence globale du modèle, d’une manière similaire à ce que nous avons présenté au chapitre précédent. En effet, si nous effectuons une opération de découpe sur le bord d’une tuile, la tuile adjacente doit être modifiée en conséquence, qu’elle soit en mémoire ou sur disque.

3.3 Représentation des relations hiérarchiques

La définition du modèle de pyramide descendante tuilée conduit à la définition de relations *prédécesseur/successeur* entre les brins et les régions. Dans cette section, nous nous intéressons à la manière dont nous déterminons le successeur d’un élément dans un schéma de construction descendante. Ce schéma s’appuie sur la construction d’une carte intermédiaire G'^k entre les cartes G^k et G^{k+1} . La carte G'^k correspond à la projection des éléments de G^k dans (I^{k+1}, T^{k+1}) . Sa construction constitue l’étape préliminaire au raffinement des régions conduisant à la construction de G^{k+1} (Figure 3.6) et permet de garantir la causalité de la pyramide. Notons que les cartes combinatoires de G^k et de sa projection G'^k sont isomorphes [Cor75, DdJ⁺09]².

2. Deux cartes $(\mathcal{B}, \delta_1, \delta_2)$ et $(\mathcal{B}', \delta'_1, \delta'_2)$ sont isomorphes s’il existe une bijection $f : \mathcal{B} \rightarrow \mathcal{B}'$ telle que $\forall b \in \mathcal{B}, f(\delta_1(b)) = \delta'_1(f(b))$ et $f(\delta_2(b)) = \delta'_2(f(b))$.

Dans la Section 3.3.1, nous détaillons la manière dont nous établissons le projeté des frontières des régions. La frontière d'une région correspondant à un $(0, 1)$ -chemin (Section 1.2.3), nous introduisons la notion de projection d'un $(0, 1)$ -chemin de G^k pour définir G'^k . Dans la Section 3.3.2, nous expliquons comment sont construites les relations *prédécesseur/successeur* au niveau des brins et des régions entre G^k et G'^k .

3.3.1 Projection des $(0,1)$ -chemins

La définition de relations *prédécesseur/successeur* entre éléments interpixel est étroitement liée à la notion de projection des frontières des régions entre deux niveaux consécutifs d'une pyramide descendante tuilée. Considérons la problématique de la projection de la matrice des éléments interpixel d'un niveau k au niveau $k + 1$. De façon à respecter la propriété de causalité, chaque $(0, 1)$ -chemin existant au niveau k doit exister au niveau $k + 1$.

Nous représentons implicitement les relations *prédécesseur/successeur* des pointels et des pixels en nous appuyant sur la notion de fenêtre de réduction de la pyramide d'images régulière. Nous ne définissons pas directement de relation *prédécesseur/successeur* au niveau des lignels frontières dans la mesure où nous autorisons des modifications du tracé des $(0, 1)$ -chemins lorsque la topologie des niveaux k et $k + 1$ reste la même. À partir de la notion de fenêtre de réduction des pointels (voir Définition 55), nous définissons la *bande de projection* d'un $(0, 1)$ -chemin dans la Définition 61. Intuitivement, la bande de projection d'un $(0, 1)$ -chemin correspond à l'épaisseur de ce chemin au niveau suivant. Nous désignons par \mathcal{P}^k (*resp.* \mathcal{L}^k) l'ensemble des pointels (*resp.* lignels) du niveau k .

Définition 61 (Bande de projection). *Soit $c = (p_0, l_0, \dots, p_n)$ un $(0, 1)$ -chemin de niveau k . La bande de projection de c est un sous-complexe $\mathcal{BP}(c) = P \cup L$ du niveau $k + 1$ avec :*

- P est un ensemble de pointels tel que $P = \bigcup_{i=0}^n \mathcal{FR}_{r_x^k, r_y^k}(p_i)$;
- L est un ensemble de lignels tel que $L = \{l \in \mathcal{L}^{k+1} \mid l \text{ est incident à deux pointels distincts de } P\}$.

La Proposition 11 nous assure qu'il est toujours possible d'établir un $(0, 1)$ -chemin entre deux pointels de la bande de projection.

Proposition 11. *La bande de projection d'un $(0, 1)$ -chemin est un sous-complexe connexe de dimension 1.*

Démonstration. Soit c un $(0, 1)$ -chemin de G^k et $\mathcal{BP}(c) = P \cup L$ sa bande de projection. Par définition, l'ensemble des pointels d'un $(0, 1)$ -chemin est

un ensemble connexe de pointels. De plus, d'après la Proposition 9, le fait d'utiliser une fenêtre de réduction de taille $r_x^k \times r_y^k$ nous place dans le cadre d'une pyramide de pointels non trouée et non recouvrante. D'après le Corollaire 2, l'union des fenêtres de réduction des pointels d'un $(0, 1)$ -chemin d'un niveau k forme un ensemble de pointels connexe dans \mathcal{P}^{k+1} donc P est connexe dans \mathcal{P}^{k+1} .

Soient $(p, p') \in P^2$. Il existe donc un chemin $c = (p_0 = p, \dots, p_n = p')$ tel que $\forall i \in \{0, \dots, n\}$, $p_{i+1} \in V_4(p_i)$. Par définition de L , le lignel l incident à p_i et à p_{i+1} appartient à L . Par conséquent, le $(0, 1)$ -chemin $(p_0, l_0, p_1, \dots, p_n)$ est connexe et inclus dans $\mathcal{BP}(c)$.

Soient $(l, l') \in L^2$ et p (resp. p') un pointel de P incident à l (resp. l'). Il existe un $(0, 1)$ -chemin $(p_0 = p \dots p_n = p')$ inclus dans $\mathcal{BP}(c)$. Par conséquent, l'ensemble $\{l, p_0, l_0, \dots, p_n, l'\}$ est connexe et inclus dans $\mathcal{BP}(c)$. Nous pouvons donc relier l et l' par un chemin de $\mathcal{BP}(c)$.

De la même manière, nous pouvons montrer que tout couple de pointel, lignel peut être relié par un chemin dans $\mathcal{BP}(c)$. Ce dernier sous-complexe est donc connexe. \square

Nous définissons le projeté d'un $(0, 1)$ -chemin c comme tout $(0, 1)$ -chemin c' contenu dans la bande de projection de c (Définition 62). Les extrémités du chemin projeté doivent correspondre aux extrémités du chemin parent.

Définition 62 (Projeté d'un $(0, 1)$ -chemin). *Soit $c = (p_0, l_0, \dots, p_n)$ un $(0, 1)$ -chemin d'un niveau k . Son projeté dans G^k est un $(0, 1)$ -chemin $c' = (p'_0, l'_0, \dots, p'_m)$ tel que :*

- $p'_0 \in \mathcal{FR}_{r_x^k, r_y^k}(p_0)$;
- $p'_m \in \mathcal{FR}_{r_x^k, r_y^k}(p_n)$;
- $\{p'_0, \dots, p'_m\} \cup \{l'_0, \dots, l'_{m-1}\} \subset \mathcal{BP}(c)$.

La Figure 3.7(a) illustre la bande de projection d'un $(0, 1)$ -chemin composé de cinq pointels au niveau précédent. Dans la Figure 3.7(b), nous avons arbitrairement choisi un projeté de ce même $(0, 1)$ -chemin. Notons que les pointels extrémités 1 et 2 du $(0, 1)$ -chemin et les extrémités 3 et 4 de son projeté vérifient bien : $3 \in \mathcal{FR}_{r_x^k, r_y^k}(1)$ et $4 \in \mathcal{FR}_{r_x^k, r_y^k}(2)$.

Considérons maintenant la problématique de la projection de plusieurs $(0, 1)$ -chemins d'un même niveau k . Puisque la pyramide n'est pas recouvrante, l'intersection des bandes de projections est vide si les chemins n'ont pas de pointels en commun (Proposition 12). La définition d'un $(0, 1)$ -chemin garantit que cette condition est toujours vérifiée, excepté au niveau des extrémités.

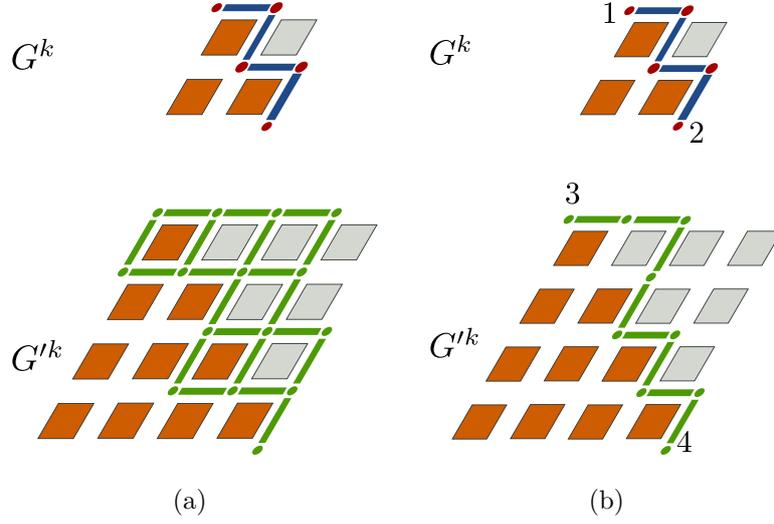


FIGURE 3.7 – Projection d'un $(0, 1)$ -chemin entre G^k et G'^k . (a) La bande de projection associée au chemin. (b) Sélection arbitraire d'un projeté contenu dans la bande de projection.

Proposition 12. Soient $c = (p_0, l_0, \dots, p_n)$ et $c' = (p'_0, l'_0, \dots, p'_m)$ deux $(0, 1)$ -chemins d'un niveau k :

$$\forall i, 0 \leq i \leq n, \forall j, 0 \leq j \leq m, p_i \neq p'_j, \mathcal{BP}(c) \cap \mathcal{BP}(c') = \emptyset.$$

Démonstration. Soient $c = (p_0, l_0, \dots, p_n)$ et $c' = (p'_0, l'_0, \dots, p'_m)$ deux $(0, 1)$ -chemins d'un niveau k tels que $\forall i, 0 \leq i \leq n, \forall j, 0 \leq j \leq m, p_i \neq p'_j$. Supposons $\mathcal{BP}(c) \cap \mathcal{BP}(c') \neq \emptyset$. Il existe donc une k -cellule e de dimension 0 ou 1 commune aux deux bandes de projection.

1. Cas $\dim(e) = 0$. La k -cellule e est un pointel appartenant à la fois à la fenêtre de réduction d'un pointel de c et de c' . D'après la Proposition 9, il existe donc un pointel commun à c et c' ce qui est impossible.
2. Cas $\dim(e) = 1$. Par définition d'une bande de projection (Définition 61), la k -cellule e est un lignel incident à la fois à deux pointels de $\mathcal{BP}(c) \cap \mathcal{P}^{k+1}$ et à deux pointels de $\mathcal{BP}(c') \cap \mathcal{P}^{k+1}$. Par conséquent, il existe un pointel commun entre les bandes de projection de c et de c' . Comme dans le cas précédent, il existe donc un pointel commun à c et c' , ce qui est contradictoire avec l'hypothèse initiale.

Par conséquent, l'intersection des bandes de projection de deux chemins distincts est vide. \square

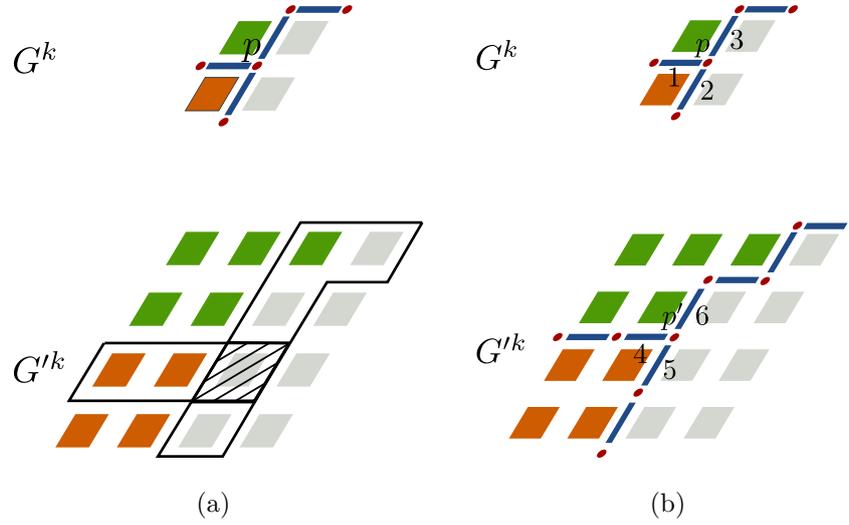


FIGURE 3.8 – Projection d'un ensemble de $(0,1)$ -chemins frontières. (a) Superposition des bandes de projection des trois $(0,1)$ -chemins au niveau de leur intersection p . (b) Ajout d'une contrainte sur p' pour respecter la topologie de G^k dans G'^k .

La Proposition 12 garantit que quels que soient les projetés d'un ensemble de chemins, les intersections (ou chevauchements) ne peuvent se produire qu'au niveau de leurs extrémités. Nous illustrons cet effet de recouvrement dans la Figure 3.8(a). Dans cet exemple, nous considérons trois $(0,1)$ -chemins d'un niveau k ayant une extrémité commune p . L'intersection des bandes de projection des trois chemins au niveau $k+1$ correspond à la fenêtre de réduction de p .

Dans le cas de la projection des frontières d'une partition, une déconnexion ou un effet de recouvrement peut donc se produire dans cette fenêtre commune. Afin de prévenir ce risque, nous définissons une méthode de projection particulière des lignels frontières incidents aux nœuds (voir Définition 26).

Considérons un pointel p dans G^k correspondant à une intersection entre plusieurs $(0,1)$ -chemins ainsi que son projeté p' dans G'^k . Pour préserver la topologie de G^k dans G'^k , nous imposons que p' désigne la même extrémité pour l'ensemble des chemins projetés. Nous choisissons donc arbitrairement le projeté direct de p (Définition 56). De plus, nous imposons la direction des lignels frontières incidents à p' d'après la direction des lignels frontières parents incidents à p . De cette façon, p' est l'unique point d'intersection à l'intérieur de la fenêtre de réduction de p et $\text{degré}(p) = \text{degré}(p')$.

Dans l'exemple proposé dans la Figure 3.8(b), le pointel p marque l'intersection de trois $(0, 1)$ -chemins du niveau k . Les trois chemins projetés partagent la même extrémité p' de \mathcal{P}^{k+1} qui correspond au projeté direct de p . Par ailleurs, la direction des lignels 4, 5 et 6 incidents à p' est imposée et correspond respectivement à la direction des lignels 1, 2 et 3 par rapport à p .

Au cours de cette section, nous avons défini les relations hiérarchiques entre les éléments interpixel d'une pyramide d'images à partir de la projection des $(0, 1)$ -chemins. De cette manière, puisqu'une pyramide descendante tuilée s'appuie sur la représentation interpixel pour définir les frontières des partitions, nous avons défini les contraintes nécessaires à la projection du plongement des brins d'une carte tuilée.

3.3.2 Relations *prédécesseur/successeur* dans une carte tuilée

Une pyramide descendante tuilée définit une hiérarchie entre un ensemble de cartes tuilées à travers des relations *prédécesseur/successeur* applicables aux tuiles topologiques, aux brins et aux régions. Pour des raisons d'optimisation, nous n'associons pas explicitement à chaque entité l'ensemble de ses successeurs mais un unique représentant que nous appelons *projeté*. Autrement dit, étant donné un brin b (*resp.* région r , tuile t) d'un niveau k , nous ne représentons qu'un unique représentant des successeurs de b (*resp.* r , t) au niveau $k + 1$, noté *successeur*(b) (*resp.* *successeur*(r), *successeur*(t)).

En effet, puisque l'ensemble des successeurs de b (*resp.* r , t) est connexe, nous pouvons retrouver tous les successeurs de b (*resp.* r , t) en identifiant parmi les voisins b_i de *successeur*(b) (*resp.* les régions voisines r_i de *successeur*(r), les tuiles adjacentes t_i de *successeur*(t)) ceux qui vérifient *prédécesseur*(b_i) = b (*resp.* *prédécesseur*(r_i) = r , *prédécesseur*(t_i) = t).

Notons que la dénomination identique des fonctions *prédécesseur* et *successeur* pour les brins, les régions et les tuiles ne pose pas de problème d'ambiguïté du fait de leurs domaines de définition différents.

Entre les brins

La projection de toutes les frontières de G^k s'effectue à topologie équivalente (Section 3.3.1) : les cartes combinatoires de G^k et de sa projection G'^k sont donc isomorphes. À chaque brin b de G^k correspond un unique brin b' de G'^k . Lors du raffinement des régions de G'^k qui conduit à la création de G^{k+1} , ce brin b' peut être décomposé en une séquence de brins (b'_1, \dots, b'_p) appartenant à G^{k+1} . Cette décomposition provient des opérations d'insertion de sommet sur la frontière

correspondant au plongement du brin b' . Par conséquent, la séquence de brins résultante est de la forme suivante :

$$\forall i, 1 \leq i < p, \exists k_i \mid b'_{i+1} = (\delta_1 \circ \delta_2)^{k_i}(b'_i).$$

Autrement dit, tout brin de la séquence appartient à l'orbite sommet du brin suivant. Nous appelons *successeur* d'un brin b le premier brin de cette séquence (Définition 63).

Définition 63 (Successeur d'un brin). *Soit b un brin de G^k et (b'_1, \dots, b'_p) sa séquence de brins correspondante dans G^{k+1} . Le successeur de b au niveau $k + 1$, noté $\text{successeur}(b)$, est le brin b'_1 .*

La fonction inverse, appelée *prédécesseur*, permet de retrouver le brin parent (si $\text{successeur}(b) = b'$ alors $\text{prédécesseur}(b') = b$). Par exemple, dans la Figure 3.9(a), $\text{successeur}(2) = 12$ et $\text{prédécesseur}(12) = 2$. Nous pouvons remarquer que la méthode que nous avons proposée pour la projection des $(0, 1)$ -chemins frontières entraîne que $\text{pointel}(\text{successeur}(b)) = P_{\text{direct}}(\text{pointel}(b))$, où $P_{\text{direct}}(\text{pointel}(b))$ désigne le projeté direct de p (Définition 56).

Conservation des propriétés causales

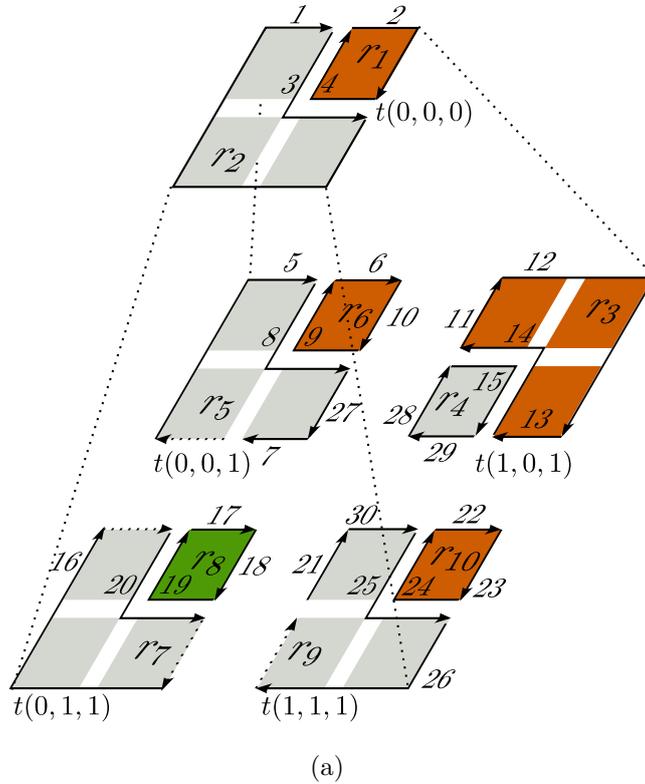
Afin de préserver la causalité de notre modèle de pyramide descendante tuilée, les relations hiérarchiques définies entre les brins et les régions respectent les deux points suivants :

1. Tous les brins (*resp.* régions) de G^{k+1} résultant de la décomposition d'un même brin (*resp.* région) de G^k ont le même prédécesseur. De plus, lors de la décomposition d'un brin successeur, nous nous assurons que la relation successeur désigne toujours le premier brin de la séquence des brins fils.
2. Tous les successeurs d'un brin b associé à un bord réel sont associés à un bord réel. Par exemple, dans la Figure 3.9(a), les brins 10, 22, 27 et 30 ont le même prédécesseur 3 qui est incident à un bord réel. Par conséquent, ils sont considérés comme incidents à un bord réel dans G^{k+1} .

Entre les régions

Nous utilisons le *premier pixel* d'une région (Section 2.4.1) afin d'établir les relations *prédécesseur/successeur*. Ainsi, étant donnée une région r appartenant à G^k , son successeur est la région contenant le projeté direct du *premier pixel* (voir Définition 57) de la région parente (Définition 64).

3.3. Représentation des relations hiérarchiques



(a)

brin	δ_1	δ_2	pred.	succ.	brin	δ_1	δ_2	pred.	succ.
1	3	∞	-	5	16	5	∞	1	-
2	4	∞	-	6	17	18	7	-	-
3	1	4	-	10	18	19	21	-	-
4	2	3	-	13	19	17	20	-	-
5	8	∞	1	-	20	21	19	-	-
6	10	∞	1	-	21	30	18	-	-
7	20	17	-	-	22	23	13	3	-
8	27	9	-	-	23	24	∞	1	-
9	6	8	-	-	24	22	25	-	-
10	9	11	3	-	25	26	24	-	-
11	12	10	4	-	26	16	∞	1	-
12	13	∞	2	-	27	7	28	3	-
13	14	22	4	-	28	15	27	4	-
14	11	15	-	-	29	28	30	4	-
15	29	14	-	-	30	25	29	3	-

région	pred.	succ.
1	-	3
2	-	5
3	1	-
4	1	-
5	2	-
6	2	-
7	2	-
8	2	-
9	2	-
10	2	-

(b)

FIGURE 3.9 – Relations *prédécesseur/successeur* dans une pyramide descendante tuilée. (a) Pyramide composée de deux niveaux avec un ratio $r_x = r_y = 2$ et une taille de tuile constante *largeur* = *hauteur* = 2px. Les brins en pointillés sont incidents à un bord fictif; (b) Relations *prédécesseur/successeur* entre les brins et les régions.

Définition 64 (Successeur d'une région). *Soit r une région de G^k . La région r' est le successeur de r si et seulement si $P_{direct}(premier\ pixel(r)) \in r'$.*

Comme dans le cas des brins, nous introduisons la fonction inverse *prédécesseur* permettant de retrouver une région parente (si $successeur(r) = r'$ alors $prédécesseur(r') = r$). Par exemple, dans la Figure 3.9(a), $successeur(r_2) = r_5$ et $prédécesseur(r_4) = r_1$.

Enfin, les structures de données que nous utilisons pour les cartes tuilées (Section 2.6) doivent être complétées pour prendre en compte l'introduction des relations hiérarchiques. Ainsi, nous rajoutons deux champs dans les structures `structure_brin` et `structure_région` désignant l'identifiant du brin *prédécesseur* et du brin *successeur* dans la carte tuilée du niveau précédent et du niveau suivant. Nous rajoutons également une coordonnée dans `structure_tuile` pour indiquer la profondeur d'une tuile dans la pyramide. Cette information permet de coder implicitement les relations *prédécesseur/successeur* entre les tuiles. En effet, le successeur d'une tuile de niveau k et d'origine (i, j) est la tuile de niveau $k + 1$ d'origine (ir_x^k, jr_y^k) .

3.4 Utilisation pratique des relations hiérarchiques

Les relations hiérarchiques d'une pyramide descendante tuilée sont représentées à travers un ensemble de relations *prédécesseur/successeur* entre les cartes tuilées qui composent les niveaux. Nous nous intéressons ici à la manière d'utiliser ces relations pour retrouver l'ensemble des éléments provenant de la découpe d'un élément à un niveau donné. En effet, la relation *successeur* donne accès à un seul élément du niveau suivant que nous appelons *projeté*. Étant donné un niveau k , le projeté permet de retrouver l'ensemble connexe du niveau $k + 1$ qui définit un objet (par exemple une arête ou une région) du niveau k . L'objectif consiste à identifier parmi les éléments voisins d'un projeté ceux qui correspondent au même objet parent. Ce type de relation est nécessaire afin d'analyser une image de manière hiérarchique.

Par exemple, considérons une pyramide descendante tuilée construite à partir d'une image histologique et une région r . Supposons que r corresponde à un foyer tumoral identifié au niveau k . Nous recherchons alors l'ensemble des régions r' du niveau $k + 1$ dont le parent est r et qui peuvent correspondre à des mitoses. De manière réciproque, étant donnée une mitose, nous recherchons son foyer tumoral défini au niveau précédent afin de caractériser la zone concernée de manière globale.

Pour répondre à ces besoins, nous proposons ici deux procédures pour retrouver les arêtes et les régions de G^{k+1} qui composent l'arête ou la région de G^k . Ces deux procédures font respectivement l'objet de la Section 3.4.1 et de la Section 3.4.2.

3.4.1 Identification des arêtes filles

Étant donné que notre modèle de pyramide descendante tuilée est une structure causale, toute frontière définie à un niveau k existe au niveau $k + 1$. Cependant, en raison des opérations de raffinement, ces frontières sont potentiellement décomposées en plusieurs frontières connexes. Nous proposons avec l'Algorithme 12 une méthode pour identifier dans G^{k+1} les arêtes filles d'une arête de G^k . La problématique consiste à retrouver l'ensemble des brins du niveau $k + 1$ codant un même brin du niveau k . Pour cela, nous utilisons les relations *prédécesseur/successeur* pour sélectionner parmi les voisins du projeté d'un brin ceux qui ont le même brin parent.

Soit b un brin de G^k dont nous cherchons les fils. Nous recherchons donc la séquence (b'_1, \dots, b'_m) de brins de G^{k+1} . Nous procédons en trois étapes :

1. Sélection du premier fils de b . Notre définition de la relation *successeur* entre les brins (Section 3.3.2) nous assure que *successeur*(b) est le premier brin de la séquence cherchée. Nous avons donc $b'_1 = \text{successeur}(b)$.
2. Détermination du brin fils suivant. Le brin b'_{i+1} appartient à l'orbite sommet de $\delta_1(b'_i)$. De plus, b'_i et b'_{i+1} ont le même parent b . De la sorte :

$$b'_{i+1} = \delta_1((\delta_1 \circ \delta_2)^k)(b'_i),$$
 avec $k = \min\{p \in \{0, \dots, 3\} \mid \text{prédécesseur}((\delta_1 \circ \delta_2)^p(\delta_1(b'_i))) = b\}$.
3. Itération de l'étape 2 jusqu'au critère d'arrêt $\delta_2(b'_{i+1}) = b'_i$ (il n'y a aucun brin suivant ayant le même parent) ou $b'_{i+1} = b'_1$ (cas des boucles).

Nous retrouvons ces étapes dans l'Algorithme 12. La **ligne 1** correspond à la sélection du premier fils. À la **ligne 3**, nous recherchons le brin suivant de l'orbite sommet ayant le même successeur. La **ligne 2** correspond à l'itération de l'étape 2. La **ligne 4** et la **ligne 5** correspondent aux deux critères d'arrêt évoqués.

Notons que dans une carte tuilée, une frontière est une séquence d'arêtes séparées par des sommets de degré 2 (Section 2.3.3). Par exemple, les deux arêtes (1,2) et (3,4) de la Figure 3.10(a) représentent une même frontière. Dans ce cas, nous appliquons l'Algorithme 12 sur la séquence de brins parents codant la frontière de G^k .

Nous présentons dans la Figure 3.10 deux cas pratiques mettant en avant les critères d'arrêt évoqués lors de l'identification des arêtes filles.

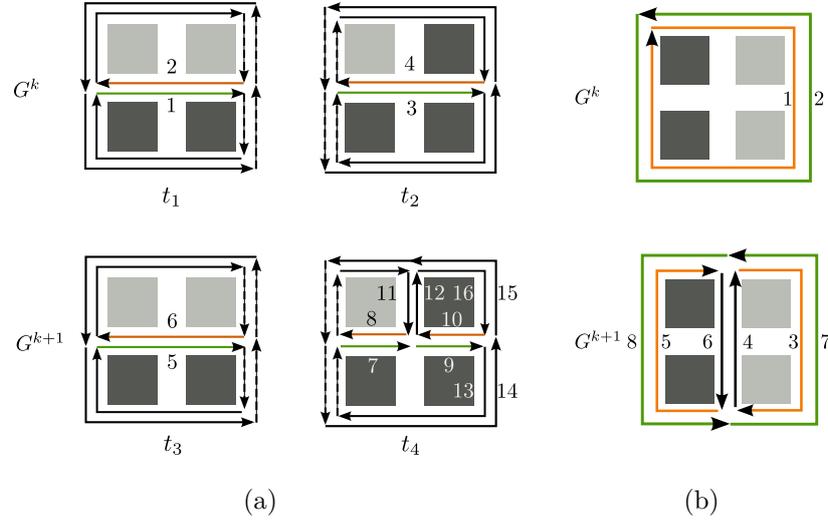


FIGURE 3.10 – Deux exemples de configurations lors du processus d'identification des arêtes filles. L'ensemble des arêtes filles de G^{k+1} issues d'un même parent code une même frontière dans G^k . (a) La frontière composée des deux arêtes (1,2) et (3,4) du niveau k se décompose en trois arêtes (5,6), (7,8) et (9,10) au niveau $k+1$; (b) La boucle (1,2) du niveau k se décompose en deux arêtes (3,7) et (5,8) au niveau $k+1$.

Dans la Figure 3.10(a), considérons l'identification des fils du brin 3. Le brin 7 est le premier fils du brin 3 ($\text{successeur}(3) = 7$). L'orbite sommet de $\delta_1(7)$ est composée des trois brins $\{9, 12, 8\}$ et seul le brin 9 a pour parent le brin 3 donc le brin 9 est un fils du brin 3. Parmi les trois brins $\{13, 15, 10\}$ de l'orbite sommet de $\delta_1(9)$, aucun n'a le brin 3 pour parent (ligne 4 de l'Algorithme 12). Par conséquent, le brin 3 de G^k est décomposé en deux brins fils 7 et 9 dans G^{k+1} . En étendant ce processus aux brins 1 et 3 des deux arêtes (1, $\delta_2(1) = 2$) et (3, $\delta_2(3) = 4$) du niveau k , nous obtenons la séquence d'arêtes (5, 6), (7, 8), (9, 10) du niveau $k+1$.

La Figure 3.10(b) illustre la configuration où un brin parent correspond à une boucle au niveau k . Considérons l'identification des fils du brin 1. Le brin 3 est le premier fils de 1 ($\text{successeur}(1) = 3$). Le brin 5 est le brin de l'orbite sommet de $\delta_1(3)$ ayant 1 pour parent donc 5 est un fils de 1. En revanche, bien que le brin 3 soit un brin du sommet suivant $\{6, 3, 8\}$ dont le parent est 1, comme $3 = \text{successeur}(1)$, nous sommes dans le cas d'une boucle (ligne 5 de l'Algorithme 12). Par conséquent, l'arête (1, $\beta_2(1) = 2$) du niveau k correspond aux deux arêtes (3, 7) et (5, 8) du niveau $k+1$.

Algorithme 12 : Identification des arêtes filles

Données : Un brin $b \in G^k$.
Résultat : La séquence de brins de G^{k+1} correspondant à b .
 $s \leftarrow$ liste vide de brins;
 $fin \leftarrow$ faux;
1 $cur \leftarrow$ successeur(b);
ajouter cur au début de s ;
2 **tant que** fin est faux **faire**
 $pre \leftarrow cur$;
 $cur \leftarrow \delta_1(cur)$;
3 **tant que** fin est faux et que prédécesseur(cur) $\neq b$ **faire**
4 $cur \leftarrow \delta_1(\delta_2(cur))$;
 si $\delta_2(cur) = pre$ **alors**
 $fin \leftarrow vrai$;
5 **si** $cur = successeur(b)$ **alors**
 $fin \leftarrow vrai$;
 si fin est faux **alors**
 ajouter cur à la fin de s ;
retourner s ;

En utilisant l'Algorithme 12 sur la séquence de brins codant une même frontière (Section 2.3.3), nous sommes donc en mesure de retrouver l'ensemble des arêtes d'un niveau donné qui correspondent à une même frontière au niveau précédent.

3.4.2 Identification des régions filles

Pour identifier les régions filles d'un niveau $k + 1$ composant une région donnée r d'un niveau k , nous parcourons les régions voisines de la région projetée et nous utilisons les relations *prédécesseur/successeur* pour déterminer celles qu'il faut considérer.

À cette fin, nous proposons une procédure, détaillée par l'Algorithme 13, pouvant se synthétiser en quatre étapes principales. Dans un premier temps, comme pour l'identification des arêtes filles, nous utilisons *successeur*(r) comme premier représentant de r au niveau $k + 1$ (ligne 1). Dans un second temps, nous parcourons la composante connexe des brins appartenant aux régions filles (ligne 2). La troisième étape consiste à identifier les régions r_i de cette composante connexe dont le *prédécesseur* est la région r (ligne 3). Enfin, la dernière étape fait appel à la procédure `identifier_régions_imbriquées`,

détaillée ci-après, pour tenir compte des régions imbriquées dans les régions filles identifiées (ligne 4).

Algorithme 13 : Identification des régions filles

Données : Une région r^k de G^k .

Résultat : L'ensemble res des régions de G^{k+1} correspondant à r^k .

```

1  $b_0 \leftarrow rep(successeur(r^k));$ 
2 pour chaque  $b \in \langle \delta_1, \delta_2 \rangle(b_0)$  tel que  $prédécesseur(région(b)) = r^k$  faire
   |    $r \leftarrow région(b);$ 
3   |   si  $r$  n'est pas marquée et  $prédécesseur(r) = r^k$  alors
   |   |   ajouter  $r$  à la fin de  $res$ ;
   |   |   marquer  $r$ ;
4   |   |    $res \leftarrow res \cup identifier\_régions\_imbriquées(r, r^k);$ 
   |   retourner  $res$ ;
```

Cas des régions imbriquées

L'opération d'identification des régions filles (Algorithme 13) fait appel à la procédure `identifier_régions_imbriquées` en ligne 4 pour tenir compte des régions imbriquées dans les régions filles identifiées. Nous proposons une méthode pour traiter ces cas d'imbrication avec l'Algorithme 14. Soit r^k une région de G^k et r une région de G^{k+1} vérifiant $prédécesseur(r) = r^k$. Nous nous appuyons sur l'arbre d'imbrication des régions de la carte pour obtenir les régions r_i imbriquées dans r (ligne 1) et dont le prédécesseur est r^k (ligne 2). Le processus est ensuite itéré récursivement pour les régions r_i identifiées (ligne 3). En effet, puisque nous utilisons la relation *fil*s de l'arbre d'imbrication, nous obtenons uniquement les régions imbriquées du premier niveau de l'arbre d'imbrication des régions de r (régions vérifiant $père(r_i) = r$). Il est donc nécessaire d'appliquer récursivement l'algorithme pour obtenir les niveaux suivants d'imbrication. En pratique, cette opération n'est pas très coûteuse car la profondeur de l'arbre d'imbrication ne dépasse que très rarement cinq niveaux.

Dans la Figure 3.11(a), considérons par exemple l'identification des régions filles de la région 2. La région 5 (dont le brin représentant est b) est la première fille de 2 ($successeur(2)=5$). Lors du parcours de l'orbite composante connexe de son brin représentant ($\langle \delta_1 \circ \delta_2 \rangle(b)$), nous obtenons six régions 5, 6, 7, 8, 9 et 10 dont le parent est 2. Étant donné qu'aucune de ces régions ne contient de région imbriquée, la procédure d'identification des régions imbriquées n'ajoute

3.4. Utilisation pratique des relations hiérarchiques

Algorithme 14 : identifier_régions_imbriquées

Données : Soit r^k une région de G^k .

Soit r une région de G^{k+1} telle que $\text{prédécesseur}(r) = r^k$.

Résultat : res : liste des régions imbriquées dans r et dont le prédécesseur est r^k .

$res \leftarrow$ liste vide de régions ;

ajouter r à res ;

- 1 **pour** chaque $r' \in \text{fils}(r)$ **faire**
- 2 **si** $\text{prédécesseur}(r') = r^k$ **alors**
- 3 ajouter r' à la fin de res ;
- 3 $res \leftarrow res \cup \text{identifier_régions_imbriquées}(r', r^k)$;

retourner res ;

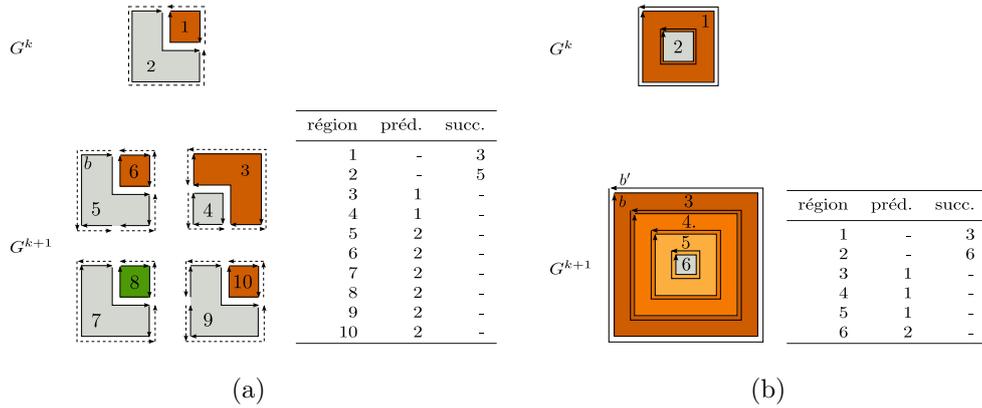


FIGURE 3.11 – Identification des régions filles entre deux niveaux consécutifs d'une pyramide descendante tuilée. (a) Cas d'une région répartie sur plusieurs tuiles; (b) Cas des régions imbriquées.

aucune région supplémentaire. La région 2 du niveau k correspond ainsi aux régions 5, 6, 7, 8, 9 et 10 du niveau $k + 1$.

La Figure 3.11(b) met en avant le cas particulier des relations d'imbrication. Aussi, considérons le problème de l'identification des régions filles de la région 1. La région 3 (dont le brin représentant est b) est la première fille de 1 ($\text{successeur}(1)=3$). En revanche, le parcours des régions voisines par l'orbite $\langle \delta_1 \circ \delta_2 \rangle$ ne nous permet pas d'obtenir les régions imbriquées 4 et 5 ($\langle \delta_1 \circ \delta_2 \rangle(b) = \{b, b'\}$). En revanche, notre procédure d'identification des régions imbriquées permet de les récupérer ($4 \in \text{fils}(3)$). Puisque l'algorithme est récursif, nous obtenons de la même façon la région 5 imbriquée dans 4.

Notons que la région 6 est bien ignorée par la procédure car sa région parente est la région 2.

3.5 Construction d'une pyramide descendante tuilée

Dans cette section, nous proposons un processus de construction d'une pyramide descendante tuilée à partir d'une image multi-résolution. Nous adoptons une approche descendante se décomposant en deux étapes principales permettant de construire le niveau $k + 1$ d'une pyramide à partir du niveau précédent k . Dans un premier temps, nous réalisons une projection des frontières interpixel de G^k au niveau $k + 1$ pour former G'^k (Section 3.5.1). Cette projection nous permet de garantir la propriété de causalité de notre modèle. Dans un second temps, nous proposons un algorithme pour l'extraction des successeurs d'une tuile par focus d'attention, permettant de raffiner uniquement les régions d'intérêt du niveau k (Section 3.5.2). Notons que nous interdisons la suppression des frontières insérées par l'étape de projection lors de l'extraction. Dans la Section 3.5.3, nous étendons cet algorithme à la construction de notre modèle pyramidal.

3.5.1 Projection des frontières

Dans le cadre de la construction d'une pyramide descendante tuilée, nous proposons deux implantations différentes du processus de projection d'un $(0, 1)$ -chemin (Section 3.3.1) que nous appelons projection *directe* et projection *optimisée*. Dans chaque cas, nous considérons le problème de la projection d'un $(0, 1)$ -chemin $c = (p_0, l_0, \dots, l_{n-1}, p_n)$ du niveau k sur un $(0, 1)$ -chemin $c' = (p'_0, l'_0, \dots, l'_{m-1}, p_m)$ du niveau $k + 1$.

Projection directe

Une première approche pour déterminer le $(0, 1)$ -chemin projeté consiste à sélectionner pour chaque pointel p_i appartenant à c , le pointel $p'_i = P_{direct}(p_i)$. Nous relierons ensuite p'_i et p'_{i+1} ($0 \leq i < n$) par une ligne droite de longueur r_x^k si le lignel incident à p_i est horizontal, r_y^k sinon (Figure 3.12). Cette projection a l'avantage d'être facile à mettre en œuvre. En revanche, cette méthode induit plusieurs inconvénients. Nous constatons par exemple des effets de « blocs » en raison du choix des lignes droites pour la sélection de $(0, 1)$ -chemins du niveau $k + 1$. Cet effet est d'autant plus important que le ratio entre deux

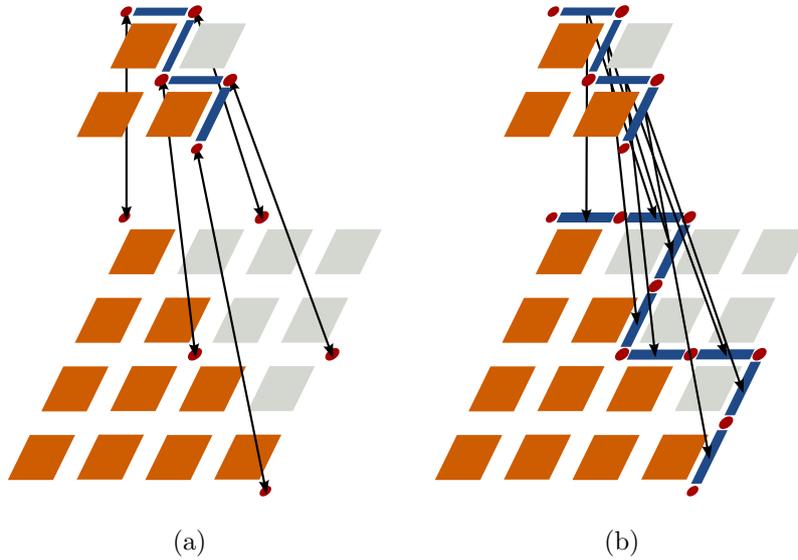


FIGURE 3.12 – Projection directe. (a) Sélection des pointels successeurs en utilisant les projetés directs des pointels parents; (b) Sélection d'un $(0,1)$ -chemin en reliant par des lignes droites deux pointels successeurs consécutifs.

niveaux consécutifs est grand. En outre, il entraîne l'apparition de nombreuses petites régions au niveau du bord des régions lors du raffinement du niveau $k + 1$ pour compenser la projection grossière. Notons que cette méthode peut être partiellement améliorée en reliant p'_i à p'_{i+2} (effet « escalier »). Enfin, nous pouvons noter que cette projection ne tire pas parti de l'information présente à la résolution $k + 1$ pour effectuer la projection.

Projection optimisée

Nous proposons d'améliorer la projection directe en tenant compte des informations additionnelles du niveau $k + 1$ pour affiner la géométrie des contours. Les deux couples (p'_0, l'_0) et (l_{m-1}, p_m) marquant les extrémités du chemin c' sont imposés pour éviter les problèmes de déconnexion ou de recouvrement au niveau des nœuds (Section 3.3.1). Nous utilisons également l'algorithme de Dijkstra pour déterminer le plus court $(0,1)$ -chemin au niveau $k + 1$ entre p'_1 et p'_{m-1} en pondérant les arêtes par une énergie à minimiser. Pour cela, nous établissons un graphe dont les nœuds correspondent aux pointels des fenêtres de réduction des pointels de c et dont les arêtes sont les lignels. Un chemin optimal entre p'_1 et p'_{m-1} est obligatoirement trouvé du fait de la connexité de l'union des fenêtres de réduction (Corollaire 2). Par exemple, nous

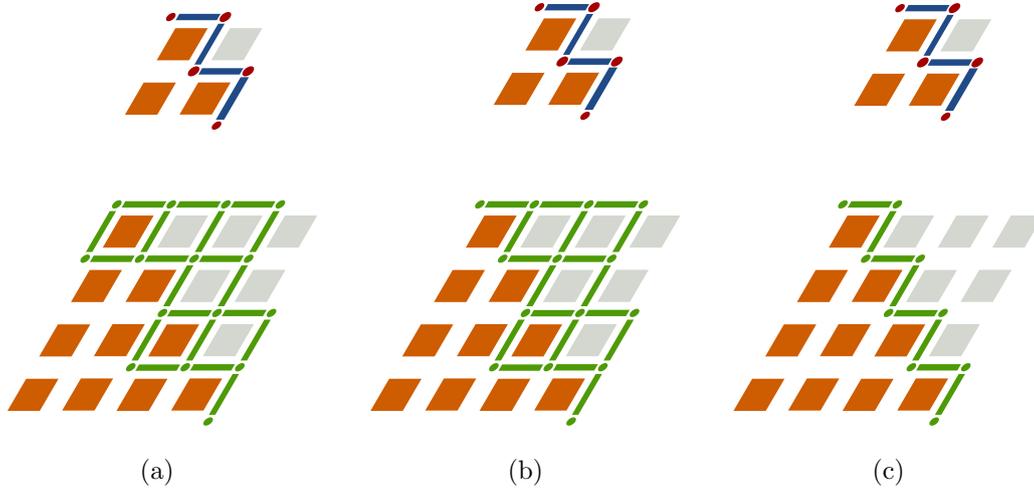


FIGURE 3.13 – Projection optimisée d’un $(0, 1)$ -chemin. (a) Bande de projection; (b) Restrictions aux extrémités du chemin pour éviter les effets de recouvrement; (c) Calcul d’un plus court chemin.

proposons d’utiliser l’inverse du gradient calculé entre les deux pixels incidents à un lignel. À partir de ce graphe pondéré, l’algorithme de Dijkstra [Dij59] nous fournit un $(0, 1)$ -chemin minimal entre les deux successeurs des pointels extrémités du chemin c' . Une illustration des étapes principales de ce processus de projection est proposée dans la Figure 3.13.

3.5.2 Extraction des successeurs d’une tuile

L’extraction d’une pyramide descendante tuilée consiste à fournir une hiérarchie de partitions d’une image. Étant donné que nous utilisons une structure tuilée, l’extraction s’effectue tuile par tuile de façon incrémentale. Nous garantissons ainsi la faisabilité mémoire du processus de construction. Par conséquent, nous présentons une méthode de construction permettant d’extraire l’ensemble des tuiles filles de niveau $k + 1$ correspondant à la même tuile parente du niveau k que nous notons $t = t(i, j, k)$ dans la suite de cette section.

Nous avons défini dans la Section 3.5.1, l’opération de projection des frontières entre deux niveaux consécutifs qui garantit la causalité de la structure. L’objectif de cette section est de proposer une méthode de construction descendante combinant cette opération de projection avec l’algorithme d’extraction d’une carte tuilée (Section 2.4.3). Notre méthode de construction tire parti de la définition descendante du modèle de pyramide tuilée en proposant un

3.5. Construction d'une pyramide descendante tuilée

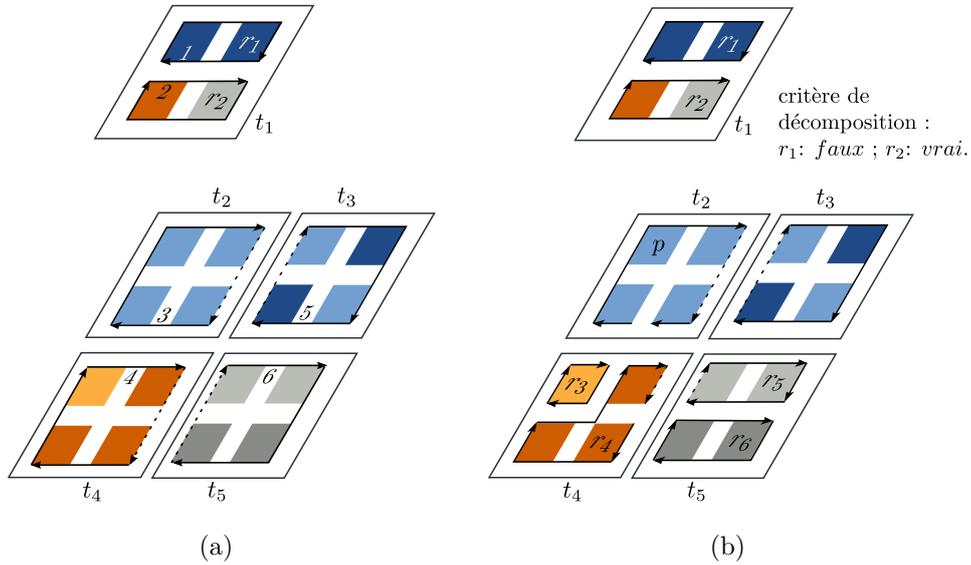


FIGURE 3.14 – Extraction des successeurs $\{t_2, t_3, t_4, t_5\}$ de la tuile t_1 . La pyramide d'images est composée de deux niveaux avec un ratio $r_x = r_y = 2$. Les tuiles sont de taille constante (2×2 pixels). (a) Projection : l'étape de projection des frontières garantit la causalité du modèle en préservant les bords existants au niveau supérieur ; (b) Raffinement par focus d'attention : seules les régions dont le *critère de décomposition* est *vrai* sont raffinées.

raffinement par focus d'attention des régions d'intérêt. La propriété de focus d'attention repose sur la définition d'un *critère de décomposition* que nous considérons comme un prédicat $P : \mathcal{R} \rightarrow \{vrai, faux\}$. Ce prédicat indique si une région est une région d'intérêt devant être raffinée au niveau suivant. Par conséquent, l'opération d'extraction ne doit insérer aucun bord à l'intérieur des régions dont le critère de décomposition est *faux*.

Ces deux étapes sont illustrées dans la Figure 3.14 sur l'exemple d'une pyramide descendante tuilée composée de deux niveaux k et $k + 1$ associés à deux images I^k et I^{k+1} ($r_x^k = r_y^k = 2$). Dans la Figure 3.14(a), la projection des frontières des régions r_1 et r_2 garantit la causalité de la pyramide. Par exemple, l'arête (1,2) marquant le bord entre les régions r_1 et r_2 de G^k se retrouve sous la forme de deux arêtes (3,4) entre t_2 et t_4 et (5,6) entre t_3 et t_5 . De plus, cette étape définit les relations *prédécesseur/successeur* entre G^k et G^{k+1} : lorsque les brins 3 et 4 sont créés, ils sont reliés à leurs *prédécesseurs* respectifs 1 et 2. Dans la Figure 3.14(b), la construction par focus d'attention ne raffine que les régions dont le critère de décomposition est *vrai*. Ainsi, aucun bord n'est inséré

dans les régions *successeurs* de r_1 tandis que la région r_2 est décomposée en quatre régions ($\{r_3, r_4\}$ appartenant à t_4 et $\{r_5, r_6\}$ appartenant à t_5).

L'opération d'extraction intègre l'étape de projection et la notion de focus d'attention dans la fonction *Oracle* déterminant si deux régions sont identiques. Cette fonction *Oracle* est ainsi redéfinie suivant l'Algorithme 15. Une frontière projetée est caractérisée par les lignels qui la constituent dans la matrice de Khalimsky de niveau $k + 1$. Dans ce cas, le bord est conservé pour garder la propriété de causalité (ligne 1). Lorsque le critère de décomposition de la région parente est faux, la région fille ne doit pas être raffinée (ligne 2). Afin de retrouver efficacement la région d'un pixel donné, nous utilisons une structure temporaire similaire à une image de labels. Nous pouvons ainsi déterminer rapidement le critère de décomposition de la région contenant le parent du pixel traité. Par exemple, dans la Figure 3.14(b), nous pouvons déterminer que la région de *prédécesseur*(p) est la région r_1 dont le critère de décomposition est *faux*. Par conséquent, la région r_1 ne doit pas être raffinée au niveau $k + 1$. En dehors de ces deux cas de figure, les régions peuvent être raffinées (ligne 3).

Algorithme 15 : Oracle

Données : Deux pixels adjacents p et p' d'une tuile $t \in G^{k+1}$.

Résultat : vrai si p et p' appartiennent à la même région et faux sinon.

$l \leftarrow$ lignel incident à p et p' ;

si l est un lignel frontière **alors**

1 \lfloor retourner *faux*;

si *critère_de_décomposition*(région(*prédécesseur*(p))) est *faux* **alors**

2 \lfloor retourner *vrai*;

3 retourner *même_région*(région(p), région(p'));

La procédure de construction des tuiles successeurs à partir de leur tuile parente est détaillée dans l'Algorithme 16. Cette opération intègre l'opération de projection des frontières du niveau précédent (construction de G'^k) avant d'effectuer le raffinement des régions (construction de G^{k+1}). Nous avons mis en œuvre deux méthodes pour cette dernière opération. La première réutilise le principe d'extraction d'une carte (Section 2.4) tandis que la seconde s'appuie sur le raffinement par décomposition/fusion (Section 2.5). Ces deux méthodes sont détaillées ci-après.

Raffinement par extraction

De la même manière que pour l'extraction d'une carte tuilée (Algorithme 4), nous utilisons l'extraction d'une carte topologique (Algorithme 1) sur une

Algorithme 16 : extraction_succeurs

Données : Une pyramide dont le niveau $k + 1$ est en cours de construction et dont les régions ont les bords composés d'arêtes élémentaires.
 Une tuile $t(i, j, k)$ appartenant à la pyramide.
 Un *Oracle* défini suivant l'Algorithme 15.

Résultat : Extraction des *succeurs* de t au niveau $k + 1$.

```

pour  $y = 0$  à  $r_y^k$  faire
  pour  $x = 0$  à  $r_x^k$  faire
     $t \leftarrow t(i \times r_x^k + x, j \times r_y^k + y, k + 1)$ ;
    projeter les frontières de  $t(i, j, k)$  sur  $t$ ;
1   extraire_carte( $t$ , Oracle);
    si il existe une tuile  $t'$  adjacente à gauche à  $t$  alors
2     | connecter_tuiles( $t, t'$ );
    si il existe une tuile  $t'$  adjacente en haut à  $t$  alors
3     | connecter_tuiles( $t, t'$ );
  
```

subdivision de la résolution $k + 1$ de l'image I (ligne 1 de l'Algorithme 16). L'aspect hiérarchique intervient au niveau de la fonction *Oracle* définie par l'Algorithme 15 lors de l'extraction de la carte topologique d'une des tuiles succeurs (ligne 1). En effet, l'*Oracle* prend en compte le critère de décomposition de la région parente. Notons que pour des raisons d'optimisation, l'étape de simplification du processus d'extraction (ligne 4 de l'Algorithme 1) n'est pas appliquée aux brins de \mathcal{B}_θ^t étant donné que la simplification est effectuée lors de la connexion de t avec ses tuiles adjacentes (ligne 2 et ligne 3 de l'Algorithme 16).

Raffinement par décomposition/fusion

Nous pouvons réutiliser la méthode décrite dans la Section 2.5 pour raffiner au niveau $k + 1$ une région du niveau k dont le critère de décomposition est *vrai*. Contrairement au raffinement par extraction, cette méthode effectue un raffinement global en décomposant la région projetée en régions élémentaires puis en les fusionnant d'après un critère de fusion. Afin de ne pas supprimer les frontières projetées, nous restreignons l'opération de fusion aux régions issues d'un même parent. Néanmoins, cette approche est plus coûteuse en mémoire que le raffinement par extraction dans la mesure où l'étape de décomposition produit un nombre important de brins et de régions (une région et quatre

brins par pixel). Par la suite, nous utilisons le raffinement par extraction pour la construction de notre modèle.

3.5.3 Extraction d'une pyramide

À travers l'Algorithme 16, nous avons proposé une méthode pour extraire une tuile d'un niveau $k + 1$ à partir de sa tuile parente du niveau k . Dans cette section, nous étendons à présent ce processus d'extraction à l'ensemble des niveaux composant une pyramide descendante tuilée. Nous adoptons un procédé descendant itératif construisant un niveau d'après le précédent (Algorithme 17).

La première étape de la construction d'une pyramide est la construction de la partition correspondant à la carte G^0 du premier niveau que nous obtenons par extraction de la carte tuilée correspondant à la plus faible résolution d'une pyramide d'images (ligne 1).

La seconde étape établit les différents critères de segmentation et permet ainsi de définir la fonction *Oracle* utilisée pour le raffinement (ligne 2).

La troisième étape extrait les tuiles *successeurs* de chaque tuile du niveau précédent. L'opération utilise l'Algorithme 16 (ligne 3) lors d'un balayage ligne par ligne partant de la tuile en haut à gauche jusqu'à la tuile en bas à droite. Notons que les tuiles *prédécesseur*, à partir desquelles sont effectuées les extractions, sont déchargées à la fin de l'extraction de leurs successeurs. Le processus ne requiert alors qu'un nombre limité de tuiles en mémoire ce qui nous permet de garantir la faisabilité de notre méthode.

Algorithme 17 : extraction_pyramide

Données : Une pyramide de tuiles $(\mathcal{I} = (I^0, \dots, I^m), \mathcal{T})$ associée à une pyramide d'images, avec l et h la largeur et la hauteur d'une tuile.
 Une fonction *Oracle*.

Résultat : Extraction de la pyramide descendante tuilée $P = (\mathcal{T}, \mathcal{G})$.

```

1  $G^0 \leftarrow \text{extraire\_carte\_tuilée}(I^0, \text{Oracle}, l, h);$ 
  pour  $k = 0$  à  $m - 1$  faire
2   Soit Oracle la fonction définie par Algorithme 15 au niveau  $k$ ;
   pour  $j = 0$  à  $H^k/h - 1$  faire
     pour  $i = 0$  à  $L^k/l - 1$  faire
3     [  $\text{extraire\_successeurs}(t(i, j, k), I^{k+1}, \text{Oracle});$ 
       simplification des sommets de degré 2 du niveau  $k + 1$ ;
     ]
  
```

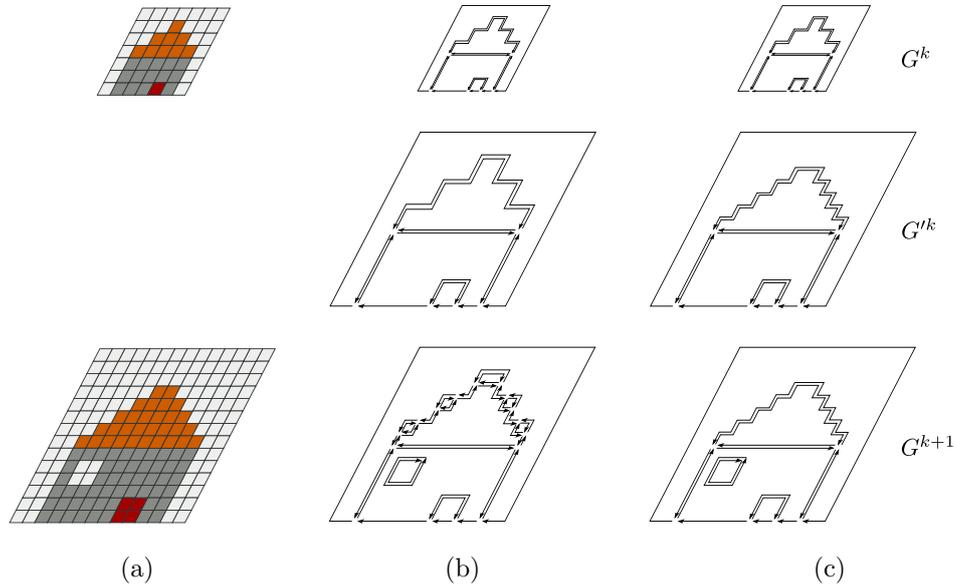


FIGURE 3.15 – Comparaison du processus d'extraction après l'étape de projection des bords des régions. (a) Une image multi-résolution composée de deux niveaux avec un ratio 2×2 ; (b) Extraction de G^{k+1} après projection directe des bords des régions de G^k ; (c) Extraction de G^{k+1} après projection optimisée des bords des régions de G^k .

Sur l'exemple de la Figure 3.15, nous illustrons la partition obtenue par notre procédé d'extraction après une projection directe des bords des régions (Figure 3.15(b)) et une projection optimisée (Figure 3.15(c)).

3.6 Conclusion

Au cours de ce chapitre, nous avons défini un nouveau modèle de pyramide descendante tuilée adapté au traitement de grandes images hiérarchiques. Notre définition repose sur la combinaison de trois modèles hiérarchiques : une pyramide régulière d'images comportant un ensemble de résolutions de plus en plus grandes, une pyramide régulière de tuiles définissant un pavage de chaque image et une pyramide irrégulière de cartes tuilées successivement raffinées (Section 3.2). Notre modèle permet ainsi de représenter les relations géométriques, topologiques et hiérarchiques d'une image multi-résolution.

Nous avons ensuite établi un ensemble de relations hiérarchiques entre deux niveaux consécutifs d'une pyramide. Ces relations nous ont permis de définir un processus de projection respectant la causalité du modèle (Section 3.3).

À partir de cet ensemble de relations hiérarchiques, nous avons illustré l'utilisation de notre modèle à travers deux exemples permettant de retrouver les arêtes et les régions filles dans la pyramide après une opération de découpe à un niveau supérieur (Section 3.4).

Dans la Section 3.5, nous avons étudié la construction de notre modèle en détaillant les deux étapes de projection et de raffinement. Notre opération de projection garantit une structure causale en préservant les frontières d'un niveau à l'autre et le raffinement s'effectue par focus d'attention.

L'originalité de notre modèle vient de son procédé de construction descendant : chaque niveau est composé d'une carte tuilée qui correspond à un raffinement du niveau précédent. L'avantage d'utiliser un procédé de construction descendant par rapport à une construction ascendante est double. Premièrement, le modèle se construit suivant la notion de focus d'attention qui est une approche naturelle de l'analyse psycho-visuelle d'une grande image et reproduit le comportement des pathologistes dans le cadre du traitement d'images histologiques. Deuxièmement, cette notion de focus d'attention permet de ne pas raffiner certaines régions (comme le fond de l'image) ce qui permet de minimiser l'occupation mémoire du modèle. Ce dernier point est particulièrement important dans notre cas en raison des contraintes techniques imposées par les dimensions de l'image.

Dans le chapitre suivant, nous étudions l'utilisation de notre modèle de pyramide descendante tuilée pour la représentation et la segmentation de grandes images multi-résolution.

Chapitre 4

Segmentation et expérimentations

Sommaire

4.1	Introduction	146
4.2	Segmentation de grandes images	146
4.2.1	Préliminaires	147
4.2.2	Approches par blocs	148
4.2.3	Approches hiérarchiques	150
4.2.4	Approches par architecture matérielle dédiée	152
4.2.5	Bilan	153
4.3	Expérimentations sur le modèle	153
4.3.1	Préliminaires	154
4.3.2	Segmentation par blocs avec le modèle de carte tuilée	157
4.3.3	Segmentation hiérarchique avec le modèle de pyramide descendante	163
4.4	Utilisation du modèle pour la segmentation d'images histologiques	165
4.4.1	Processus de segmentation	167
4.4.2	Extraction des mitoses	170
4.5	Conclusion	171

4.1 Introduction

L'objectif de ce chapitre est de préciser l'apport de notre modèle de pyramide descendante tuilée à la segmentation d'images et de démontrer son applicabilité à travers un exemple précis. La construction d'une carte tuilée (Chapitre 2) et d'une pyramide descendante tuilée (Chapitre 3) repose en partie sur la définition d'un *Oracle*. Cet *Oracle* définit une partition de l'image en régions et donc une segmentation s'il capture une notion d'homogénéité de régions. La segmentation d'images constitue ainsi le champ d'expérimentation privilégié de notre modèle.

Dans ce chapitre, nous commençons par étudier dans la Section 4.2, différentes méthodes de segmentation. Le domaine de la segmentation étant très vaste, nous avons restreint notre étude aux méthodes utilisées pour la segmentation de grandes images. Nous nous sommes focalisés sur les applications en microscopie et plus particulièrement aux domaines de l'histologie qui présente les plus grands défis pour l'analyse d'images. Dans la Section 4.3, nous étudions l'apport que peut représenter notre modèle pour la segmentation de grandes images multi-résolution. Dans la Section 4.4, nous présentons l'intégration pratique d'un processus de segmentation au sein de notre modèle pour le traitement des images d'histologie.

4.2 Segmentation de grandes images

L'opération de segmentation d'une image consiste à utiliser un critère d'homogénéité pour définir une partition en régions homogènes. Il s'agit d'une étape importante puisqu'elle permet de mettre en évidence les zones d'intérêt utiles à l'utilisateur. De façon générale, l'objectif consiste à construire un critère adapté à l'information qu'il s'agit d'extraire de l'image comme les zones de couleurs proches ou la forme des contours d'une région. Ces deux derniers exemples constituent deux grandes familles de critères de segmentation : les approches dites *basées région* et les approches dites *basées contour*. Il est néanmoins très réducteur de limiter les méthodes de segmentation à ces deux catégories d'approches. Le domaine de la segmentation est en effet très vaste et a donné lieu à de très nombreuses méthodes (plus d'un millier selon Serra [Ser06]) que nous ne pouvons pas toutes décrire dans le cadre de ce manuscrit. Nous avons ainsi choisi de focaliser notre étude sur le cas de la segmentation des grandes images et plus particulièrement sur les grandes images histologiques.

Dans la Section 4.2.1, nous proposons une définition de l'opération de segmentation. Nous détaillons les différentes méthodes développées pour

le traitement des grandes images en les regroupant suivant trois types d'approches : les approches par blocs (Section 4.2.2), les approches hiérarchiques (Section 4.2.3) et les approches par architecture matérielle dédiée (Section 4.2.4).

4.2.1 Préliminaires

La segmentation d'une image repose essentiellement sur la définition d'un critère d'homogénéité. Ce critère se construit à l'aide d'un prédicat $P : \mathcal{R} \rightarrow \{\text{vrai}, \text{faux}\}$, défini sur l'ensemble \mathcal{R} des régions d'une partition de l'image et indiquant si une région est homogène d'après le critère considéré. Horowitz et Pavlidis définissent la notion de segmentation d'une image [HP76] que nous rappelons dans la Définition 65. Notons que pour définir une partition, le prédicat doit être consistant ce qui impose que tout sous-ensemble d'une région homogène est forcément homogène.

Définition 65 (Segmentation en régions homogènes). *Soit I une image et P un prédicat. La segmentation en régions homogènes de I pour le prédicat P est une partition \mathcal{R} de I telle que :*

- $P(r) = \text{vrai}$ pour toute région $r \in \mathcal{R}$;
- $P(r_i \cup r_j) = \text{faux}$ pour tout couple $(r_i, r_j) \in \mathcal{R}^2$, $i \neq j$, de régions r_i et r_j adjacentes.

Notons que puisque \mathcal{R} forme une partition de I (voir Définition 21, page 25), la réunion des régions homogènes représente l'image entière ($I = \bigcup_{r \in \mathcal{R}} r$).

Une des contraintes majeures soulevées par l'application d'une méthode de segmentation à une image de grande taille est un problème technique relatif à l'espace mémoire disponible. En effet, les lames virtuelles permettent d'acquérir une image avec un grossissement $\times 20\,000$ ou $\times 40\,000$ pour un prélèvement moyen de 4 cm^2 de tissu. La taille d'une image (sans compression) atteint alors $4,5\text{ Go}$ (grossissement $\times 20\,000$) voire 18 Go (grossissement $\times 40\,000$). De plus, les algorithmes de segmentation requièrent généralement des structures de données demandant plusieurs fois la taille de l'image en mémoire. Par conséquent, dans la plupart des cas, il n'est pas envisageable d'utiliser directement les algorithmes de segmentation disponibles car les ressources nécessaires en temps de calcul ou en mémoire sont alors trop importantes.

De plus, en raison de la taille très importante des images, celles-ci sont généralement enregistrées via un format de compression avec perte comme le JPEG. Dans le cas des images médicales, le ratio de compression peut aller jusqu'à 40 :1. L'algorithme de compression JPEG atténue les hautes fréquences

de l'image car elles sont peu perceptibles à l'œil mais peut introduire des effets de blocs gênants ou entraîner des pertes d'informations sensibles pour les algorithmes de traitement.

Comparé au nombre de méthodes de segmentation développées, il existe relativement peu de méthodes spécifiques aux très grandes images. Cela s'explique simplement par le fait que les méthodes d'acquisition d'images à très haute résolution sont assez rares et réservées à des domaines comme l'imagerie satellite ou médicale. De plus, les scanners permettant d'acquérir l'image complète à haute résolution sont récents et peu de méthodes ont été développées pour leur segmentation.

Notons enfin que la segmentation est une étape généralement coûteuse en ressources. D'après Felzenszwalb, les méthodes de segmentation sont estimées efficaces lorsque leur complexité s'exprime linéairement ou *quasi*-linéairement d'après le nombre de pixels [FH04]. Dans le cas des grandes images, l'étape de segmentation est ainsi intrinsèquement coûteuse étant donné le nombre très important de pixels contenus à la plus forte résolution.

4.2.2 Approches par blocs

Le premier type d'approches décompose le problème en sous-parties en s'appuyant sur une décomposition en blocs (ou tuiles) de l'image. Cette décomposition découpe l'image en un ensemble d'images qui sont traitées indépendamment. Le problème de la segmentation se ramène alors à l'échelle d'un bloc. D'un point de vue technique, cette approche a l'avantage d'être facilement parallélisable. En revanche, les méthodes de segmentation utilisant des critères globaux peuvent être mises en défaut car les caractéristiques globales d'une tuile peuvent être très différentes des caractéristiques globales de l'image. De plus, la segmentation produite peut entraîner différents effets indésirables au niveau du bord des blocs. En effet, les objets tronqués sont mal détectés et leur dénombrement est délicat car ils sont répartis sur plusieurs tuiles. Une technique classique pour limiter l'impact de ces effets de bord consiste à définir un facteur de recouvrement entre les tuiles [CPR03].

Cette approche par blocs du traitement de grandes images a été utilisée par la société Aperio Technologies pour la quantification de marquages membranaires et nucléaires. L'image est décomposée en un ensemble d'images de taille $1\,000 \times 1\,000$ qui sont ensuite traitées indépendamment par des méthodes de seuillage couleur et de morphologie mathématique pour l'extraction de la taille, de la forme et du nombre de noyaux. Le recouvrement entre les différentes images est calculé suivant la taille théorique maximale du diamètre d'un

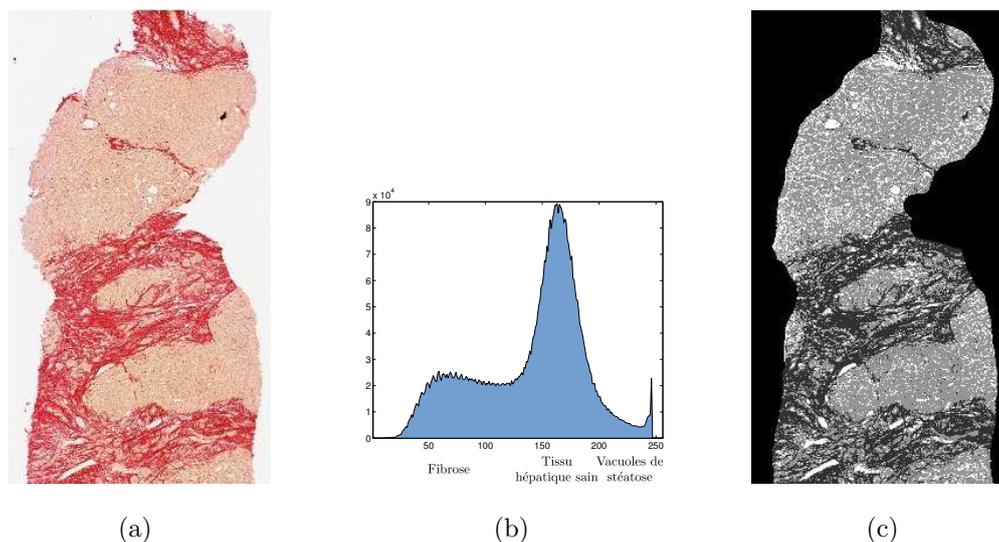


FIGURE 4.1 – Un exemple d’approche par blocs pour la segmentation et la classification des vacuoles de stéatose tiré de [Rou08]. (a) Image originale ($15\,000 \times 25\,000$ pixels); (b) Histogramme permettant de distinguer trois classes : la fibrose, le tissu hépatique sain et les vacuoles de stéatose; (c) Classification de la stéatose (blanc), du tissu sain (gris) et de la fibrose (noir).

noyau. Pour des raisons de performance, le traitement est restreint à des zones d’intérêt [Ols06].

Une approche par blocs a également été proposée par Roullier *et al.* [Rou08] pour la segmentation et la classification de la stéatose et de la fibrose hépatique sur lames histologiques. Fondée sur une décomposition en blocs de $5\,000 \times 5\,000$ pixels, cette méthode se déroule en trois étapes. La première étape est la construction incrémentale de l’histogramme de l’image. La seconde étape effectue une classification en trois classes à partir de l’histogramme précédemment construit. Ces trois classes représentent les vacuoles de graisse (stéatose), le tissu sain (parenchyme hépatique) et la fibrose. La dernière étape du processus de segmentation et de classification consiste à reconstruire les partitions correspondant aux trois classes extraites. Là encore, l’opération est effectuée bloc par bloc en raison de l’impossibilité de charger l’ensemble de l’image ($15\,000 \times 25\,000$ pixels) en mémoire.

D’autres méthodes utilisant des approches par blocs ont été mises en œuvre pour des applications médicales en histologie. Par exemple, Mete *et al.* proposent une classification d’imagelettes de 128×128 pixels à partir d’un histogramme couleur pour la reconnaissance des zones malignes dans les cancers

de la tête et du cou [MXFS07]. Mosaliganti *et al.* utilisent des méthodes à base de patches pour l'évaluation des changements de phénotypes sur des embryons de souris [MPR⁺08].

Si une approche par blocs permet généralement de garantir la faisabilité mémoire d'un processus de segmentation appliqué à une grande image, le temps de calcul reste néanmoins important et constitue une contrainte majeure. Il est alors intéressant de combiner l'approche tuilée avec une approche hiérarchique.

4.2.3 Approches hiérarchiques

Les approches hiérarchiques sont naturellement utilisées pour la segmentation de grandes images. En effet, puisque l'objectif d'une segmentation consiste à identifier une structure donnée, un niveau de détail très élevé n'est pas toujours nécessaire selon le type de la structure recherchée [SPHR08]. Intuitivement, l'avantage principal apporté par une approche hiérarchique est qu'il est possible d'utiliser la segmentation obtenue à moindre coût d'un niveau k pour accélérer la segmentation du niveau $k + 1$ (approche descendante).

Traitements à faible résolution et décimations

L'approche hiérarchique la plus simple à mettre en place consiste à effectuer une décimation de l'image de plus forte résolution, appliquer le traitement sur l'image décimée et projeter le résultat sur l'image originale lors de la visualisation. Par exemple, cette méthode a été utilisée pour l'étude de la structure crânienne et de la quantification vasculaire [HLSP06, DPLH08]. Ces traitements à faible résolution sont en effet suffisants pour des applications comme le dénombrement et la répartition de compartiments cellulaires ou de structures tissulaires comme ont pu le démontrer les contrôles de qualité effectués par Françoise *et al.* [FMPH05] et Elie *et al.* [ELMH05]. En revanche, l'inconvénient immédiat de ces approches est qu'il n'est pas possible d'utiliser le niveau de détail cellulaire apporté par la plus forte résolution. Une approche mixte proposée par Fiscor *et al.* [FVT⁺08] consiste à segmenter les noyaux de l'épithélium à forte résolution tandis que les glandes sont segmentées à partir d'une décimation ($\times 8$) de l'image originale.

Approches multi-résolution

Une approche multi-résolution présente plusieurs intérêts. Tout d'abord, il est possible d'utiliser une même méthode de segmentation pour l'ensemble de la pyramide mais en adaptant les paramètres de l'algorithme à la résolution

d'un niveau donné. Par exemple, dans le cadre d'une application pour la quantification du stroma d'images de neuroblastome, la méthode proposée par Sertel *et al.* [SKS⁺09] produit une classification fondée sur l'algorithme des k plus proches voisins dont le seuil d'appartenance à une même classe est raffiné suivant la résolution. De la même façon, des méthodes d'apprentissage pour un ensemble de caractéristiques comme la moyenne, la variance, l'entropie ou l'homogénéité ont permis à Kong *et al.* [KSS⁺07] d'accélérer de moitié le temps de traitement nécessaire à la classification d'images de neuroblastome.

Une autre façon d'exploiter une structure hiérarchique consiste à adapter un traitement spécifique à chaque niveau de résolution suivant les objets les plus pertinents à identifier. Par exemple, des travaux sur l'analyse de zones inflammatoires dans les poumons effectués par Apfeldorfer *et al.* [AUJ⁺08] utilisent différents seuillages ainsi qu'un ensemble de règles spécifiques à chaque type d'objet à identifier à une résolution donnée.

Combinaison avec les approches par blocs

Les approches hiérarchiques sont souvent combinées avec une décomposition par blocs (ou tuiles) des différents niveaux de résolution. Par exemple, une stratégie multi-résolution a été retenue pour la segmentation de l'épithélium squameux pour le traitement du cancer du col de l'utérus [HZL05, Wan08]. La démarche générale de la méthode proposée consiste à effectuer une classification des blocs à faible résolution puis d'affiner les frontières de régions à haute résolution. Une analyse de la répartition, densité et surface des noyaux est enfin effectuée à pleine résolution pour effectuer le classement final d'images de taille 250×250 . Le traitement d'une image de taille $152\,000 \times 42\,000$ est estimé à environ 3 heures (Intel PentiumIV@3,4GHz, 2 Go de mémoire RAM).

Une approche multi-résolution a été proposée par [RLTE10] pour l'extraction et la classification des mitoses d'images histologiques. Cette approche hiérarchique est combinée avec une décomposition par blocs de 240×240 pixels des niveaux de la pyramide. Chaque niveau est segmenté par un algorithme de classification non-supervisée en deux classes après une étape de régularisation. Chaque niveau est construit depuis le précédent (approche descendante) en effectuant une réplique de l'étiquetage des régions d'intérêt par une projection standard. L'effet d'escalier (*stair-casing* en anglais) engendré par cette projection est compensé par un algorithme de raffinement spatial, appliqué sur une bande étroite autour du contour. Ce processus est illustré sur la Figure 4.2.

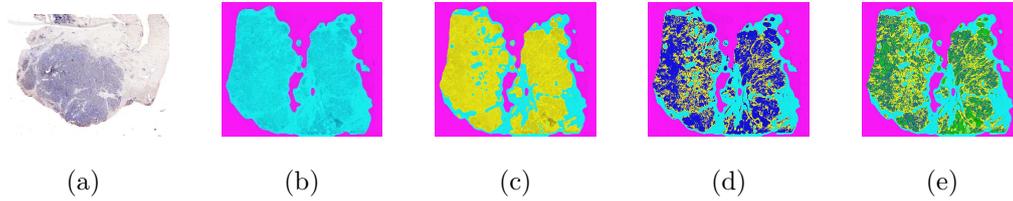


FIGURE 4.2 – Un exemple d’approche hiérarchique pour la segmentation et la classification des images histologiques tiré de [RLTE10]. (a) Image originale; (b) Niveau 1 : différenciation entre le fond et le tissu; (c) Niveau 2 : différenciation entre le tissu et les lésions; (d) Niveau 3 : différenciation entre le cancer in situ et le stroma; (e) Niveau 4 : différenciation entre le cordon tumoral et le reste.

4.2.4 Approches par architecture matérielle dédiée

Un dernier type d’approche pour le traitement de grandes images consiste à traiter le problème de façon matérielle pour accélérer les temps de traitement. À cette fin, les algorithmes de segmentation doivent être parallélisés pour répartir la charge de travail sur un nombre plus importants d’unités de calcul. Il devient alors possible de tirer parti de la puissance de traitement des cartes graphiques ou des grilles de calcul. Le cas des grandes images d’histologie se prête bien à ces solutions matérielles dans la mesure où le traitement d’un bloc se fait de façon indépendante. Les approches par blocs (Section 4.2.2) peuvent ainsi facilement tirer parti d’une répartition parallèle des traitements : à chaque unité de calcul est attribuée une liste de blocs à traiter et il n’y a pas de synchronisation particulière à effectuer entre les données.

Parmi l’ensemble des solutions matérielles dédiées à la segmentation des grandes images, nous pouvons distinguer les méthodes utilisant le GPU (*Graphics Processing Unit* en anglais) des cartes graphiques, les grilles de calculateurs ou encore certaines cartes externes et dédiées à certaines opérations spécifiques particulièrement coûteuses. L’avantage immédiat de ces solutions matérielles est une réduction importante du temps de calcul suivant la puissance fournie par l’architecture retenue. En revanche, ces solutions peuvent être onéreuses et délicates à mettre en place. Concernant les cartes graphiques, leur principal inconvénient vient de leur difficulté de programmation malgré l’émergence récente des interfaces de développements comme CUDA ou OpenCL.

Par exemple, les travaux de Kong *et al.* [KSS+07] sur la classification d’images de neuroblastome (Section 4.2.3) ont été portés sur une grille de calcul disposant de 64 unités de traitement (ou nœud de calcul) correspondant

chacune à un processeur dual cœur. Cette architecture permet par exemple de traiter une image de $75\,000 \times 68\,000$ pixels en une demi-heure. Les cartes graphiques disposent de nombreuses unités de calcul spécifiques très efficaces et utilisées dans le domaine du jeux vidéo et du rendu 3D pour un coût peu élevé. Il est possible de tirer parti de cette puissance lors du traitement d'images histologiques. Par exemple, en couplant une grille de calcul et l'utilisation de plusieurs cartes graphiques, la solution proposée par Hartley *et al.* [HCR⁺08] permet de traiter une image de $33\,000 \times 66\,000$ pixels en seulement 4 secondes.

4.2.5 Bilan

Au cours de cette section, nous avons présenté les différentes façons d'approcher la segmentation de grandes images. Les exemples et travaux présentés concernent essentiellement le domaine médical et le traitement des images histologiques. Nous pouvons remarquer que la démarche que nous avons retenue pour la construction de pyramides descendantes tuilées combine plusieurs solutions rencontrées dans la littérature.

En effet, la première approche consiste à proposer une décomposition par blocs des images de façon à ne plus être limité par la contrainte technique d'espace mémoire disponible (Section 4.2.2) de façon similaire à la décomposition en tuiles des cartes topologiques que nous avons proposée dans le Chapitre 2. La seconde catégorie de méthodes proposées correspond à l'utilisation de structures hiérarchiques, éventuellement combinées avec une décomposition par blocs des niveaux (Section 4.2.3). De la même façon, nous avons proposé une extension hiérarchique des cartes tuilées lors de la définition de notre modèle de pyramides descendantes tuilées dans le Chapitre 3. Nous retrouvons ainsi une analogie entre les méthodes de segmentation des grandes images et les choix de construction de notre modèle.

4.3 Expérimentations sur le modèle

Dans cette section, nous présentons différentes expérimentations que nous avons conduites afin de démontrer l'efficacité de notre modèle pour la segmentation de grandes images. Les expérimentations que nous conduisons dans cette section mettent en avant les propriétés fondamentales des pyramides descendantes tuilées :

- l'opération de raffinement d'une partition par un processus de décomposition/fusion de ses régions ;
- la décomposition en tuiles d'une partition pour la représentation des grandes images ;

- la construction d’une hiérarchie de partitions par un processus descendant.

La construction des partitions repose sur la définition d’un *Oracle* (Algorithme 15, page 140) qui est constitué de deux critères distincts :

- le critère de décomposition qui indique si une région est une région d’intérêt devant être raffinée au niveau suivant (principe de focus d’attention) ;
- le critère de fusion qui indique la façon dont l’ensemble des régions résultant d’une même décomposition doivent être fusionnées (étape de fusion).

Dans la Section 4.3.1, nous détaillons la façon dont nous évaluons l’espace mémoire requis par notre modèle de pyramide descendante tuilée. Nous présentons également les différentes classes de critères que notre modèle permet de mettre en œuvre pour la construction d’une pyramide. Dans la Section 4.3.2, nous abordons la problématique de la représentation de grandes images à l’aide du modèle de carte tuilée. Dans la Section 4.3.3, nous nous intéressons à l’utilisation des pyramides descendantes tuilées pour la segmentation d’images multi-résolutions.

Nos expérimentations sont effectuées à partir d’une implantation en C++ du modèle de pyramide descendante tuilée. Elles sont conduites sur une machine personnelle équipée d’un processeur Intel E5300@2GHz et disposant de 2 Go de mémoire vive (RAM).

4.3.1 Préliminaires

Évaluation de l’espace mémoire

Une des problématiques qui a motivé la définition du modèle de pyramide descendante tuilée est la contrainte mémoire imposée par la taille des images à manipuler. Par conséquent, nous détaillons la manière dont nous évaluons l’espace mémoire requis par notre modèle.

Étant donné qu’une tuile topologique est composée d’une carte combinatoire, d’une matrice d’éléments interpixel et d’un arbre d’imbrication des régions (Section 2.2.1), nous évaluons l’espace mémoire occupé par une tuile topologique en considérant séparément ces trois modèles.

La taille de la carte combinatoire associée dépend du nombre de régions et de brins de la partition. Notons $taille(brin)$ et $taille(région)$ la taille en octets d’un brin et d’une région. Dans notre implantation, nous avons en pratique $taille(brin) = 60$ octets et $taille(région) = 80$ octets. L’espace mémoire maximal occupé par une tuile topologique est atteint lorsque

4.3. Expérimentations sur le modèle

chaque pixel est codé par une région différente, correspondant au niveau 0 de la carte topologique (Section 1.3.4). Nous pouvons ainsi borner l'occupation mémoire de la carte combinatoire, notée $\text{ram}(\text{carte})$ d'une tuile topologique par l'expression suivante :

$$\text{ram}(\text{carte}) \leq (4lh + 2(l + h)) \times \text{taille}(\text{brin}) \text{ octets}$$

où l et h désignent respectivement la largeur et la hauteur de la tuile.

La taille de la matrice des éléments interpixel associée à une tuile topologique est fixe et dépend uniquement des dimensions de la tuile. Notre modèle utilise une matrice de Khalimsky dont chaque élément est codé sur 1 octet (Section 1.2.3). La taille mémoire occupée par la matrice des éléments interpixel d'une tuile topologique de largeur l et de hauteur h est alors :

$$\text{ram}(\text{géométrie}) = (l + 1)(h + 1) \text{ octets}$$

L'espace mémoire occupé par l'arbre d'imbrication des régions, noté $\text{ram}(\text{arbre})$, correspond directement au nombre de régions présentes dans la carte. En effet, les relations d'imbrications sont codées au niveau des régions sous la forme de trois liens vers la région parente, la région fille et la région désignant la composante connexe suivante. Nous obtenons :

$$\text{ram}(\text{arbre}) \leq (lh + 1) \times \text{taille}(\text{région}) \text{ octets}$$

Notons que le modèle de carte tuilée n'entraîne pas de surcoût spécifique en mémoire et que par conséquent, l'espace mémoire d'une carte tuilée peut être estimé par la somme de l'espace mémoire occupé par les tuiles qui la composent. De plus, les relations *prédécesseur/successeur* qui définissent la hiérarchie d'une pyramide descendante tuilée sont codées au niveau des brins et des régions et sont donc comprises dans $\text{taille}(\text{brin})$ et $\text{taille}(\text{région})$. Les autres relations *prédécesseur/successeur* étant implicites, l'espace mémoire occupé par une pyramide descendante tuilée correspond également à l'espace mémoire occupé par les tuiles topologiques composant les différents niveaux.

Dans le Tableau 4.1, nous présentons l'espace mémoire maximal nécessaire à la construction du modèle de tuile topologique pour différentes tailles de tuiles.

Chaque tuile topologique code une subdivision de l'image. Nous utilisons la bibliothèque *bigtiff* afin de ne charger en mémoire que les subdivisions d'images associées aux tuiles chargées en mémoire. Les images que nous manipulons sont enregistrées au format *tiff*¹. La bibliothèque *bigtiff*² est une extension de la

1. <http://www.libtiff.org/>

2. <http://www.aperio.com/bigtiff/>

TABLE 4.1 – Occupation mémoire maximale théorique d’une tuile topologique dans notre implantation du modèle de pyramide descendante tuilée.

taille (px)	64	128	256	512	1 024	2 048
ram(carte) (Mo)	1.0	3.8	15.1	60.1	240.2	960.5
ram(géométrie) (Mo)	0.0	0.0	0.1	0.3	1.0	4.0
ram(arbre) (Mo)	0.2	1.0	3.9	15.7	62.9	251.7
total (Mo)	1.2	4.8	19.1	76.1	304.1	1 216.2

bibliothèque *libtiff* qui permet de représenter des images de taille supérieure à 4 Go en proposant un codage sur 64 bits (contre 32 bits pour la *libtiff*).

Critères de construction

Le processus de construction d’une pyramide descendante tuilée requiert la définition d’un *Oracle* permettant la construction d’un nouveau niveau à partir du niveau précédent (Section 3.5.3). Cet *Oracle* repose notamment sur la définition de deux éléments :

- un critère de décomposition permettant d’indiquer si une région doit être raffinée au niveau suivant (raffinement par focus d’attention) ;
- un critère d’homogénéité entre les régions (ou critère de segmentation), noté `même_région` et utilisé pour le raffinement des régions dont le parent est une région d’intérêt.

Ces deux critères peuvent combiner différentes propriétés du modèle. Nous pouvons ainsi utiliser des informations de nature géométrique, colorimétrique, topologique ou hiérarchique.

Les critères de nature *colorimétrique* désignent les critères calculés depuis la pyramide d’images comme par exemple la valeur moyenne des niveaux de gris ou encore la variance d’un ensemble de pixels. Comparés à un seuil défini par l’utilisateur, ces critères permettent d’évaluer simplement si deux régions doivent être fusionnées.

Nous qualifions de *géométrique* tout critère utilisant la matrice des éléments interpixel associée à la carte tuilée d’un niveau de la pyramide. Par exemple, le nombre de pixels composant le bord externe d’une région donne une première idée de sa taille. Un critère de décomposition simple consiste alors à considérer qu’une région est une région d’intérêt si sa taille en pixels est suffisamment importante. En d’autres termes, ce critère de nature géométrique permet d’ignorer les petites régions correspondant généralement au bruit de l’image. Le nombre de segments de droite qui composent une frontière constitue un

autre exemple de critère géométrique. Celui-ci permet d'évaluer la complexité d'une frontière.

Par critère *topologique*, nous faisons référence aux critères exploitant les informations de la carte combinatoire et de l'arbre d'imbrication des régions comme par exemple le nombre de régions adjacentes à une région donnée ou sa profondeur dans l'arbre d'imbrication.

Enfin, les critères de nature *hiérarchique* désignent les critères utilisant les relations *prédécesseur/successeur* de la pyramide. En effet, il peut être intéressant de comparer les propriétés d'une région (par exemple, sa variance) aux propriétés de sa région parente.

La Figure 4.3 propose deux exemples de construction d'une pyramide descendante tuilée illustrant deux critères de segmentation différents pour un même critère de décomposition. Notons $\max(r)$, $\min(r)$, $\text{moy}(r)$ et $\text{var}(r)$ le niveau de gris maximum, minimum, moyen et la variance de l'ensemble des pixels appartenant à la région r . Soit *seuil* un seuil quelconque défini par l'utilisateur. Dans les colonnes (a) et (c), le critère de segmentation utilisé fusionne deux régions adjacentes r et r' si $|\text{moy}(r) - \text{moy}(r')| < \text{seuil}$ avec *seuil* valant respectivement 100, 40 et 10 pour les niveaux $G^0/G^1/G^2$. Dans les colonnes (b) et (d), le critère de segmentation utilisé fusionne deux régions adjacentes r et r' issues d'un même parent si $|\text{var}(r \cup r') - \text{var}(\text{prédécesseur}(r))| < \text{seuil}$ avec *seuil* valant respectivement 100, 40 et 10 pour les niveaux $G^0/G^1/G^2$.

4.3.2 Segmentation par blocs avec le modèle de carte tuilée

Dans la mesure où nous avons proposé une décomposition en tuiles des différents niveaux de représentation d'une image multi-résolution, nous avons réalisé différentes expérimentations afin d'évaluer les conséquences de ces décompositions. Dans un premier temps, nous comparons le temps d'exécution et l'utilisation mémoire nécessaires à la construction de deux pyramides descendantes avec et sans décomposition en tuiles. Dans un second temps, nous évaluons l'impact du choix initial de la taille des tuiles. En effet, une taille de tuile plus réduite requiert moins d'espace mémoire pour la construction du modèle mais induit plus d'opérations de chargement/déchargement (Section 2.6).

Comparaison approches tuilées et non tuilées

Dans le Tableau 4.2, nous présentons le temps d'exécution et l'espace mémoire requis pour la construction d'une carte non tuilée associée à l'image

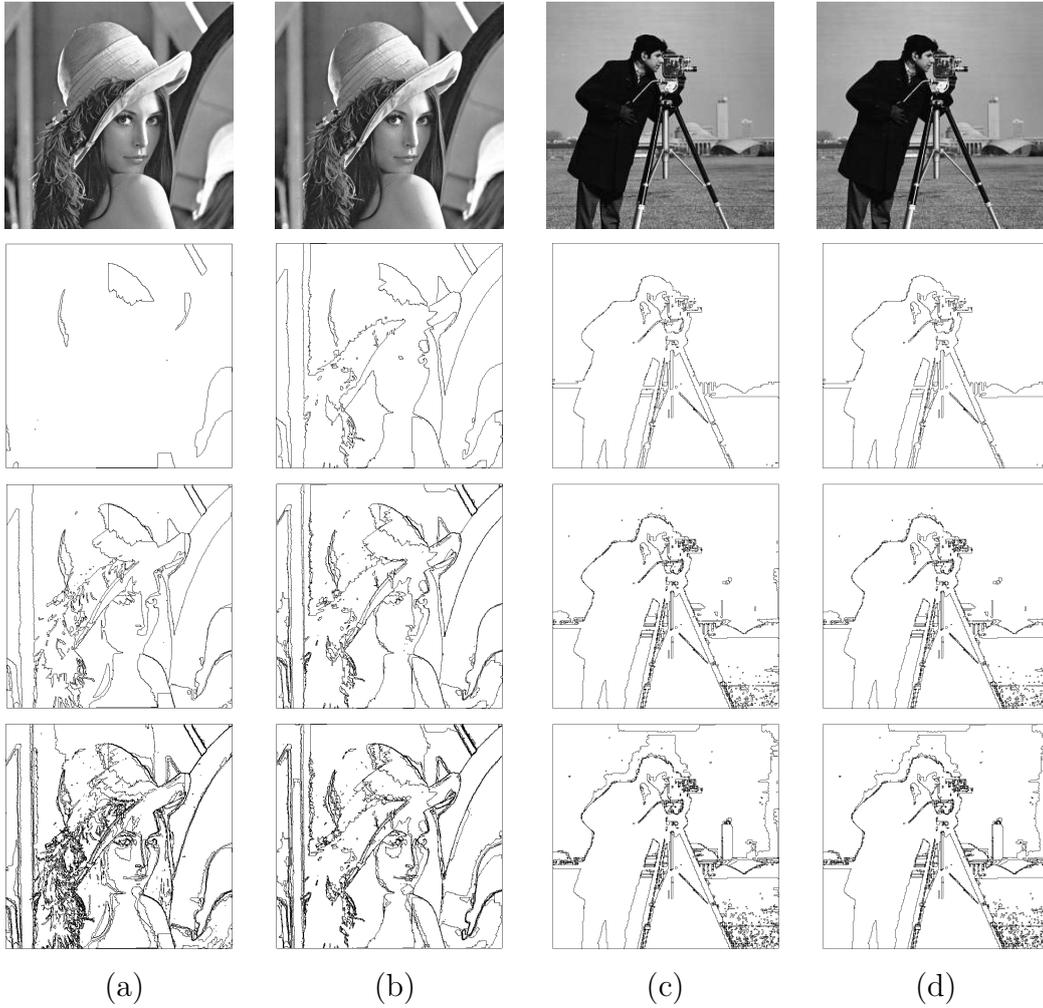


FIGURE 4.3 – Partitions résultant de la construction d’une pyramide descendante tuilée composée de trois niveaux pour deux critères simples de segmentation. Colonnes (a) et (c) : critère colorimétrique. La comparaison est effectuée sur la moyenne du niveau de gris de deux régions adjacentes. Colonnes (b) et (d) : critère hiérarchique. La comparaison est effectuée par rapport à la variance de la région parente.

TABLE 4.2 – Construction de cartes topologiques pour différentes mises à l'échelle de l'image *Lena*. L'espace mémoire nécessaire à la représentation du modèle devient rapidement trop volumineux.

taille (px)	extraction (s)	ram (Mo)	nombre de brins	nombre de régions
512	0.9	92.2	2 374	875
1 024	3.7	366.6	1 804	773
2 048	15.4	1 412.3	1 866	829
4 096		na		

Lena. En pratique, une carte non tuilée correspond à une unique tuile topologique dont les dimensions sont celles de l'image et correspond ainsi à un modèle de représentation d'image équivalent au modèle de carte topologique. L'*Oracle* utilisé pour la construction des partitions est une classification en deux classes des pixels appartenant à une même tuile. La deuxième colonne indique la durée du processus de construction. La troisième colonne indique l'espace mémoire occupé par le modèle. Les deux dernières colonnes indiquent respectivement le nombre de brins et de régions des partitions produites. Cette expérimentation met en évidence les limites du modèle de cartes non tuilées pour la représentation des grandes images car il n'est pas possible de représenter des images de taille supérieure à $2\,048 \times 2\,048$.

Dans le Tableau 4.3, nous démontrons l'intérêt du modèle de carte tuilée pour la construction de grandes partitions. En effet, la subdivision en tuiles permet de représenter n'importe quelle grande image car seul un nombre limité de tuiles est chargé en mémoire à un moment donné, ce qui se traduit par un espace mémoire nécessaire borné.

Taille et nombre de tuiles

Dans le Tableau 4.4 nous étudions l'impact du choix initial de la taille des tuiles sur la durée de construction d'une carte tuilée. Pour cette expérimentation, nous considérons une même image *Lena* de dimensions $1\,024 \times 1\,024$ pour laquelle nous construisons quatre cartes tuilées en utilisant notre algorithme d'extraction (Algorithme 4, page 89) et en faisant varier la taille d'une tuile. Cette expérimentation met en évidence la diminution de l'espace mémoire nécessaire à la construction du modèle avec la taille des tuiles. En revanche, une subdivision trop fine engendre une augmentation de la durée d'extraction principalement due à l'augmentation des accès disques effectués lors des

TABLE 4.3 – Construction de cartes tuilées pour la représentation de grandes images. Les images considérées correspondent à différentes mises à l'échelle de l'image *Lena*. La taille des tuiles est ici constante et égale à 512×512 .

taille de l'image (px)	nombre de tuiles	extraction (s)	ram (Mo)	disque (Mo)	nombre de régions
512	1	0.9 s	92	0.8	875
1 024	4	3.7 s	94	2.4	1 050
2 048	16	15.8 s	95	8.5	878
4 096	64	81.6 s	101	33	1 658
8 192	256	9 mn 43 s	99	132	7 651

opérations de chargement et de déchargement des tuiles. Notons que la taille occupée sur disque correspond à l'ensemble des fichiers correspondant aux structures décrites dans la Section 2.6 pour la représentation des tuiles topologiques.

Notre processus de construction d'une carte tuilée requiert un maximum de trois tuiles chargées en mémoire (la tuile courante et ses deux voisines adjacentes à gauche et en haut). Ainsi, pour obtenir les meilleures performances lors de la construction d'une pyramide descendante tuilée, nous choisissons un nombre réduit de tuiles afin de minimiser les accès disques et les opérations de connexion entre tuiles adjacentes, tout en nous assurant que la taille maximale occupée par quatre tuiles topologiques n'excède pas la mémoire disponible. Les résultats obtenus montrent une augmentation linéaire du temps de calcul en fonction du nombre de tuiles. En revanche, la mémoire nécessaire décroît de façon inversement proportionnelle au nombre de tuiles.

Détection des bords fictifs

La permutation δ_1 et l'involution δ_2 du modèle de carte tuilée s'appliquent sur les brins d'un ensemble de tuiles topologiques correspondant à un bord réel (Section 2.3.3). Nous avons ainsi partitionné l'ensemble des brins appartenant aux bords des tuiles en deux sous-ensembles \mathcal{B}_R^T et $\mathcal{B}_{\bar{R}}^T$ correspondant respectivement aux bords réels et aux bords fictifs. L'opération de connexion entre des tuiles adjacentes que nous avons proposée (Section 2.4.2) permet de différencier les brins de ces deux ensembles. La Figure 4.4 illustre ce partitionnement en présentant une vue comparée de l'ensemble des tuiles topologiques et de la carte tuilée correspondante. L'*Oracle* retenu pour établir ce partitionnement résulte

TABLE 4.4 – Comparaison de la durée d’extraction et de l’espace mémoire nécessaire lors de l’extraction d’une carte tuilée associée à l’image *Lena* (1024×1024) en fonction de la taille des tuiles.

tuiles		durée (s)	mémoire (Ko)	
taille (px)	nombre		ram	disque
1024	1	4.8	5 350	2 300
512	4	5.0	2 992	2 300
256	16	5.3	1 084	2 400
128	64	6.4	434	2 800
64	256	10.9	203	4 300
32	1 024	28.4	100	8 300
16	4 096	98.0	55	33 000

d’une classification en trois classes de l’image originale : $Oracle(r, r') = vrai$ si et seulement si les pixels de r et les pixels de r' appartiennent à la même classe. Notons la présence des sommets de degré 2 au niveau du bord des tuiles composant la carte tuilée. Dans cet exemple, 98 bords fictifs sont détectés entre les quatre tuiles, correspondant à 49 couples de brins reliés par l’involution β'_2 .

Raffinement des régions

Le processus de construction que nous avons proposé pour notre modèle de pyramide descendante tuilée consiste à raffiner les régions d’une partition pour établir le niveau suivant. Par conséquent, nous comparons ici deux méthodes différentes (Section 3.5.2) que nous avons implantées dans notre modèle :

- le raffinement par décomposition/fusion ;
- le raffinement par extraction.

L’expérimentation que nous proposons pour la comparaison de ces deux méthodes consiste à construire une première partition composée d’une unique région englobant toute l’image. Nous comparons alors les deux algorithmes de décomposition/fusion et d’extraction pour le raffinement de cette région. Notons que l’utilisation des mêmes critères garantit que les deux partitions obtenues sont équivalentes et contiennent donc le même nombre de brins et de régions. Le Tableau 4.5 présente les temps et l’espace mémoire requis par les deux algorithmes pour le raffinement de différentes mises à l’échelle de l’image *Lena*. Le critère de segmentation retenu considère que deux pixels



(a)



(b)



(c)

FIGURE 4.4 – Détection des bords fictifs dans une carte tuilée. (a) Classification en trois classes de l'image *Lena*; (b) Décomposition en quatre tuiles topologiques; (c) Carte tuilée correspondante. Tous les brins correspondent à un bord réel.

appartiennent à la même région s'ils ont la même couleur. La taille d'une tuile est fixée à 512×512 pixels.

À partir de ces résultats, nous constatons que l'algorithme d'extraction est plus efficace, à la fois en temps d'exécution et en mémoire consommée. Cela justifie notre choix de cet algorithme pour la construction des modèles de cartes tuilées et de pyramides descendantes tuilées. En revanche, l'opération de raffinement par décomposition/fusion reste intéressante pour le raffinement *a posteriori* des partitions. En effet, l'algorithme d'extraction utilise un balayage ligne par ligne de l'image et est donc plus adapté au raffinement de régions rectangulaires, contrairement à l'opération de décomposition/fusion.

TABLE 4.5 – Comparaison des méthodes de raffinement d’une partition appliquée pour différentes mises à l’échelle de l’image *Lena*. La partition initiale est composée d’une unique région.

taille (px)	régions	extraction		décomposition/fusion	
		durée (s)	ram (Mo)	durée (s)	ram (Mo)
1 024 × 1 024	76 976	6.0	73.5	9.2	183.3
2 048 × 2 048	271 171	17.5	75.3	32.6	171.4
4 096 × 4 096	782 587	61.0	76.0	119.0	151.7
8 192 × 8 192	1 850 589	346.3	62.7	601.0	82.9

4.3.3 Segmentation hiérarchique avec le modèle de pyramide descendante

Dans cette section, nous nous intéressons aux opérations hiérarchiques utilisées pour la construction du modèle de pyramide descendante tuilée. La méthode que nous avons proposée (Section 3.5) s’appuie sur une opération préliminaire de projection des frontières de la partition du niveau précédent (construction de G^k), puis d’une étape de raffinement par focus d’attention des régions d’intérêt (construction de G^{k+1}).

Méthodes de projection

À travers la notion de bande de projection des $(0, 1)$ -chemins (Section 3.3.1), nous avons défini l’opération de projection des frontières d’une partition en respectant la contrainte de causalité : la carte G^k et sa carte projetée G'^k sont isomorphes. Nous avons proposé deux méthodes de projection différentes : la projection directe et la projection optimisée (Section 3.5.1). Cette dernière permet notamment de tirer parti de l’information additionnelle apportée par le niveau de projection. Afin de mettre en avant les différences entre les deux méthodes, le ratio entre les niveaux doit être important. En effet, afin de garantir la préservation de la topologie entre G^k et G'^k , la géométrie d’une frontière est modifiée à l’intérieur de la bande de connexion dont l’épaisseur dépend directement du ratio : plus le ratio est grand, plus la bande de projection est « large » et plus la géométrie du chemin projeté peut être raffinée.

La Figure 4.5 permet de comparer les deux méthodes de projection. La pyramide est composée de deux niveaux. L’image I^0 du premier niveau correspond à un sous-échantillonnage de l’image *Lena* (64×64) dont une classification en deux classes donne la partition initiale (Figure 4.5(a)). L’image

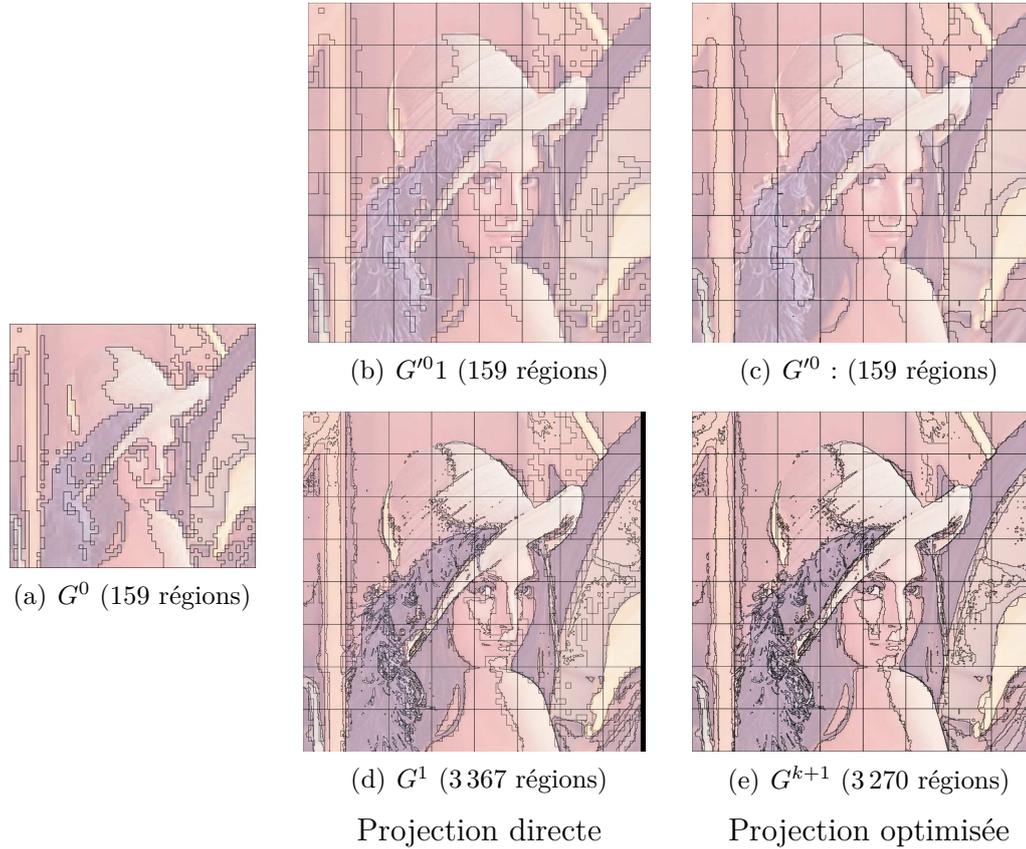


FIGURE 4.5 – Comparaison entre la projection directe et la projection optimisée dans une pyramide descendante tuilée de deux niveaux, associée à l'image *Lena* ($r_x^k = r_y^k = 8$). La taille des tuiles est fixée à 64×64 px. (a) Partition initiale; (b) Projection directe : le ratio important entre les deux niveaux fait apparaître des effets de blocs; (c) Projection optimisée : utilisation de l'information du niveau $k + 1$ pour affiner la géométrie des frontières tout en conservant la même topologie; (d) Extraction directe : raffinement par classification de l'ensemble des régions après projection directe; (e) Extraction optimisée : raffinement par classification de l'ensemble des régions après projection optimisée.

I^1 du second niveau correspond à l'image *Lena* (512×512), soit un ratio $r_x^0 = r_y^0 = 8$. La première ligne représente les opérations de projection : le critère de décomposition (focus d'attention) est forcé à *faux* pour toutes les régions. Seule la projection est donc utilisée pour la construction des deux niveaux G'^0 . Nous remarquons l'apparition d'importants effets de blocs par la méthode de projection directe (Figure 4.5(b)) qui sont atténués par la projection optimisée (Figure 4.5(c)) comme par exemple au niveau de l'épaule. La seconde ligne ajoute l'opération de raffinement aux opérations de projection en utilisant le même critère. Le raffinement après la projection directe produit une partition composée de 3 367 régions (Figure 4.5(d)) tandis que la partition résultant du raffinement après projection optimisée en comporte 3 270 (Figure 4.5(e)). Ce résultat s'explique par la nécessité de raffiner plus précisément les contours projetés grossièrement lors de la projection directe.

Focus d'attention

La notion de focus d'attention permet de raffiner uniquement les zones d'intérêt identifiées au niveau précédent. Afin d'illustrer ce procédé, nous construisons dans la Figure 4.6 deux pyramides descendantes tuilées en utilisant les mêmes critères de segmentation et les mêmes méthodes de projection. La première pyramide est construite sans focus d'attention (toutes les régions sont raffinées). Dans la seconde pyramide, nous utilisons l'étiquetage faible du premier niveau comme focus d'attention : seules les régions étiquetées en vert sont raffinées dans les niveaux G^1 et G^2 . À travers cet exemple, nous soulignons l'importance du rôle du focus d'attention dans le procédé de construction de notre modèle. Ce focus d'attention est lié à la méthode de construction descendante que nous avons adoptée et permet d'accélérer les temps de traitement et de diminuer l'espace mémoire requis par le modèle en éliminant dès le premier niveau les régions à ignorer.

4.4 Utilisation du modèle pour la segmentation d'images histologiques

Nous présentons un cas pratique d'utilisation du modèle de pyramide descendante tuilée pour l'analyse des grandes images histologiques produites par les scanners de lames entières.

Dans le cadre de cette application, les images d'histologie fournies se présentent sous la forme d'images multi-résolution composées de trois résolutions croissantes. L'analyse manuelle effectuée par les pathologistes se traduit par

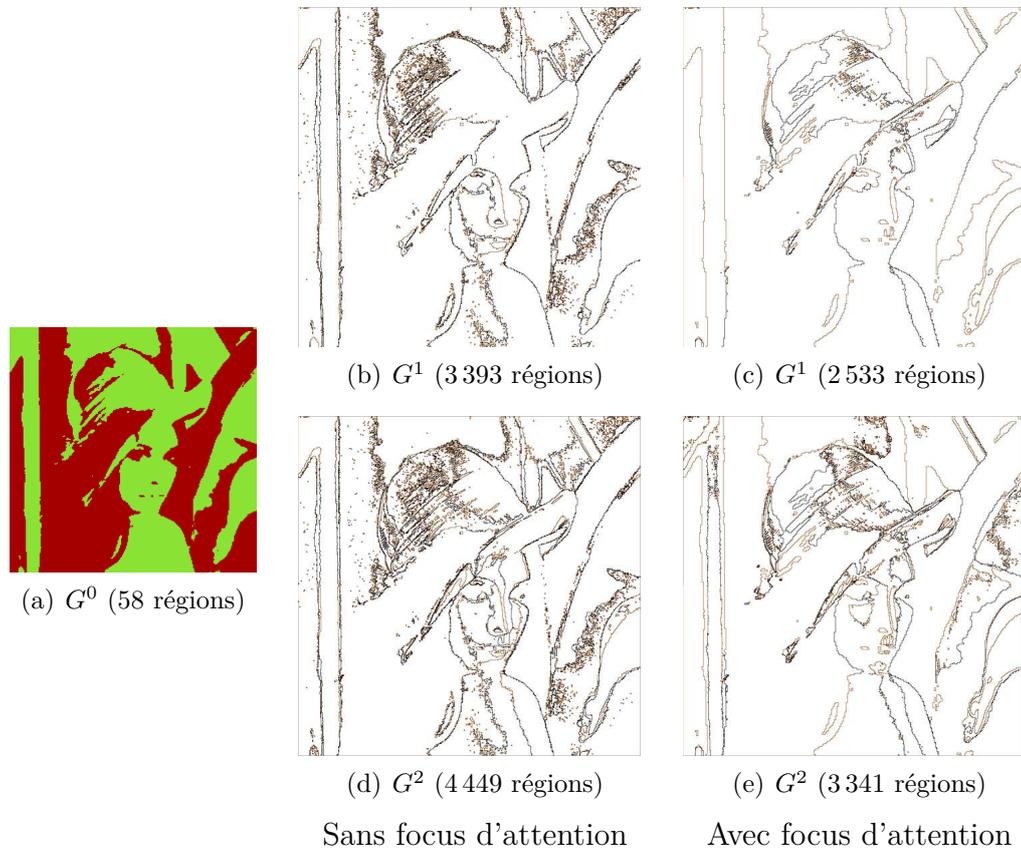


FIGURE 4.6 – Rôle du focus d'attention pour la segmentation d'une pyramide descendante tuilée. (a) Étiquetage faible d'une partition initiale. (b) et (d) Pas de focus d'attention : raffinement de l'ensemble des régions; (c) et (e) Focus d'attention : seules les régions dont le parent dans G^0 est une région étiquetée en vert sont raffinées.

un focus d'attention à partir des régions d'intérêt détectées à la plus faible résolution qui sont alors étudiées plus en détail à la résolution la plus importante. Cela permet de simplifier l'opération de dénombrement des cellules en phase de mitoses (possible uniquement à cette résolution) après avoir restreint leur localisation probable dans l'image à faible résolution. Ce dénombrement des mitoses constitue un des critères principaux à l'établissement du diagnostic précoce du cancer du sein.

Nous proposons une segmentation descendante s'appuyant sur la méthodologie proposée par Roullier *et al.* [RLTE10]. Notre méthode s'appuie sur notre modèle de pyramide descendante tuilée dans lequel chaque niveau est segmenté par une classification en deux classes que nous détaillons dans la

section Section 4.4.1. Une fois que la classification de la résolution la plus importante a été effectuée, nous pouvons extraire les mitoses, comme détaillé dans la Section 4.4.2.

4.4.1 Processus de segmentation

La coloration histochimique des prélèvements permet d'utiliser des critères colorimétriques pour identifier les composantes histologiques pertinentes. De façon générale, les zones foncées correspondent aux régions d'intérêt tandis que les zones claires peuvent être ignorées. Cette propriété permet d'utiliser un processus de classification récursif en deux classes permettant de séparer les zones claires et les zones foncées. Pour effectuer le partitionnement en régions d'un même niveau, nous utilisons l'algorithme des k -moyennes (k -means en anglais) dont une implantation est proposée par Kanungo *et al.* [KMN⁺02]. En effet, l'algorithme des k -moyennes permet d'effectuer une partition d'un ensemble en k classes, k étant défini par l'utilisateur, suivant les étapes principales suivantes :

1. la sélection aléatoire de k couleurs représentantes $\{c_1, \dots, c_k\}$ appelés *centres* des k classes $\{C_1, \dots, C_k\}$;
2. l'affectation des couleurs de l'image à la classe la plus proche au sens d'une mesure d de similarité ($\min_{1 \leq i \leq k} d(p, c_i)$, pour tout pixel p) ;
3. la mise à jour des centres des classes correspondant à la moyenne de tous les objets de la classe ($c_i = \frac{1}{|C_i|} \sum_{p \in C_i} p$) ;
4. la répétition des étapes 2 et 3 tant que la différence entre les premiers et les derniers centres d'une même classe reste supérieure à un seuil donné.

La partition associée à cette classification est composée de régions dont tous les pixels appartiennent à une même classe. À partir de cet étiquetage des régions, nous définissons le critère de décomposition de façon à ce qu'une région soit raffinée au niveau suivant si son étiquette correspond à la classe d'intérêt du niveau précédent.

Ainsi, les classifications successives permettent de différencier les composantes histologiques majeures de l'image (Figure 2, page 12) à un niveau de résolution donné. À la résolution 1, la classification permet de séparer les régions appartenant au fond de l'image des régions représentant le tissu. À la résolution 2, seules les régions identifiées comme appartenant précédemment au tissu sont raffinées, permettant d'extraire les lésions. Ces lésions sont raffinées à la résolution 3 de façon à éliminer le stroma et garder les régions cancéreuses (*cancer in situ*).

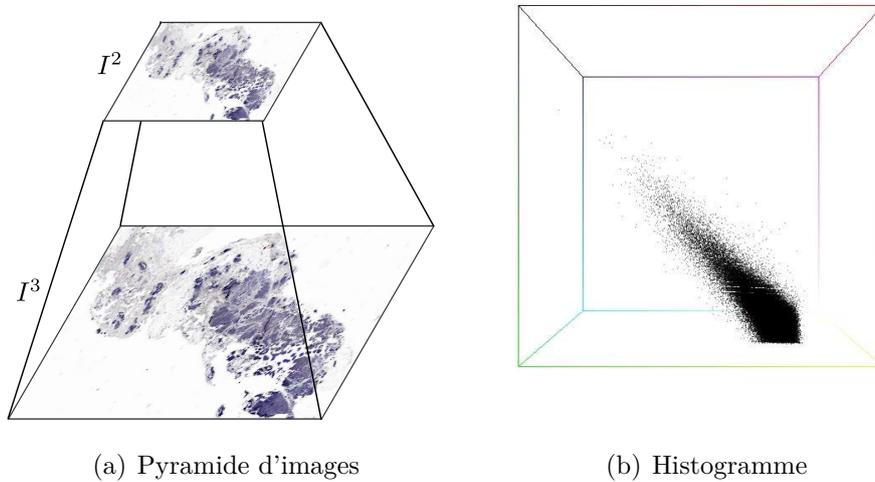


FIGURE 4.7 – Exemple d’histogramme 3D calculé sur l’ensemble des pixels de l’image I^3 dont le prédécesseur dans I^2 appartient à une région d’intérêt.

Différentes classifications sont utilisées suivant la résolution. À la résolution 1, la classification est effectuée sur l’ensemble des pixels après une étape de régularisation de l’image. Cette régularisation permet de « lisser » l’image originale et d’obtenir des régions plus homogènes. Pour les résolutions suivantes, nous effectuons la classification sur l’histogramme (Figure 4.7) des pixels de niveau k dont le prédécesseur au niveau $k - 1$ appartient à une région d’intérêt. L’occupation mémoire de l’opération de segmentation est bornée par la taille des tuiles. De plus, la segmentation obtenue ne présente pas d’effet de blocs dû à la décomposition en tuiles.

Le Tableau 4.6 présente les résultats expérimentaux obtenus lors de la segmentation de l’image histologique proposée dans la Figure 4.8(a). Les colonnes 2, 3 et 4 indiquent respectivement la dimension en pixels des différentes résolutions de l’image, le nombre de tuiles composant le pavage qui leur est associé et le nombre de régions de leur partition. La colonne 5 indique la durée de la construction de la pyramide. Les colonnes 6 et 7, correspondent à l’allocation mémoire nécessaire : nous séparons les ressources nécessaires à la construction de la pyramide descendante tuilée (modèle) des ressources nécessaires à la classification par k -moyennes (segmentation).

Ces résultats montrent la capacité de notre modèle à fournir un accès efficace aux propriétés hiérarchiques, topologiques et géométriques d’une image multi-résolution moyennant un surcoût modéré du processus de segmentation. Notons que les temps de calcul obtenus pour l’extraction des différents niveaux met en évidence la complexité linéaire suivant le nombre pixels

4.4. Utilisation du modèle pour la segmentation d'images histologiques

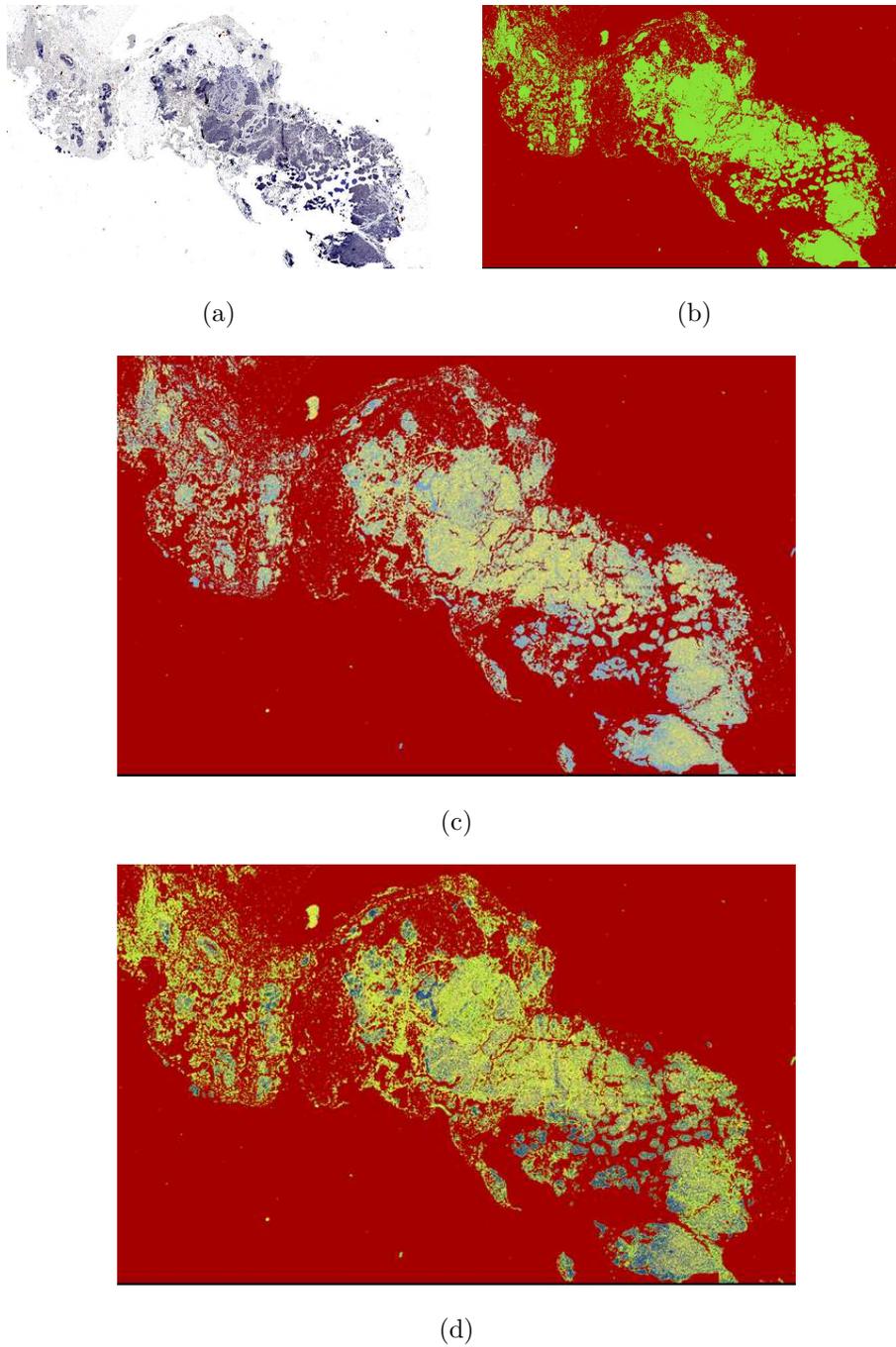


FIGURE 4.8 – Classifications successives d'une image histologique. (a) Image originale; (b) Résolution 1 : distinction entre le tissu et le fond de l'image; (c) Résolution 2 : raffinement du tissu pour extraire les lésions; (d) Résolution 3 : raffinement des lésions pour extraire le cancer *in situ*.

TABLE 4.6 – Utilisation du modèle de pyramide descendante tuilée pour la segmentation d’une image histologique.

niveau	taille de l’image (px)	nombre de tuiles	nombre de régions	durée (s)	ram (Mo)	
					modèle	segmentation
1	2 048 × 1 280	40	8 135	15.4	104	804
2	4 096 × 2 560	160	201 433	25.6	101	729
3	8 192 × 5 120	640	585 769	201.4	106	1 221

de notre méthode. Nous pouvons remarquer que ce temps de traitement augmente légèrement avec le nombre de tuiles en raison d’un nombre plus important d’accès disque nécessaires lors des opérations de chargement et de déchargement (Section 2.6).

Enfin, le modèle de carte tuilée permet de maintenir une occupation mémoire bornée d’environ 100 Mo. L’occupation mémoire est directement dépendante de la taille des tuiles (Section 4.3.1) que nous avons fixée à 256×256 pixels dans cet exemple.

4.4.2 Extraction des mitoses

Le processus de segmentation que nous avons mis en place dans notre modèle de pyramide descendante tuilée permet d’obtenir une classification de la base de la pyramide. Ce dernier niveau correspond à la résolution la plus importante de l’image histologique qui permet d’observer les structures au niveau cellulaire et donc d’identifier les cellules en phase de division cellulaire (mitoses).

La coloration histochimique appliquée au prélèvement permet de mettre en évidence les mitoses, le colorant se fixant au moment de la division cellulaire. De la même façon que Roullier *et al.* [RLTE10], nous proposons de caractériser les mitoses par leur couleur rougeâtre et donc d’appliquer un seuillage colorimétrique sur les régions d’intérêt identifiées à la base de la pyramide suivant la formule $2r - b - v$, où r , v et b désignent respectivement les trois composantes rouge, verte et bleue d’un pixel. Ce filtre permet d’extraire les régions du modèle correspondant aux mitoses de l’image. De plus, notre modèle permet de combiner simplement des critères de différentes natures. Nous pouvons par exemple rajouter une contrainte sur la taille des régions des mitoses afin d’éviter les larges zones correspondant à des dépôts de colorant. La Figure 4.9 illustre l’identification d’une mitose

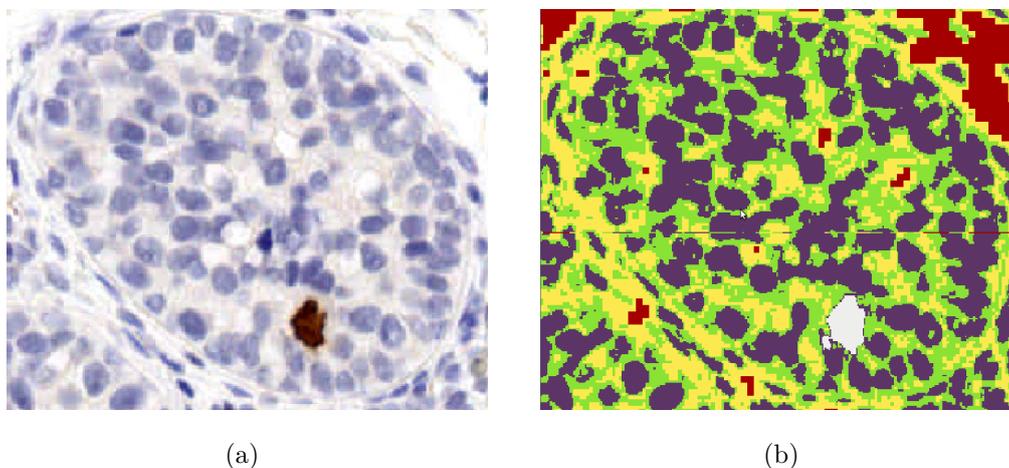


FIGURE 4.9 – Extraction des mitoses dans le dernier niveau d’une pyramide descendante tuilée. (a) Échantillon de l’image I^4 correspondant au dernier niveau : les mitoses sont colorées histochimiquement par un colorant de couleur brune; (b) Partition correspondante : les régions étiquetées en blanc correspondent aux figures de mitoses de l’image.

dans le dernier niveau de la pyramide correspondant à la Figure 4.8(a). Contrairement à Roullier *et al.* [RLTE10], la causalité de notre modèle garantit que les régions correspondant aux mitoses appartiennent aux composantes histologiques d’intérêt identifiées dans les niveaux précédents de la pyramide.

4.5 Conclusion

Dans ce chapitre, nous nous sommes intéressés à l’utilisation de notre modèle de pyramide descendante tuilée pour la segmentation d’images. En particulier, nous avons étudié dans la Section 4.2, le cas pratique de la segmentation de grandes images histologiques. Comparé au nombre de méthodes existantes pour la segmentation d’images, peu ont été proposées pour résoudre les problèmes de ce domaine. En effet, ces images représentent un grand défi pour le traitement et l’analyse d’images en raison de leur taille, conduisant à un volume de données difficile à manipuler à l’aide d’un modèle. De plus, l’analyse descendante utilisée par les pathologistes implique un modèle hiérarchique capable de représenter l’aspect multi-échelle et multi-résolution de ces images.

Dans la Section 4.3, nous avons étudié la problématique de la représentation de grandes images par notre modèle. Nous avons abordé le problème sous deux angles d’approche. Premièrement, la décomposition par blocs (ou tuiles) de

notre modèle de carte tuilée permet la représentation de grandes images en conservant un espace mémoire borné. Nous avons étudié à travers différentes expérimentations les possibilités qu'offre le modèle de carte tuilée pour la représentation de grandes images. Deuxièmement, nous avons utilisé la construction hiérarchique des pyramides tuilées pour la segmentation des images d'histologie. Dans ce type d'application, certaines informations sont disponibles uniquement à une échelle donnée. Par exemple, les différentes phases de mitoses ne sont identifiables qu'à la résolution la plus élevée tandis que les structures tissulaires ne sont observables qu'à faible résolution.

Enfin, dans la Section 4.4, nous avons appliqué notre modèle au cas spécifique de la segmentation des images d'histologie. En nous appuyant sur un processus de segmentation existant et dédié à ce type d'images, nous avons montré les avantages qu'offre le modèle de pyramide descendante tuilée pour ce genre d'application. En effet, de par sa conception, notre modèle permet de représenter les relations topologiques, hiérarchiques et géométriques dans une structure pyramidale irrégulière. De plus, il code naturellement la dualité des relations hiérarchiques et multi-résolution inhérentes aux grandes images.

Conclusion et Perspectives

Au cours de cette thèse, nous avons défini un modèle topologique et hiérarchique adapté à la segmentation de grandes images multi-résolution. Nos travaux sont fondés sur les modèles existants de *carte topologique* et de *pyramide combinatoire*.

Notre contribution au modèle de carte topologique permet l'utilisation de ce modèle pour la représentation de grandes images. En effet, l'espace mémoire requis par une carte topologique peut devenir trop important dans le cas d'images de grande taille. C'est pourquoi nous avons proposé une approche dite *par blocs* pour décomposer une carte topologique en sous-ensembles de plus petite taille appelés *tuiles topologiques*. Cette décomposition s'effectuant de manière géométrique suivant un pavage régulier associé à l'image, nous avons proposé une méthode pour distinguer les bords *fictifs* (correspondants aux frontières introduites par le pavage) des bords *réels* (lorsque la frontière du pavage coïncide avec la frontière d'une région). De cette façon, nous avons défini une carte topologique globale composée d'un ensemble de tuiles topologiques. La définition de nouveaux opérateurs permet d'abstraire la notion de tuile topologique et de la rendre ainsi transparente d'un point de vue utilisateur : les tuiles sont chargées en mémoire ou enregistrées sur disque dur suivant les traitements effectués. Les algorithmes de construction que nous avons proposés étant incrémentaux, l'espace mémoire requis pour la construction d'une carte tuilée reste borné. Par conséquent, le modèle de carte tuilée supprime la contrainte technique d'espace mémoire nécessaire à la représentation des grandes images.

Ces grandes images représentent intrinsèquement une information hiérarchique. Par exemple, dans le cas des images multi-résolution d'histologie, une structure tissulaire (observable à faible résolution) peut se concevoir comme la réunion des structures cellulaires (observables à forte résolution) qui la composent. Ces images présentent ainsi des structures multi-résolution et multi-échelle. Afin de représenter cette information hiérarchique, nous avons proposé une extension de notre modèle de carte tuilée : la *pyramide descendante tuilée*. Nous avons introduit des relations hiérarchiques entre les différents

objets d'une carte tuilée et proposé un processus de construction garantissant la construction d'une structure de données causale. L'originalité de notre approche vient de sa méthode de construction dite *descendante* par opposition aux méthodes *ascendantes* existantes.

En effet, dans le cadre des structures hiérarchiques régulières, les deux types de méthodes sont employés : par exemple, les quadrees constituent un modèle descendant et les pyramides régulières sont un modèle ascendant. En revanche, concernant les structures hiérarchiques irrégulières, il n'existe aucun modèle à notre connaissance adoptant un processus de construction descendant. Notre construction descendante d'une pyramide irrégulière (avec ou sans tuiles) constitue ainsi une approche originale à la définition de modèles hiérarchiques irréguliers.

Dans un processus de construction descendant, la partition initiale correspond à la partition la moins détaillée de l'image et code généralement la résolution la plus faible. Les niveaux suivants de la pyramide sont alors construits par projection puis raffinement du niveau précédent suivant le principe de *focus d'attention* : seules les régions d'intérêt identifiées au niveau précédent sont raffinées. Dans le cas des images d'histologie, ce procédé reprend l'analyse manuelle effectuée par les pathologistes. De plus, notre approche permet d'optimiser le temps de traitement et de minimiser l'espace mémoire nécessaire. En effet, seules les régions d'intérêt de la pyramide sont raffinées ce qui permet d'éviter l'étape coûteuse de représentation de partitions détaillées des premiers niveaux d'une pyramide ascendante.

Les travaux que nous avons conduits au cours de cette thèse peuvent être poursuivis de différentes manières. Nous présentons ici quelques unes des perspectives qui nous paraissent intéressantes.

Tout d'abord, les travaux que nous avons menés appartiennent à la fois au domaine de la modélisation géométrique et au domaine de la segmentation d'images. Nous avons ainsi défini un nouveau modèle capable de conduire une segmentation exploitant à la fois des critères topologiques, géométriques et hiérarchiques dans le cas des grandes images multi-résolution. En ce qui concerne l'aspect segmentation, nous avons focalisé nos travaux sur le cas particulier des images d'histologie. Nous souhaiterions étudier davantage l'aspect segmentation en proposant l'intégration de nouvelles méthodes en histologie ou pour d'autres domaines d'application. Aussi, une des conséquences de l'approche par blocs que nous avons retenue lors de la définition de notre modèle de carte tuilée se traduit parfois par l'apparition d'effets de blocs dans les résultats de segmentation obtenus.

Du point de vue du modèle, nous souhaitons approfondir la définition du modèle de carte tuilée en proposant une structure globale pour la représentation

des relations d'imbrication entre les régions d'une carte tuilée. En effet, le modèle de carte tuilée utilise actuellement un arbre d'imbrication des régions local à chaque tuile. Ces arbres ne représentent donc pas les relations globales d'imbrication entre les régions (potentiellement réparties sur plusieurs tuiles) de la carte tuilée.

Nous souhaitons intégrer dans un avenir proche les travaux effectués sur les partitions déformables dans notre modèle. Les algorithmes développés par [dL09, DDL09] permettent en effet d'envisager une déformation du contour des régions tout en préservant sa topologie. Ces travaux étendent la problématique que nous avons abordée lors de la projection des frontières entre deux niveaux. La méthode que nous avons proposée pour optimiser l'étape de projection consiste à autoriser un déplacement des frontières des régions dans une bande du niveau suivant dont la largeur correspond au ratio entre les deux niveaux considérés. Le fait de restreindre les déplacements dans cette bande définit une opération réversible qui permet de conserver la topologie et la causalité entre un niveau et son projeté. Notre méthode propose donc un surcoût de calcul modéré pour améliorer l'étape de projection. Les modèles de partitions déformables engendrant généralement des calculs coûteux, nous envisageons pour le moment de restreindre leur utilisation à certaines régions d'intérêt. En particulier, dans le cas des images d'histologie, nous envisageons de raffiner uniquement les petites régions du dernier niveau de la pyramide correspondant aux mitoses. L'intérêt d'intégrer un modèle de partitions déformables à notre processus de construction serait alors d'améliorer la précision du contour géométrique de ces régions.

Un autre aspect que nous souhaitons approfondir concerne la définition d'opérations de modifications sur le modèle. En effet, dans cette thèse, nous nous sommes focalisés sur la définition du modèle ainsi que sur sa construction. Cependant, des modifications *a posteriori* d'une partition peuvent être envisagées. La définition de telles opérations entraîne des problèmes de connexions entre les relations hiérarchiques (*relinking* en anglais) [Nac95], notamment pour préserver la propriété de causalité du modèle mais permettrait d'utiliser interactivement le modèle après sa construction. En ce sens, les travaux conduits par [GK00, GHK11] dans le cadre des pyramides combinatoires nous semblent très intéressants à adapter dans notre modèle.

Enfin, une extension en dimension supérieure du modèle de carte tuilée nous paraît tout à fait pertinente. En effet, les modèles de cartes topologiques sont utilisés pour la représentation d'image 3D [Dup09, Dam10] mais le volume de données correspondant à ces images entraîne des difficultés de représentation en mémoire, de la même façon que dans le cadre des grandes images 2D. En appliquant la notion de tuile topologique à un sous-volume de l'image,

il serait envisageable de proposer une extension du modèle de carte tuilée pour la représentation d'images 3D. Comme dans le cas des grandes images, l'avantage immédiat serait la garantie de la faisabilité de la construction du modèle, indépendamment de la taille de l'image.

Références Bibliographiques

- [AFG99] Ehoud Ahronovitz, Christophe Fiorio, and Sylvain Glaize. Topological operators on the topological graph of frontiers. In *Discrete Geometry for Computer Imagery*, number 1568 in Lecture Notes in Computer Science, pages 207–217, Marne-la-Vallée, France, 1999.
- [Ale37] P. Alexandroff. Diskrete räume. *Mat. Sbornik*, 2 :501–518, 1937.
- [AUJ⁺08] Coralie Apfeldorfer, Kristina Ulrich, Gareth Jones, David Goodwin, Susie Collins, Emanuel Schenck, and Virgile Richard. Object orientated automated image analysis : quantitative and qualitative estimation of inflammation in mouse lung. *Diagn Pathol*, 3 Suppl 1, 2008.
- [BA83] Peter J. Burt and Edward H. Adelson. A multiresolution spline with application to image mosaics. *ACM Trans. Graph.*, 2(4) :217–236, 1983.
- [BCR90] Michel Bister, Jan Cornelis, and Azriel Rosenfeld. A critical view of pyramid segmentation algorithms. *Pattern Recognition Letters*, 11(9) :605–617, 1990.
- [BD99] Achille Braquelaire and Jean-Philippe Domenger. Representation of segmented images with discrete geometric maps. *Image and Vision Computing*, 17(10) :715–735, 1999.
- [BDM03] Luc Brun, Jean-Philippe Domenger, and Myriam Mokhtari. Incremental modifications of segmented image defined by discrete maps. *Journal of Visual Communication and Image Representation*, 14(3) :251–290, 2003.
- [BK00] Luc Brun and Walter G. Kropatsch. The construction of pyramids with combinatorial maps. Technical Report 63, Institute of Computer Aided Design, Vienna University of Technology, lstr. 3/1832,A-1040 Vienna Austria, June 2000.
- [BK03a] Luc Brun and Walter G. Kropatsch. Combinatorial pyramids. In *ICIP (2)*, pages 33–36, 2003.

- [BK03b] Luc Brun and Walter G. Kropatsch. Implicit encoding of combinatorial pyramids. In Ondrej Drbohlav, editor, *Proceedings of the Computer Vision Winter Workshop*, pages 49–54, Valtice, Czech Republic, February 2003.
- [BMD03] Luc Brun, Myriam Mokhtari, and Jean-Philippe Domenger. Incremental modifications on segmented image defined by discrete maps. *Journal of Visual Communication and Image Representation*, 14 :251–290, 2003.
- [BRP09] Srečko Brlek, Christophe Reutenauer, and Xavier Provençal, editors. *Discrete Geometry for Computer Imagery, 15th IAPR International Conference, DGCI 2009, Montréal, Canada, September 30 - October 2, 2009. Proceedings*, volume 5810 of *Lecture Notes in Computer Science*. Springer, 2009.
- [Bru96] Luc Brun. *Segmentation d'images couleur à base topologique*. Thèse de doctorat, Université Bordeaux I, 351 cours de la Libération, 33405 Talence, December 1996.
- [CK93] Dmitry Chetverikov and Walter G. Kropatsch, editors. *Computer Analysis of Images and Patterns, 5th International Conference, CAIP'93, Budapest, Hungary, September 13-15, 1993, Proceedings*, volume 719 of *Lecture Notes in Computer Science*. Springer, 1993.
- [CLR95] Luigi Cinque, Stefano Levialdi, and Azriel Rosenfeld. Fast pyramidal algorithms for image thresholding. *Pattern Recognition*, 28(6) :901–906, 1995.
- [CLRS90] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. MIT Press, 1990.
- [Cor75] Robert Cori. Un code pour les graphes planaires et ses applications. In *Astérisque*, volume 27. Soc. Math. de France, Paris, France, 1975.
- [CPR03] David J. Crisp, Peter G. Perry, and Nicholas J. Redding. Fast segmentation of large images. In Michael J. Oudshoorn, editor, *ACSC*, volume 16 of *CRPIT*, pages 87–93. Australian Computer Society, February 2003.
- [Dam01] Guillaume Damiand. *Définition et étude d'un modèle topologique minimal de représentation d'images 2d et 3d*. Thèse de doctorat, Université Montpellier II, Décembre 2001.
- [Dam10] Guillaume Damiand. *Contributions aux Cartes Combinatoires et Cartes Généralisées : Simplification, Modèles, Invariants Topologiques et Applications*. Habilitation à diriger des recherches, Université Claude Bernard Lyon 1, Septembre 2010.

-
- [DBF04] Guillaume Damiand, Yves Bertrand, and Christophe Fiorio. Topological model for two-dimensional image representation : definition and optimal extraction algorithm. *Computer Vision and Image Understanding*, 93(2) :111–154, February 2004.
- [DdJ⁺09] Guillaume Damiand, Colin de la Higuera, Jean-Christophe Janodet, Émilie Samuel, and Christine Solnon. A polynomial algorithm for submap isomorphism. In Andrea Torsello, Francisco Escolano, and Luc Brun, editors, *GbRPR*, volume 5534 of *Lecture Notes in Computer Science*, pages 102–112. Springer, May 2009.
- [DDL09] Alexandre Dupas, Guillaume Damiand, and Jacques-Olivier Lachaud. Multi-label simple points definition for 3d images digital deformable model. In Brlek et al. [BRP09], pages 156–167.
- [Dij59] Edsger W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1 :269–271, 1959.
- [DL03] Guillaume Damiand and Pascal Lienhardt. Removal and contraction for n-dimensional generalized maps. In *Discrete Geometry for Computer Imagery*, number 2886 in *Lecture Notes in Computer Science*, pages 408–419, Naples, Italy, November 2003. Springer Berlin / Heidelberg.
- [dL09] François de Vieilleville and Jacques-Olivier Lachaud. Digital deformable model simulating active contours. In Brlek et al. [BRP09], pages 203–216.
- [Dom92] Jean-Philippe 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.
- [DPLH08] Anne Dorandeu, Benoît Plancoulaine, Georges Léonetti, and Paulette Herlin. Apport potentiel des lames virtuelles et de l'analyse d'images automatique dans l'estimation de la vascularisation des sutures crâniennes. *Annales de Pathologie*, 28(4) :342–346, 2008.
- [DRS80] Charles R. Dyer, Azriel Rosenfeld, and Hanan Samet. Region representation : boundary codes from quadtree. *Communications of the ACM*, 23(3) :171–179, March 1980.
- [Dup09] Alexandre Dupas. *Opérations et Algorithmes pour la Segmentation Topologique d'Images 3D*. PhD thesis, Université de Poitiers, Novembre 2009. written in french.
- [Edm60] Jack Edmonds. A combinatorial representation for polyhedral surfaces. *Notices of the American Mathematical Society*, 7, 1960.

- [ELMH05] Nicolas Elie, Alexandre Labiche, Jean-Jacques Michels, and Paulette Herlin. Control of low resolution scanning of avarian tumor stromal compartment. 24(2) :85–93, 2005.
- [FB74] Raphael A. Finkel and Jon Louis Bentley. Quad trees : A data structure for retrieval on composite keys. *Acta Inf.*, 4 :1–9, 1974.
- [FH04] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2) :167–181, 2004.
- [Fio95] Christophe Fiorio. *Approche interpixel en analyse d'images : une topologie et des algorithmes de segmentation*. Thèse de doctorat, Université Montpellier 2, Novembre 1995.
- [Fio96] Christophe Fiorio. A topologically consistent representation for image analysis : the frontiers topological graph. In *Discrete Geometry for Computer Imagery*, number 1176 in Lecture Notes in Computer Science, pages 151–162, Lyon, France, November 1996.
- [FKN80] Henry Fuchs, Zvi M. Kedem, and Bruce F. Naylor. On visible surface generation by a priori tree structures. In *Proceedings of the 7th annual conference on Computer graphics and interactive techniques*, pages 124–133. ACM New York, NY, USA, 1980.
- [FMPH05] Ronan Françoise, Jean-Jacques Michels, Benoît Plancouline, and Paulette Herlin. Optimal resolution for automatic quantification of blood vessels on digitized images of the whole cancer section. 24(1) :59–67, 2005.
- [Fra91] Jean Françon. Topologie de Khalimsky et Kovalevsky et algorithmique graphique. Rapport de recherche 91-10, Université Louis-Pasteur, Centre de Recherche en Informatique, Strasbourg, France, 1991.
- [FV92] Francisco Ferri and Enrique Vidal. Colour image segmentation and labeling through multiedit-condensing. *Pattern Recogn. Lett.*, 13(8) :561–568, 1992.
- [FVT⁺08] Levente Fiscor, Viktor Sebestyén Varga, Attila Tagscherer, Zsolt Tulassay, and Bela Molnar. Automated classification of inflammation in colon histological sections based on digital microscopy and advanced image analysis. 73(3) :230–237, March 2008.
- [GCL06] Laurent Guigues, Jean-Pierre Cocquerez, and Hervé Le Men. Scale-sets image analysis. *International Journal of Computer Vision*, 68(3) :289–317, 2006.

-
- [GDL05a] Carine Grasset-Simon, Guillaume Damiand, and Pascal Lienhardt. Pyramids of n-dimensional generalized maps. In *Proceedings of 5th IAPR-TC15 Workshop on Graph-based Representations in Pattern Recognition*, volume 3434 of *Lecture Notes in Computer Science*, pages 142–152, Poitiers, France, Avril 2005.
- [GDL05b] Carine Grasset-Simon, Guillaume Damiand, and Pascal Lienhardt. Receptive fields for generalized map pyramids : The notion of generalized orbit. In Eric Andres, Guillaume Damiand, and Pascal Lienhardt, editors, *DGCI*, volume 3429 of *Lecture Notes in Computer Science*, pages 56–67. Springer, 2005.
- [GDL06] Carine Grasset-Simon, Guillaume Damiand, and Pascal Lienhardt. N-d generalized map pyramids : Definition, representations and basic operations. *Pattern Recognition*, 39(4) :527–538, 2006.
- [GHK11] Michael Gerstmayer, Yll Haxhimusa, and Walter G. Kropatsch. Hierarchical interactive image segmentation using irregular pyramids. In Xiaoyi Jiang, Miquel Ferrer, and Andrea Torsello, editors, *GbRPR*, volume 6658 of *Lecture Notes in Computer Science*, pages 245–254. Springer, May 2011.
- [GK00] Roland Glantz and Walter G. Kropatsch. Guided relinking of graph pyramids. In Francesc J. Ferri, José Manuel I nesta Quereda, Adnan Amin, and Pavel Pudil, editors, *SSPR/SPR*, volume 1876 of *Lecture Notes in Computer Science*, pages 367–376. Springer, August 2000.
- [HCR+08] Tim Hartley, Umit Catalyurek, Antonio Ruiz, Francisco Igual, Rafael Mayo, and Manuel Ujaldon. Biomedical image analysis on a cooperative cluster of gpus and multicores. In *ICS '08 : Proceedings of the 22nd annual international conference on Supercomputing*, pages 15–25, New York, NY, USA, 2008. ACM.
- [HGK03] Yll Haxhimusa, Roland Glantz, and Walter G. Kropatsch. Constructing stochastic pyramids by mides - maximal independent directed edge set. In Edwin R. Hancock and Mario Vento, editors, *GbRPR*, volume 2726 of *Lecture Notes in Computer Science*, pages 24–34. Springer, 2003.
- [HGS+02] Yll Haxhimusa, Roland Glantz, Maamar Saib, Georg Langs, and Walter G. Kropatsch. Reduction factors of graph pyramids. In H. Wildenauer and W. Kropatsch, editors, *Proceedings of 7th Computer Vision Winter Workshop CVWW'02*, pages 29–38, February 2002.
- [HLSP06] Paulette Herlin, Bruno Lay, Nicolas Signolle, and Benoît Plancou-laine. Automatic estimation of tumor vascularization using virtual

- slides. In *VIIIth European Conference on Telepathology and 2nd International Congress on Virtual Microscopy*, 2006.
- [HP74] Steven L. Horowitz and Theodosios Pavlidis. Picture segmentation by a directed split and merge procedure. In *ICPR*, pages 424–433, 1974.
- [HP76] Steven L. Horowitz and Theodosios Pavlidis. Picture segmentation by a tree traversal algorithm. *J. ACM*, 23(2) :368–388, 1976.
- [HT84] Peri N. Hartman and Steven L. Tanimoto. An hexagonal pyramid data structure for image processing. *IEEE Transactions on Systems, Man, and Cybernetics*, 14(2) :247–256, July 1984.
- [HZL05] Yinghua He, Bo Zhang, and Jianmin Li. A new multiresolution classification model based on partitioning of feature space. In Xiaohua Hu, Qing Liu, Andrzej Skowron, Tsau Young Lin, Ronald R. Yager, and Bo Zhang, editors, *GrC*, pages 462–467. IEEE, July 2005.
- [Jac70] A. Jacques. Constellations et graphes topologiques. In *Combinatorial Theory and Applications*, volume 2, pages 657–673, 1970.
- [JM92] Jean-Michel Jolion and Annick Montanvert. The adaptative pyramid : A framework for 2d image analysis. *CVGIP*, 55(3) :339–348, May 1992.
- [Jol01] Jean-Michel Jolion. Data driven decimation of graphs. In J.M. Jolion, W. Kropatsch, and M. Vento, editors, *Proceedings of 3rd IAPR-TC15 Workshop on Graph based Representation in Pattern Recognition*, pages 105–114, May 2001.
- [JR94] Jean-Michel Jolion and Azriel Rosenfeld. *A Pyramid Framework for Early Vision : Multiresolutional Computer Vision*. Kluwer Academic Publishers, Norwell, MA, USA, 1994.
- [KKM90] Efim Khalimsky, Ralph Kopperman, and Paul R. Meyer. Boundaries in digital planes. *Journal of Applied Mathematics and Stochastic Analysis*, 3(1) :27–55, 1990.
- [KKM91] T. Yung Kong, Ralph Kopperman, and Paul R. Meyer. A topological approach to digital topology. *American Mathematical Monthly*, 98(10) :901–917, 1991.
- [KM95] Walter G. Kropatsch and Herwig Macho. Finding the structure of connected components using dual irregular pyramids. In *Discrete Geometry for Computer Imagery*, pages 147–158, September 1995.

-
- [KMN⁺02] Tapas Kanungo, David M. Mount, Nathan S. Netanyahu, Christine D. Piatko, Ruth Silverman, and Angela Y. Wu. An efficient k-means clustering algorithm : Analysis and implementation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(7) :881–892, 2002.
- [Koe84] Jan Koenderink. The structure of images. *Biological Cybernetics*, 50 :363–370, 1984.
- [Kot02] Ullrich Kothe. Xpmaps and topological segmentation - a unified approach to finite topologies in the plane. In Achille Braquelaire, Jacques-Olivier Lachaud, and Anne Vialard, editors, *Proceedings of 10th International Conference on Discrete Geometry for Computer Imagery (DGCI 2002)*, volume 2301 of *Lecture Notes in Computer Science*, pages 327–350, Bordeaux, France, April 2002. Springer Berlin / Heidelberg.
- [Kov89] Vladimir A. Kovalevsky. Finite topology as applied to image analysis. *Computer Vision, Graphics, and Image Processing*, 46 :141–161, 1989.
- [Kov00] Vladimir A. Kovalevsky. Algorithms and data structures for computer topology. In *Digital and Image Geometry*, pages 38–58, 2000.
- [Kro95] Walter G. Kropatsch. Building irregular pyramids by dual-graph contraction. *Vision, Image and Signal Processing*, 142(6) :366–374, December 1995.
- [Kro97] Walter G. Kropatsch. From equivalent weighting functions to equivalent contraction kernels. In *Dimitrov (Eds.), Digital Image Processing and Computer Graphics : Applications in Humanities and Natural Sciences*, pages 310–320, 1997.
- [KSS⁺07] Jun Kong, Olcay Sertel, Hiroyuki Shimada, Kim L. Boyer, Joel H. Saltz, and Metin N. Gurcan. Computer-aided grading of neuroblastic differentiation : Multi-resolution and multi-classifier approach. In *ICIP (5)*, pages 525–528. IEEE, 2007.
- [Lie89] Pascal Lienhardt. Subdivision of n-dimensional spaces and n-dimensional generalized maps. In *5th Annual ACM Symposium on Computational Geometry*, pages 228–236, Saarbrücken, Germany, 1989.
- [Lie91] Pascal Lienhardt. Topological models for boundary representation : a comparison with n-dimensional generalized maps. *Computer-Aided Design*, 23(1) :59–82, 1991.

- [Lie94] Pascal Lienhardt. N-dimensional generalized combinatorial maps and cellular quasi-manifolds. *International Journal of Computational Geometry and Applications*, 4(3) :275–324, 1994.
- [MAM04] Jocelyn Marchadier, Didier Arquès, and Sylvain Michelin. Thinning grayscale well-composed images. *Pattern Recognition Letters*, 25(5) :581–590, 2004.
- [Mee89] Peter Meer. Stochastic image pyramids. *Computer Vision, Graphics, and Image Processing*, 45(3) :269–294, 1989.
- [MK05] Hans Meine and Ullrich Köthe. The geomap : A unified representation for topology and geometry. In M. Vento Luc Brun, editor, *5th IAPR International Workshop, GbRPR 2005*, volume 3434 of *Lecture Notes in Computer Science*, pages 132–141, Poitiers, France, April 2005. Springer Berlin / Heidelberg.
- [MK06] Hans Meine and Ullrich Köthe. A new sub-pixel map for image analysis. In Ralf Reulke, Ulrich Eckardt, Boris Flach, Uwe Knauer, and Konrad Polthier, editors, *IWCIA*, volume 4040 of *Lecture Notes in Computer Science*, pages 116–130. Springer, 2006.
- [MMR91] Annick Montanvert, Peter Meer, and Azriel Rosenfeld. Hierarchical image analysis using irregular tessellations. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13(4) :307–316, 1991.
- [MPR⁺08] Kishore Mosaliganti, Tony Pan, Randall Ridgway, Richard Sharp, Lee Cooper, Alex Gulacy, Ashish Sharma, M. Okan Irfanoglu, Raghu Machiraju, Tahsin M. Kurç, Alain de Bruin, Pamela Wenzel, Gustavo Leone, Joel H. Saltz, and Kun Huang. An imaging workflow for characterizing phenotypical change in large histological mouse model datasets. *Journal of Biomedical Informatics*, 41(6) :863–873, 2008.
- [MXFS07] Mutlu Mete, Xiaowei Xu, Chun Y. Fan, and Gal Shafirstein. Automatic delineation of malignancy in histopathological head and neck slides. *BMC Bioinformatics*, 8(Suppl 7), 2007.
- [Nac95] Peter F. M. Nacken. Image segmentation by connectivity preserving relinking in hierarchical graph structures. *Pattern Recognition*, 28(6) :907–920, 1995.
- [Nic95] C. J. Nicol. A systolic approach for real time connected component labeling. *Computer Vision and Image Understanding*, 61(1) :17–31, 1995.
- [OCD95] Lori A. Overturf, Mary L. Comer, and Edward J. Delp. Color image coding using morphological pyramid decomposition. *IEEE Transactions on Image Processing*, 4(2) :177–185, 1995.

-
- [Ols06] Allen H. Olson. Image analysis using the aperio scanscope. Technical report, Aperio Technologies, 2006.
- [RLTE10] Vincent Roullier, Olivier Lézoray, Vinh-Thong Ta, and Abderrahim Elmoataz. Mitosis extraction in breast-cancer histopathological whole slide images. In *ISVC*, 2010.
- [Ros74] Azriel Rosenfeld. Adjacency in digital pictures. *Information and Control*, 26(1) :24–33, 1974.
- [Rou08] Vincent Roullier. *Classification floue et modélisation IRM : application à la quantification de la graisse pour une évaluation optimale des risques pathologiques associés à l’obésité*. PhD thesis, Université d’Angers, Juin 2008. written in french.
- [RS88] Azriel Rosenfeld and Allen C. Sher. Detection and delineation of compact objects using intensity pyramids. *Pattern Recognition*, 21(2) :147–151, 1988.
- [RS98] Azriel Rosenfeld and Chiao-Yung Sher. Detecting image primitives using feature pyramids. *Inf. Sci.*, 107(1-4) :127–147, 1998.
- [Sam80] Hanan Samet. Region representation : quadtrees from boundary codes. *Communications of the ACM*, 23 :163–170, 1980.
- [Sam84] Hanan Samet. The quadtree and related hierarchical data structures. *Computing surveys*, 16(2) :187–260, 1984.
- [SC83] Minsoo Suk and Soon Myoung Chung. A new image segmentation technique based on partition mode test. *Pattern Recognition*, 16(5) :469–480, 1983.
- [Ser06] Jean Serra. A lattice approach to image segmentation. *Journal of Mathematical Imaging and Vision*, 24(1) :83–130, 2006.
- [SKS⁺09] Olcay Sertel, Jun Kong, Hiroyuki Shimada, Ümit V. Çatalyürek, Joel H. Saltz, and Metin N. Gurcan. Computer-aided prognosis of neuroblastoma on whole-slide images : Classification of stromal development. *Pattern Recognition*, 42(6) :1093–1103, 2009.
- [SPHR08] Nicolas Signolle, Benoît Plancoulaine, Paulette Herlin, and Marinette Revenu. Texture-based multiscale segmentation : Application to stromal compartment characterization on ovarian carcinoma virtual slides. In Abderrahim Elmoataz, Olivier Lézoray, Fathallah Nouboud, and Dris Mammass, editors, *Image and Signal Processing*, volume 5099 of *Lecture Notes in Computer Science*, pages 173–182. Springer Berlin / Heidelberg, 2008.
- [SRPH10] Nicolas Signolle, Marinette Revenu, Benoît Plancoulaine, and Paulette Herlin. Wavelet-based multiscale texture segmentation :

- Application to stromal compartment characterization on virtual slides. *Signal Processing*, 90(8) :2412–2422, 2010. Special Section on Processing and Analysis of High-Dimensional Masses of Image and Signal Data.
- [Tar75] Robert Tarjan. Efficiency of a good but not linear set union algorithm. *Journal of the ACM*, 22(2) :215–225, 1975.
- [TFM96] Simon Thorpe, Denis Fize, and Catherine Marlot. Speed of processing in the human visual system. *Nature*, 381(6582) :520–522, June 1996.
- [TP75] Steven L. Tanimoto and Theodosios Pavlidis. A hierarchical data structure for picture processing. In *Computer Graphics and Image Processing*, volume 4, pages 104–119, 1975.
- [Tut63] William T. Tutte. A census of planar maps. *Canad. J. Math.*, 15 :249–271, 1963.
- [Wan08] Yin Hai Wang. *Computer assisted diagnosis of cervical intraepithelial neoplasia (CIN) using histological virtual slides*. Thèse de doctorat, School of Electronics, Electrical Engineering and Computer Science, Queen’s University, April 2008.
- [Wei85] Kevin Weiler. Edge-based data structures for solid modeling in curved-surface environments. *IEEE Comput. Graph. Appl.*, 5(1) :21–40, 1985.
- [Whi49] John Henry C. Whitehead. Combinatorial homotopy ii. *Bull. Amer. Math. Soc.*, 55(5) :453–496, 1949.
- [WK94] Dieter Willersinn and Walter G. Kropatsch. Dual graph contraction for irregular pyramids. *12th IAPR International Conference on Pattern Recognition*, 3 :251–256, 1994.

Table des figures

1	Scanners de lames entières	11
2	L'arbre des objets du domaine d'analyse histologique.	12
3	Processus hiérarchique descendant pour l'analyse des lames histologiques.	14
1.1	Pavages en dimension 2	19
1.2	Notion de connexité entre les pixels d'une image	19
1.3	Le ruban de Möbius	21
1.4	Complexes cellulaires en dimension 2	22
1.5	Topologie d'un complexe cellulaire	23
1.6	(k, l) -chemins dans un complexe cellulaire	25
1.7	Adjacences entre deux régions	26
1.8	Tableaux d'étiquettes	27
1.9	Extensions des tableaux d'étiquettes	29
1.10	Frontières des régions	29
1.11	Matrice de Khalimsky	30
1.12	Frontières interpixel	31
1.13	Graphes d'adjacence de régions	32
1.14	Graphes d'adjacence : fusion de sommets	33
1.15	Graphes duaux	35
1.16	Graphes duaux : limite de la relation d'imbrication	36
1.17	Construction d'un ensemble de brins	37
1.18	Cartes combinatoires	38
1.19	Cartes combinatoires : cycles	40
1.20	Cartes généralisées	41
1.21	Cartes topologiques	43
1.22	Simplifications progressives d'une carte combinatoire	45

1.23 Quadtrees	50
1.24 Pyramides régulières	52
1.25 Pyramides de graphes simples	54
1.26 Décimation par couplage maximal	57
1.27 Pyramides généralisées	62
2.1 Décomposition d'une image en tuiles	68
2.2 Entités équivalentes au bord des tuiles	70
2.3 Ensembles de brins d'une tuile	73
2.4 Opérateur β'_2	75
2.5 Conséquences d'une décomposition en tuiles	76
2.6 Partitionnement en bords réels et fictifs	78
2.7 Involutions sur un ensemble de tuiles	80
2.8 Opérateur δ_1	82
2.9 Cartes tuilées	83
2.10 Connexion entre deux tuiles adjacentes	88
2.11 Insertion d'une arête pendante élémentaire	91
2.12 Décomposition d'une région en régions élémentaires	93
2.13 Traitement des arêtes pendantes	93
2.14 Fusion de régions adjacentes	95
2.15 Décomposition/fusion d'une région dans une carte tuilée	97
3.1 Fenêtres de réduction	111
3.2 Fenêtres de réduction : cas des bords	112
3.3 Pavages hiérarchiques	118
3.4 Construction d'une pyramide de tuiles	119
3.5 Pyramide descendante tuilée	121
3.6 Pyramide descendante tuilée : schéma de construction	122
3.7 Projection d'un $(0, 1)$ -chemin	125
3.8 Projection des $(0, 1)$ -chemins frontières	126
3.9 Relations <i>prédécesseur/successeur</i>	129
3.10 Identification des arêtes filles	132
3.11 Identification des régions filles	135
3.12 Projection directe	137
3.13 Projection optimisée	138
3.14 Extraction des successeurs d'une tuile	139

3.15	Extraction d'une pyramide	143
4.1	Approches par blocs	149
4.2	Approches hiérarchiques	152
4.3	Critères de segmentation	158
4.4	Détection des bords fictifs	162
4.5	Comparaison des méthodes de projection	164
4.6	Focus d'attention	166
4.7	Classification sur histogramme	168
4.8	Classifications successives d'une image histologique	169
4.9	Extraction des mitoses	171

Liste des Algorithmes

1	Extraction de la carte d'une tuile	85
2	Construction de l'arbre d'imbrication des régions	87
3	Connexion de tuiles adjacentes	88
4	Extraction d'une carte tuilée	89
5	Insertion d'une arête pendante élémentaire	92
6	Décomposition d'une région dans une tuile	94
7	Fusion d'une région dans une tuile	96
8	Décomposition/fusion d'une région dans une carte tuilée	98
9	Raffinement d'une carte tuilée par décomposition/fusion des régions.	100
10	Enregistrement d'une tuile	103
11	Chargement d'une tuile	105
12	Identification des arêtes filles	133
13	Identification des régions filles	134
14	Identification des régions imbriquées	135
15	Oracle	140
16	Extraction des successeurs d'une tuile	141
17	Extraction d'une pyramide descendante tuilée	142

Index

Symboles

\mathcal{B}^T	73
\mathcal{B}_R^T	78
\mathcal{B}_R^T	78
\mathcal{B}^t	71
\mathcal{B}_∞^t	71
\mathcal{B}_∂^t	71
$\mathcal{B}_\partial^{t,t'}$	72
\mathcal{R}	77
α_i	41
β'_2	74
β_i	38
δ_1	81
δ_2	79
$\varphi^{t,t'}$	72
SC	59
CW	59
\mathcal{B}	37
\mathcal{A}	34
\mathcal{G}	120
\mathcal{I}	110
\mathcal{K}	57
\mathcal{L}	123
BS	58
SS	52
AN	57
SN	52
\mathcal{P}	111
BP	123
CR	51
FR	111

\mathcal{T}	118
\mathcal{S}	34

A

adjacence	
cellules	24
multi-	26
régions	25
tuiles	71
algorithme de Dijkstra	137
arête	
élémentaire	44
équivalente	71
contraction	34, 57
pendante	45
pendante élémentaire	91
redondante	57
suppression	34, 57
arbre	28
arbre d'imbrication	44

B

bord	
fictif	77
réel	77
boucle	45
brin	37
élémentaire	44
couture	38
prédécesseur	120, 127
projeté	127
représentant	42

- successeur 120, 127
- C**
- carte
- combinatoire 38
 - dual 40
 - généralisée 41
 - minimale 45
 - raffinement 99
 - topologique 42
 - tuilée 82
- causalité 120
- cellules
- adjacence 24
 - degré 24
- champ récepteur 51
- chemin 24
- ligne topologique 24
 - maximal 31
 - simple 24
- complexe
- étoile 22
 - adhérence 23
 - bord 23
 - cardinal 23
 - cellulaire abstrait 21
 - fermé 22
 - incidence 23
 - intérieur 23
 - ouvert 22
 - sous- 22
- composante connexe 24
- connexion
- chemin 59
 - séquence 59
- connexité 19
- 4- 19
 - 8- 19
- couplage maximal 56
- critère 25
- colorimétrique 25, 156
- d'homogénéité 25
- de décomposition 138
- de fusion 90
- de segmentation 85, 142
- fusion 141
- géométrique 25, 156
- hiérarchique 157
- topologique 25, 157
- cycle 39
- D**
- doublet 44
- E**
- ensemble
- symétrique 45
- espace
- d'Alexandroff 21
 - de Hausdorff 20
 - topologique 19
- F**
- fenêtre de réduction 50
- d'un pointel 111
 - d'une tuile 118
- focus d'attention 48, 108
- fonction de bord 71
- forêt 28
- ensembles disjoints 28
 - recouvrante 57
- frontière 28
- interpixel 30
 - minimale 31
- G**
- geomap 46
- graphe
- d'adjacence de régions 32
 - dual 34
 - topologique des frontières ... 46
- H**
- homéomorphisme 20

I	
image	18
traitement	18
imbrication	26
interpixel	
éléments	21
cellules équivalentes	69
frontière	30
lignel	21
pixel	21
pointel	21
involution	38
isomorphisme	122, 127
L	
lignel	21
M	
matrice de Khalimsky	30
N	
noeud	28, 31
noyau	
de contraction	57, 58
de suppression	58
O	
oracle	77, 140
orbite	39
orbite généralisée	61
P	
partition	25
pavage	18
permutation	38
pixel	21
plongement	43
pointel	21
fenêtre de réduction	111
projection	123
bande de	123
d'un chemin	123
directe	136
optimisée	137
projeté direct	
d'un pixel	114
d'un pointel	114
pyramide	
combinatoire	58
d'images	50, 110
de graphe simples	52
de graphes duaux	56
de tuiles	118
descendante tuilée	120
généralisée	61
non trouée et non recouvrante	51
régulière	49
recouvrante	51
représentation explicite	61
représentation hiérarchique ..	61
représentation implicite ..	60, 62
trouée et non recouvrante ...	51
Q	
quadtrees	49
quasi-variété	20
R	
région	25
élémentaire	44
adjacence	25
allongée	51
décomposition	92, 97
frontière	28
fusion	94, 97
imbrication	26
multi-adjacence	26
prédécesseur	120, 128
projeté	127
raffinement	90
successeur	120
régions	
successeur	128

résolution	48	adhérence	20
racine	28	bord	20
RAG	32	digitale	21
relation de bord	21	fermé	19
S		intérieur	20
segment	31	ouvert	19
sommet		variété	20
non survivant	52	voisinage	20
survivant	52	tuile	67
stabilité		chargement	104
externe	54	déchargement	103
interne	54	fenêtre de réduction	118
structure		prédécesseur	117, 130
causale	120	projeté	127
d'un brin	100	successeur	117, 130
d'une région	101	topologique	68
d'une tuile	101	U	
géométrique	26	union-find	28
hiérarchique	47	V	
hiérarchique ascendante	48	variété	
hiérarchique descendante	48	non-orientable	21
relationnelle	31	orientable	21
T			
topologie	19		

Pyramides irrégulières descendantes pour la segmentation de grandes images histologiques

Résumé

Différents modes d'acquisition permettent d'obtenir des images de plusieurs gigaoctets. L'analyse de ces grandes images doit faire face à deux problèmes majeurs. Premièrement, le volume de données à traiter ne permet pas une analyse globale de l'image, d'où la difficulté d'en construire une partition. Deuxièmement, une approche multi-résolution est nécessaire pour distinguer les structures globales à faible résolution. Par exemple, dans le cadre des images d'histologie, les récentes améliorations des scanners permettent d'observer les structures cellulaires sur l'ensemble de la lame. En contrepartie, les images produites représentent jusqu'à 18 Go de données. De plus, l'agencement de ces cellules en tissus correspond à une information globale qui ne peut être observée qu'à faible résolution. Ces images combinent donc un aspect multi-échelle et multi-résolution.

Dans ce manuscrit, nous définissons un modèle topologique et hiérarchique adapté à la segmentation de grandes images. Nos travaux sont fondés sur les modèles existants de *carte topologique* et de *pyramide combinatoire*. Nous présentons le modèle de *carte tuilée* pour la représentation de grandes partitions ainsi qu'une extension hiérarchique, la *pyramide descendante tuilée*, qui représente la dualité des informations multi-échelle et multi-résolution. Enfin, nous utilisons notre modèle pour la segmentation de grandes images en histologie.

Mots-clés : modèles topologiques, cartes combinatoires, traitement d'images, imagerie médicale, pyramides irrégulières, segmentation

Top-down irregular pyramids for large histological images segmentation

Abstract

Some data acquisition devices produce images of several gigabytes. Analyzing such large images raises two main issues. First, the data volume to process forbids a global image analysis, hence a hard partitioning problem. Second, a multi-resolution approach is required to extract global features at low resolution. For instance, regarding histological images, recent improvements in scanners' accuracy allow nowadays to examine cellular structures on the whole slide. However, produced images are up to 18 GB. Besides, considering a tissue as a particular layout of cells is a global information that is only available at low resolution. Thus, these images combine multi-scale and multi-resolution information.

In this work, we define a topological and hierarchical model which is suitable for the segmentation of large images. Our work is based on the models of *topological map* and *combinatorial pyramid*. We introduce the *tiled map* model in order to encode the topology of large partitions and a hierarchical extension, the *tiled top-down pyramid*, to represent the duality between multi-scale and multi-resolution information. Finally, we propose an application of our model for the segmentation of large images in histology.

Keywords: topological model, combinatorial map, image processing, medical imaging, irregular pyramids, segmentation